Veritas Storage Foundation: Storage and Availability Management for Oracle Databases

Solaris

5.1

Storage Foudation: Storage and Availability Management for Oracle Databases

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Documentation version

PN:

Legal Notice

Symantec Corporation
350 Ellis Street

Mountain View, CA 94043

http://www.symantec.com

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's maintenance offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

For information about Symantec's Maintenance Programs, you can visit our Web site at the following URL:

www.symantec.com/techsupp/

Contacting Technical Support

Customers with a current maintenance agreement may access Technical Support information at the following URL:

www.symantec.com/techsupp/

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system

- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and maintenance contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Maintenance agreement resources

If you want to contact Symantec regarding an existing maintenance agreement, please contact the maintenance agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Contents

Chapter 1

Overview of Storage Foundation storage and availability management for Oracle databases

This chapter includes the following topics:

About the Storage Foundation Database (SFDB) tools for Oracle

Storage Foundation Enterprise product suites provide enhanced I/O methods to improve database performance and a set of tools for increased ease of management of storage in a database environment.

Perfomance enhancement methods shared with the Storage Foundation Standard products include:

- Veritas extension for Oracle Disk Manager (ODM)
- Veritas extension for Cached Oracle Disk Manager (Cached ODM)
- Veritas Quick I/O
- Veritas Cached Quick I/O

Storage Foundation Standard products include the following tools for storage management:

- Storage Checkpoints
- FlashSnap
- Dynamic Storage Tiering

To learn more about Storage Checkpoints, FlashSnap, and Storage Tiering features available with the Storage Foundation Standard products:

See the *Veritas Storage Foundation Advanced Features Guide*.

Storage Foundation Enterprise products provide an extended version of the basic storage management functionality provided in Storage Foundation Standard products:

- Database Checkpoints
- Database FlashSnap
- Database Cloning
- Database Dynamic Storage Tiering

This extended toolset is the Storage Foundation for Databases (SFDB) feature for enhanced management of Oracle databases. The SFDB tools provide enhanced ease-of-use commands which can be run by a database administrator without root priveleges to optimize storage for an Oracle database environment. This guide documents the deployment and use of the SFDB tools with Storage Foundation Enterprise Products. It is a supplemental guide to be used in conjunction with Veritas Storage Foundation product guides for users with Oracle databases.

Storage Foundation Enterprise product tools for optimizing storage for Oracle databases include:

- Veritas Quick I/O is a VxFS feature that improves the throughput for Oracle databases built on VxFS file systems. Quick I/O delivers raw device performance to databases run on VxFS, providing the administrative advantages of using file systems without the performance penalties.

- Veritas Cached Quick I/O further enhances database performance by leveraging large system memory to selectively buffer the frequently accessed data.
- Veritas Extension for Oracle Disk Manager is a custom storage interface designed specifically for Oracle. Oracle Disk Manager allows Oracle to improve performance and manage system bandwidth through an improved Application Programming Interface (API) that contains advanced kernel support for file I/O.
  Veritas Extension for Oracle Disk Manager supports Oracle Resilvering. With Oracle Resilvering, the storage layer receives information from the Oracle database as to which regions or blocks of a mirrored datafile to resync after a system crash. When using Oracle Resilvering, you can turn off VxVM Dirty Region Logging (DRL), which increases performance.
- SmartSync recovery accelerator increases the availability of mirrored volumes by only resynchronizing changed data. SmartSync reduces the time required to restore consistency, freeing more I/O bandwidth for business-critical applications.
  This feature is applicable only to databases that are configured on raw volumes. If supported by the database vendor, the SmartSync feature uses an extended interface between VxVM volumes and the database software to avoid unnecessary work during mirror resynchronization. For example, Oracle automatically takes advantage of SmartSync to perform database resynchronization when it is available.
- Database Checkpoint for Enterprise products enables you to create a point-in-time image of a file system. Storage Checkpoints are treated like any other VxFS file system and can be created, mounted, unmounted, and removed with VxFS and Veritas Storage Foundation administrative utilities.
- Database FlashSnap enables you to create, resynchronize, and reverse resynchronize an online point-in-time image of a database. You can use this image to perform backup, other maintenance tasks, or off-host processing while providing continuous data availability. Also, database administrators can perform these tasks without `root` privileges.
- Database clone commands for Database Storage Checkpoints or Database FlashSnap can be used for troubleshooting, reporting, and quality assurance for databases.
- Database Dynamic Storage Tiering enables you to manage your data so that less-frequently used data can be moved to slower, less expensive disks, allowing frequently-accessed data to be stored on the faster disks for quicker retrieval.
- The Symantec Storage Plugin for Oracle Enterprise Manager (OEM) enables you to view VxFS properties,VxVM volume and LUN information for database

objects such as tablespace, redo logs, controlfile, datafiles and others through the OEM interface.

- Storage Foundation Database cloning tool for STANDBY database using point-in-time copies Database Storage Checkpoints and Database Flashsnap.
- Veritas Cross-Platform Data Sharing allows data to be serially shared among heterogeneous systems where each system has direct access to the physical devices that hold the data. This feature can be used only in conjunction with Veritas Volume Manager. Shared or parallel access is possible for read-only data.
  Cross-Platform Data Sharing provides the fastest way to use Oracle's Transportable Tablespace (TTS) feature for migrating databases to different platforms in Oracle 10g or later versions. It provides the fastest way to use Oracle's Transportable Tablespace (TTS) feature for moving sets of tablespaces between databases on the same platform in Oracle9.
  See the *Veritas Storage Foundation Advanced Features Guide*.
- The Veritas NetBackup for Oracle Advanced BLI Agent software supports Block-Level Incremental (BLI) Backup to reduce database down time, backup time, and backup volume, as well as CPU usage and network overhead. (Contact your Sales Representative for information about this optional product.)

Storage Foundation extended storage management tools for Oracle databases are included with Enterprise licensing for the following products:

- Veritas Storage Foundation (SF)
- Veritas Storage Foundation High Availability (SF HA)
- Veritas Storage Foundation Cluster File System (SFCFS)
- Veritas Storage Foundation Cluster File System High Availability (SFCFS HA)
- Veritas Storage Foundation for Oracle RAC (SF for Oracle RAC)

Veritas Storage Foundation product suites

The following table lists the Symantec products and optionally licensed features available with each Veritas Storage Foundation product suite.

Table 1-1 Contents of Veritas Storage Foundation products

Storage Foundation version	Products and features
Storage Foundation Basic	Veritas File System Veritas Volume Manager

Table 1-1 Contents of Veritas Storage Foundation products *(continued)*

Storage Foundation version	Products and features
Storage Foundation Standard	Veritas File System Veritas Volume Manager Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Optionally licensed features: Veritas Volume Replicator
Storage Foundation Standard HA	Veritas File System Veritas Volume Manager Veritas Cluster Server Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Optionally licensed features: Veritas Volume Replicator Veritas Global Cluster Option
Storage Foundation Enterprise	Veritas File System Veritas Volume Manager Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Veritas Database Storage Checkpoint option Veritas Database Flashsnap Veritas Database Dynamic Storage Tiering Optionally licensed features: Veritas Volume Replicator

Table 1-1 Contents of Veritas Storage Foundation products *(continued)*

Storage Foundation version	Products and features
Storage Foundation Enterprise HA	Veritas File System Veritas Volume Manager Veritas Cluster Server Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Veritas Database Storage Checkpoint option Veritas Database Flashsnap Veritas Database Dynamic Storage Tiering Optionally licensed features: Veritas Volume Replicator Veritas Global Cluster Option
Storage Foundation Cluster File System	Veritas Storage Foundation Cluster File System Veritas Cluster Volume Manager Veritas Cluster Server Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Veritas Database Storage Checkpoint option Veritas Database Flashsnap Veritas Database Dynamic Storage Tiering Optionally licensed features: Veritas Volume Replicator

Table 1-1 Contents of Veritas Storage Foundation products *(continued)*

Storage Foundation version	Products and features
Storage Foundation Cluster File System HA	Veritas Storage Foundation Cluster File System Veritas Cluster Volume Manager Veritas Cluster Server Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Veritas Database Storage Checkpoint option Veritas Database Flashsnap Veritas Database Dynamic Storage Tiering Optionally licensed features: Veritas Volume Replicator Veritas Global Cluster Option
Storage Foundation for Oracle RAC	Veritas Storage Foundation Cluster File System Veritas Cluster Volume Manager Veritas Cluster Server Veritas Quick I/O option Veritas Extension for Oracle Disk Manager option Veritas Database Storage Checkpoint option Veritas Database Flashsnap Veritas Database Dynamic Storage Tiering Optionally licensed features: Veritas Volume Replicator Veritas Global Cluster Option

About the Storage Foundation for Databases (SFDB) repository database

The Storage Foundation for Databases (SFDB) repository or repository database stores metadata information required by the Storage Foundation for Databases tools. This information includes data about user databases, snapshot databases, storage configuration, scheduled tasks.

Note: The repository database requires only occasional interaction outside of the initial installation and configuration of Storage Foundation Enterprise products.

In this release of Storage Foundation products, the SFDB repository is stored in a relational database and is managed by SQLite3. The default location of the repository is on the mountpoint of the SYSTEM tablespace of the Oracle database. This enables it to be shared easily among cluster nodes. However, you can choose an alternate location if needed.

About Veritas Storage Foundation and High Availability products

The following sections describe the products and component software available in this Veritas Storage Foundation and High Availability Solutions release.

Component software

The following component softwares are available in this Veritas Storage Foundation and High Availability Solutions release.

Symantec Product Authentication Service

The Symantec Product Authentication Service is a common Symantec feature. This feature validates the identities that are based on existing network operating system domains (such as NIS and NT) or private domains. The authentication service protects communication channels among Symantec application clients and services through message integrity and confidentiality services.

Before you install the authentication service, read the Storage Foundation and High Availability Solutions appendix in the *Symantec Product Authentication Service Installation Guide*.

This document is located at the following directory in the release media.

`authentication_service/docs/vxat_install.pdf`

Before you select this method of setting up the authentication service, also read the product installation guide and release notes.

Storage Foundation Management Server

Storage Foundation Management Server (SF Management Server) provides a centralized management console for Veritas Storage Foundation products. You can use SF Management Server to monitor, visualize, and manage storage resources and generate reports about those resources. You are prompted to set up an optional SF Management Server managed host during every Storage Foundation product installation. SF Management Server is not available on the Storage Foundation and High Availability Solutions release and must be obtained separately. For information on ordering SF Management Server, visit the following URL:

www.symantec.com/enterprise/sfms

Cluster Management Console

The Veritas Cluster Server Cluster Management Console enables administration and analysis of all clusters in an enterprise from a single console. The console uses any supported Web browser. Review the following guide for detailed installation information.

This document is located at the following directory in the release media.

```
cluster_server/docs/vcs_install.pdf
```

Web Server for Storage Foundation Host Management

Web Server for Storage Foundation Host Management provides web-based management capabilities for Storage Foundation products. After installing the Web server, you can remotely administer Storage Foundation products without requiring a client-side installation.

Veritas Cluster Server

Before you install this product, complete the following tasks:

- Read the product release notes.
- Review the preliminary information.
- Follow the instructions in the *Veritas Cluster Server Installation Guide*.

These documents are located at the following directory in the release media.

```
cluster_server/docs/vcs_notes.pdf
cluster_server/docs/vcs_install.pdf
```

Installation overview

Cluster Server is included in all Veritas high availability (HA) products. If you have purchased a Veritas product suite that includes Cluster Server, it is installed and updated as part of that product suite. Do not install or update it as an individual component.

About this product

Veritas Cluster Server by Symantec (VCS) is an open systems clustering solution that provides the following benefits:

- Eliminates the downtime
- Facilitates the consolidation and the failover of servers
- Effectively manages a wide range of applications in heterogeneous environments

Veritas Volume Replicator

Before you install this product, complete the following tasks:

- Read the product release notes.
- Review the preliminary information.
- Follow the instructions in the *Storage Foundation Installation Guide*.

These documents are located at the following directories in the release media.

```
storage_foundation/docs/sf_notes.pdf
storage_foundation/docs/sf_install.pdf
```

About this product

Veritas Volume Replicator by Symantec is an optional, separately-licensable feature of Veritas Volume Manager. Volume Replicator is a fully integrated component of Veritas Volume Manager. This component replicates data to remote locations over any standard IP network to provide continuous data availability.

The Volume Replicator option is available with Veritas Storage Foundation Standard and Enterprise products.

Veritas Storage Foundation

Before you install this product, complete the following tasks:

- Read the product release notes.
- Review the preliminary information.
- Follow the instructions in the *Storage Foundation Installation Guide*.

These documents are located at the following directories in the release media.

```
storage_foundation/docs/sf_notes.pdf
storage_foundation/docs/sf_install.pdf
```

For HA installations, also read the following documents.

```
cluster_server/docs/vcs_notes.pdf
cluster_server/docs/vcs_install.pdf
```

About this product

Veritas Storage Foundation by Symantec includes Veritas File System by Symantec (VxFS) and Veritas Volume Manager by Symantec (VxVM) with various feature levels.

Veritas File System is a high performance journaling file system that provides easy management and quick-recovery for applications. Veritas File System delivers scalable performance, continuous availability, increased I/O throughput, and structural integrity.

Veritas Volume Manager removes the physical limitations of disk storage. You can configure, share, manage, and optimize storage I/O performance online without interrupting data availability. Veritas Volume Manager also provides easy-to-use, online storage management tools to reduce downtime.

You add high availability functionality to Storage Foundation HA by installing Veritas Cluster Server software.

VxFS and VxVM are included in all Veritas Storage Foundation products. If you have purchased a Veritas Storage Foundation product, VxFS and VxVM are installed and updated as part of that product. Do not install or update them as individual components.

Veritas Storage Foundation has the following products:

- Storage Foundation Standard
- Storage Foundation Standard HA
- Storage Foundation Enterprise
- Storage Foundation Enterprise HA

The Veritas products table lists the optionally licensed features available with each Storage Foundation version.

See "Veritas Storage Foundation product suites" on page 20.

Veritas Storage Foundation Basic

Veritas Storage Foundation Basic by Symantec (SF Basic) is a special product that is available on a separate Storage Foundation Basic disc. Also, SF Basic is available by download from the Symantec Web site. SF Basic is not part of the Storage Foundation and High Availability Solutions product suite. For complete information on ordering this product, licensing, and technical support, visit the following URL:

www.symantec.com/enterprise/sfbasic

Use SF Basic with Veritas Storage Foundation Management Server to set up a centrally managed host to monitor, visualize, and manage storage resources across multiple installations. SF Management Server is a separately available product.

For information on ordering SF Management Server, visit the following Symantec Web site:

www.symantec.com/enterprise/sfms

About this product

Storage Foundation Basic supports all Storage Foundation Standard features, however, there are deployment and technical support limitations.

Limited deployment

Storage Foundation Basic has a limited set of configurations.

SF Basic deployment is limited to the following configurations:

- Maximum four VxVM volumes per physical server (excludes the system volumes that are required for starting the system from root disks)
- Maximum four VxFS file systems per physical server (excludes root file systems)
- Maximum server capacity of two CPU sockets

Technical support

Technical support is self-service only, available from the Veritas Support Web site. You can purchase additional support corresponding to the terms of the Storage Foundation Basic license. To access the self-service knowledge base, go to the following URL:

http://entsupport.symantec.com

Installation overview

The Veritas Storage Foundation software is on the Veritas Storage Foundation Basic disc. You cannot install Storage Foundation Basic if another Veritas Storage Foundation product is installed unless that product is first removed from the system.

Before you install this product, complete the following tasks:

- Read the product release notes.
- Review the preliminary information.
- Follow the instructions in the *Storage Foundation Installation Guide*.

```
storage_foundation/docs/sf_install.pdf
```

Upgrade paths

Storage Foundation Basic offers several upgrade paths.

Storage Foundation Basic provides the following upgrade paths:

- Upgrade from SF Basic to Storage Foundation Standard or Storage Foundation Enterprise.
- Upgrade from SF Basic to Storage Foundation Enterprise plus Veritas Volume Replicator by installing the appropriate products.
- Upgrade from SF Basic to any other Storage Foundation product by uninstalling SF Basic and by installing the new product.

To determine the release level of any Storage Foundation product on your system, run the `vxlicrep` command.

See the *Storage Foundation Installation Guide* for more information on upgrades.

Storage Foundation Cluster File System

Before you install this product, complete the following tasks:

- Read the product release notes.
- Review the preliminary information.
- Follow the instructions in the *Storage Foundation Cluster File System Installation Guide*.

These documents are located at the following directories in the release media.

```
storage_foundation_cluster_file_system/docs/sfcfs_notes.pdf
storage_foundation_cluster_file_system/docs/sfcfs_install.pdf
cluster_server/docs/vcs_notes.pdf
```

About Veritas Storage Foundation Cluster File System

Veritas Storage Foundation Cluster File System by Symantec extends Veritas File System and Veritas Volume Manager to support shared data in a storage area network (SAN) environment. Using Storage Foundation Cluster File System, multiple servers can concurrently access shared storage and files transparently to applications.

Storage Foundation Cluster File System HA adds the failover functionality of Veritas Cluster Server. This functionality can protect everything from a single critical database instance to very large multiple-application clusters in networked environments. Veritas Storage Foundation Cluster File System also provides increased automation and intelligent management of availability and performance.

The Veritas Volume Replicator feature, which replicates data to remote locations over an IP network, can also be licensed with this product.

Storage Foundation for Oracle RAC

Before you install this product, review the preliminary information.

Before you start the installation, read the component product release notes and installation guide.

These documents are located at the following directories in the release media.

```
storage_foundation_for_oracle_rac/docs/sfrac_notes.pdf
storage_foundation/docs/sf_notes.pdf
cluster_server/docs/vcs_notes.pdf
storage_foundation_for_oracle_rac/docs/sfrac_install.pdf
```

About Veritas Storage Foundation for Oracle® RAC by Symantec

Veritas Storage Foundation for Oracle® RAC by Symantec is an integrated suite of Veritas storage management and high-availability software. The software is engineered to improve performance, availability, and manageability of Real Application Cluster (RAC) environments. Certified by Oracle Corporation, Veritas Storage Foundation for Oracle RAC delivers a flexible solution that makes it easy to deploy and manage RAC.

The Veritas Volume Replicator feature, which replicates data to remote locations over an IP network, can also be licensed with this product.

How Veritas Database FlashSnap works

Veritas Database FlashSnap is a feature included with Veritas Storage Foundation Enterprise products. It is also a separately licensed option available with Veritas Storage Foundation Standard products.

Veritas Database FlashSnap offers a flexible and efficient means of managing business-critical data. Database FlashSnap lets you capture an online image of an actively changing database at a given instant, called a point-in-time copy. You can perform system backup, upgrade, or perform 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.

Database FlashSnap takes advantage of the Persistent FastResync and Disk Group Content Reorganization features of VxVM. Database FlashSnap also streamlines database operations. Once configured, the database administrator can create snapshots, resynchronize data, and reverse resynchronize data without involving the system administrator.

Storage Foundation for Databases (SFDB) tools provide three commands that can be executed by the database administrator and do not require root privileges:

- `dbed_vmchecksnap`
- `dbed_vmsnap`
- `dbed_vmclonedb`

These commands enable database administrators take advantage of the VxVM snapshot functionality without having to deal with storage operations in day-to-day database uses. To use Database FlashSnap, you must configure the volumes used by the database.

Database snapshot and backup options

You can configure the following database components for cloning and recovery of databases:

- Storage Checkpoints
- Database FlashSnap

The following sections provide a brief overview of these features.

Storage Checkpoints for recovery

A Storage Checkpoint creates an exact image of a database instantly and provides a consistent image of the database from the point in time the Storage Checkpoint was created. The Storage Checkpoint image is managed and available through the command line interface (CLI).

Because each Storage Checkpoint is a consistent, point-in-time image of a file system, Storage Rollback is the restore facility for these on-disk backups. Storage Rollback rolls back the changed blocks that are contained in a Storage Checkpoint into the primary file system for faster database restoration.

The combination of data redundancy (disk mirroring) and Storage Checkpoints is recommended for highly critical data to protect them from both physical media failure and logical errors.

Advantages and limitations of Storage Checkpoints

Storage Checkpoints and rollback provides the following advantages:

- Initially, a Storage Checkpoint contains no data—it contains only the inode list and the block map of the primary fileset. The block map points to the actual data on the primary file system.
- Because only the inode list and block map are needed and no data is copied, creating a Storage Checkpoint takes only a few seconds and very little space.
- A Storage Checkpoint keeps track of block change information and thereby enables incremental database backup at the block level.
- A Storage Checkpoint helps recover data from incorrectly modified files.
- A Storage Checkpoint can be mounted, allowing regular file system operations to be performed. Mountable Storage Checkpoints can be used for a wide range of application solutions that include backup, investigations into data integrity, staging upgrades or database modifications, and data replication solutions.

The limitations of Storage Checkpoints are as follows:

- Storage Checkpoints can only be used to restore from logical errors (for example, a human error).
- Because all the data blocks are on the same physical device, Storage Checkpoints cannot be used to restore files due to a media failure. A media failure requires a database restore from a tape backup or a copy of the database files that are kept on a separate medium.

Database FlashSnap for cloning

Veritas Database FlashSnap helps to create a point-in-time copy of a database for backup and off-host processing. Database FlashSnap lets you make backup copies of your volumes online and with minimal interruption to users.

Database FlashSnap lets you capture an online image of an actively changing database at a given instant that is known as a snapshot. A snapshot copy of the database is referred to as a database snapshot. You can use a database snapshot on the same host as the production database or on a secondary host sharing the same storage. A database snapshot can be used for off-host processing applications, such as backup, data warehousing, and decision-support queries. When the snapshot is no longer needed, the database administrator can import the original snapshot back to the primary host and resynchronize the snapshot to the original database volumes. Database FlashSnap commands are executed from the command line interface.

Database FlashSnap advantages

Database FlashSnap provides the following advantages:

- The database snapshot can be used on the same host as the production database or on a secondary host sharing the same storage.
- In many companies, there is a clear separation between the roles of system administrators and database administrators. Creating database snapshots typically requires superuser (root) privileges, the privileges that database administrators do not usually have. Because superuser privileges are not required, Database FlashSnap overcomes these obstacles by enabling database administrators to easily create consistent snapshots of the database.

How Database Dynamic Storage Tiering works

Today, more and more data needs to be retained. Eventually, some of the data is no longer needed as frequently, but it still takes up a large amount of disk space. Database Dynamic Storage Tiering matches data storage with the data's usage requirements so that data is relocated based on requirements determined by the database administrator (DBA). The feature enables you to manage your data so that less-frequently used data can be moved to slower, less expensive disks, allowing frequently-accessed data to be stored on the faster disks for quicker retrieval.

The DBA can create a file allocation policy based on filename extension before new files are created, which will create the datafiles on the appropriate tier during database creation.

The DBA can also create a file relocation policy for database files, which would relocate files based on how frequently a file is used.

About Veritas NetBackup (optional)

Veritas NetBackup provides backup, archive, and restore capabilities for database files and directories contained on client systems in a client-server network. NetBackup server software resides on platforms that manage physical backup storage devices. The NetBackup server provides robotic control, media management, error handling, scheduling, and a repository of all client backup images.

Administrators can set up schedules for automatic, unattended full and incremental backups. These backups are managed entirely by the NetBackup server. The administrator can also manually back up clients. Client users can perform backups, archives, and restores from their client system, and once started, these operations also run under the control of the NetBackup server.

Veritas NetBackup, while not a shipped component of Veritas Storage Foundation Enterprise products, can be purchased separately.

How block-level incremental backup works

Block-Level Incremental (BLI) Backup extends the capabilities of NetBackup to back up only changed data blocks of Oracle database files. BLI Backup accomplishes this backup methodology using the Storage Checkpoint facility in the Veritas File System (VxFS) available through Storage Foundation for Databases tools. BLI Backup reduces both the time required to complete a database backup and the amount of data transferred during backups. It also allows more frequent backups, resulting in more up-to-date backup images. When restoring from backups, the restore time is increased only by the extra time needed for NetBackup to apply the incremental backups after a full restore completes. However, frequent incremental backups can speed up the database recovery by reducing the number of redo logs to apply.

BLI Backup is particularly useful in a database environment where a database can be hundreds of gigabytes or terabytes. Using traditional backup methods for an offline database backup, any change in the database file—no matter how small—requires backing up the entire database file. Using BLI Backup, only modified data blocks need to be backed up.

Note: To allow BLI Backups, the database must be on VxFS file systems using the Version 4, 5, 6, or 7 layout, and you must have a valid Veritas Storage Foundation Enterprise license. Use the `fstyp -v device` command to determine the layout version of your file system. See the `vxupgrade`(1M) manual page for more information.

For information on how to install, configure, and use Veritas NetBackup for Oracle Advanced BLI Agent to perform Oracle database backups, see the *Veritas NetBackup for Oracle Advanced BLI Agent System Administrator's Guide*.

Chapter 2

Deploying Storage Foundation for Databases (SFDB) tools

This chapter includes the following topics:

About Storage Foundation for Databases (SFDB) tools deployment options

Storage Foundation extended storage management tools for Oracle databases are included with Enterprise licensing for the following products:

- Storage Foundation, which supports host systems with single-instance Oracle
- Storage Foundation HA, which supports host systems with single-instance Oracle and automatic failover
- Storage Foundation for Cluster File System, which supports clustered host systems with single-instance Oracle
- Storage Foundation for Cluster File System HA, which supports clustered host systems with automatic failover and single-instance Oracle
- Storage Foundation for Oracle RAC, which supports clustered host systems with automatic failover and multiple-instance Oracle

New features for Storage Foundation for Databases (SFDB) tools for this release

New features in the Storage Foundation for Databases tools package for database storage management for this release:

- SQLlite repository
- Multiple disk group support for FlashSnap
- Oracle Dataguard support
- Oracle Enterprise Manager (OEM) Plugin
- Storage Foundation for Cluster File (HA) System support
- Cached ODM support

Feature changes for Storage Foundation for Databases (SFDB) tools for this release

If you are upgrading from Storage Foundation for Oracle (HA) 4.x or 5.0 to Storage Foundation 5.1, or from Storage Foundation for Oracle RAC 5.0 to 5.1, the following changes in functionality will apply.

Commands which have changed:

- `sfua_db_config` functionality is changed: this command is no longer needed to create a SFDB repository. The functionality of `sfua_db_config` is now used to set user and group access to various SFDB directories.
- Use the `dbed_update` command to create a new SQLite SFDB repository.
- `sfua_rept_adm` was used in 5.0 to perform repository backup and restore and this command will be obsolete in 5.1.
- The `sfua_rept_util` command is used to perform SQLite repository backup and restore.

Commands which continue to be supported:

- `dbed_update`
- Database Storage Checkpoint commands: `dbed_ckptcreate`, `dbed_ckptdisplay`, `dbed_ckptmount`, `dbed_ckptquota`, `dbed_ckptremove`, `dbed_ckptrollback`, `dbed_clonedb`, `dbed_ckptumount`
- Quick I/O commands: `qio_getdbfiles`, `qio_recreate`, `qio_convertdbfiles`
- Database Flashsnap commands: `dbed_vmchecksnap`, `dbed_vmclonedb`, `dbed_vmsnap`
- Database Dynamic Storage Tiering commands: `dbdst_addvol`, `dbdst_admin`, `dbdst_classify`, `dbdst_convert`, `dbdst_file_move`, `dbdst_partition_move`, `dbdst_preset_policy`, `dbdst_rmvol`, `dbdst_show_fs`, `dbdst_tbs_move`, `dbdst_report`

Storage Foundation for Databases (SFDB) tools features which are no longer supported

Commands which are no longer supported:

- ORAMAP (`libvxoramap`)
- Storage mapping commands `dbed_analyzer`, `vxstorage_stats`
- DBED providers(DBEDAgent), Java GUI, and `dbed_dbprocli`.

  The SFDB Oracle features can only be accessed through the command line interface. However, Veritas Storage Foundation Manager 2.1 (a separately licensed product) can display Oracle database information such as tablespaces, database to LUN mapping, and tablespace to LUN mapping.
- Storage statistics: commands`dbdst_makelbfs`, `vxdbts_fstatsummary`, `dbdst_fiostat_collector`, `vxdbts_get_datafile_stats`
- `dbed_saveconfig`, `dbed_checkconfig`

- `dbed_ckptplan`, `dbed_ckptpolicy`
- `qio_convertdbfiles -f` option which is used to check for file fragmentation
- `dbed_scheduler`

Requirements for Storage Foundations for Databases (SFDB) tools

Product requirements are included in:

- *Veritas Storage Foundation™ Release Notes*
- *Veritas Storage Foundation™ for Cluster File System Release Notes*
- *Veritas Storage Foundation™a for Oracle RAC Release Notes*

The hardware compatibility list contains information about supported hardware and is updated regularly. For the latest information on supported hardware visit the following URL:

http://entsupport.symantec.com/docs/330441

For the most current information on Storage Foundation products and single instance Oracle versions supported, see:

http://entsupport.symantec.com/docs/331625

For the most current information on Storage Foundation for Oracle RAC and Oracle RAC versions supported, see:

http://entsupport.symantec.com/docs/280186

Review the current Oracle documentation to confirm the compatibility of your hardware and software.

The following indicates Oracle support at the time of publication.

Table 2-1 Supported Oracle and Solaris Sparc combinations

Oracle version	Solaris 9 (32-bit)	Solaris 9 (64-bit)	Solaris 10 (64-bit)
9iR2 (*) 9.2 (32-bit)	Yes	Yes	Yes
9iR2 (*) 9.2 (64-bit)	No	Yes	Yes

Table 2-1 Supported Oracle and Solaris Sparc combinations *(continued)*

Oracle version	Solaris 9 (32-bit)	Solaris 9 (64-bit)	Solaris 10 (64-bit)
10gR1 10.1(64-bit)	No	Yes	Yes
10gR2 10.2 (64-bit)	No	Yes	Yes
11gR1 11.1(64-bit)	No	Yes	Yes

Table 2-2 Supported Oracle and Solaris Operon combinations

Oracle version	Solaris 10 (64-bit)
10gR1 10.1(64-bit)	Yes
10gR2 10.2 (64-bit)	Yes

(*) ODM is not supported on Oracle 9i.

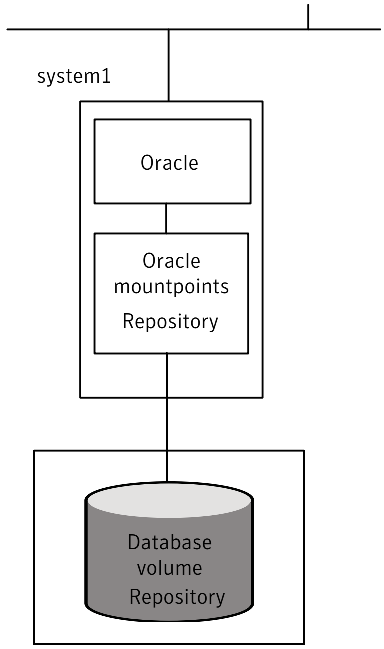
Deploying Storage Foundation for Databases (SFDB) tools with Storage Foundation

If you are deploying the SFDB tools with Storage Foundation (single instance Oracle) your setup configuration will reflect the following conditions:

- A single instance of Oracle is set up on system1 with SF.
- The database and datafiles are online on system1.
- You must run the SFDB tools commands on system1.

In the figure below the repository directory resides in the Oracle mountpoints.

Figure 2-1 Storage Foundation setup example



Deploying Storage Foundation for Databases (SFDB) tools with Storage Foundation HA

If you are deploying the SFDB tools with Storage Foundation HA (single instance Oracle) your setup configuration will reflect the following conditions:

- A highly available single instance of Oracle is set up on system1 and system2 with SF HA.
- The database and datafiles are online on system1 .
- The database repository is online on system1.
- You must run the SFDB tools commands on system1 where the database is online.
- The SFDB tools commands will fail on system2.
- ODM is supported for single instance Oracle only.

In the figures below the repository directory resides in the Oracle mountpoints.

Figure 2-2 Storage Foundation HA setup example

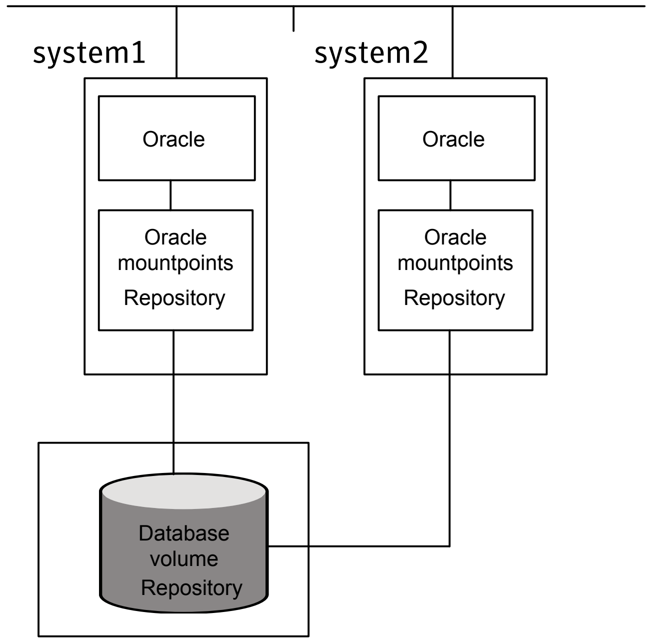
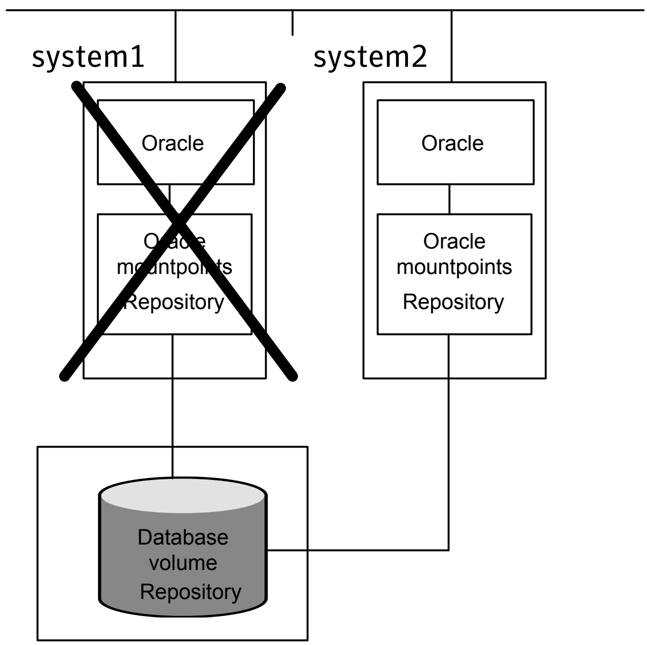


Figure 2-3 Storage Foundation HA failover example



Deploying Storage Foundation for Databases (SFDB) tools with SF Cluster File System (HA)

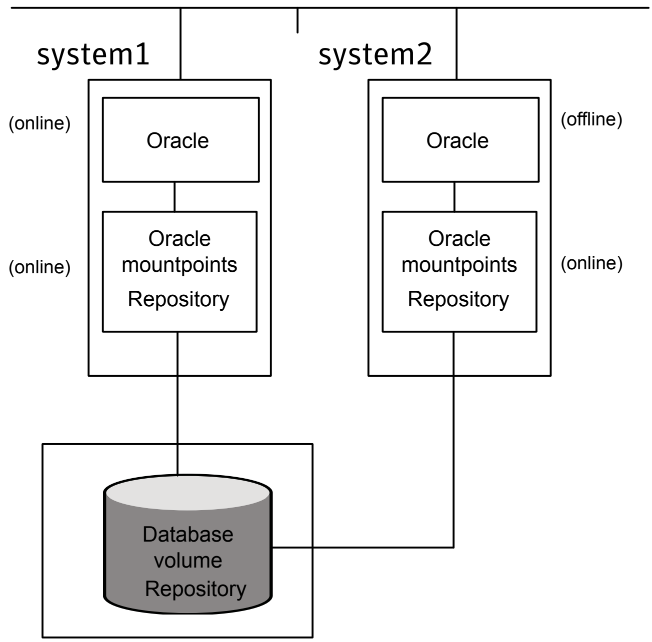
If you are deploying the SFDB tools with SF Cluster File System HA (single instance Oracle) your setup configuration will reflect the following conditions:

- A highly available parallel cluster with a single instance of Oracle is set up on system1 and system2 with SF Cluster File System HA.
- The database is online on system1.
- The datafiles are mounted and shared on system1 and system2.
- The database repository is mounted and shared on system1 and system2.
- You must run the SFDB tools commands on system1 where the database is online.
- The SFDB tools commands will fail on system2.

- ODM is supported for single instance Oracle only.
- Clustered ODM is supported.

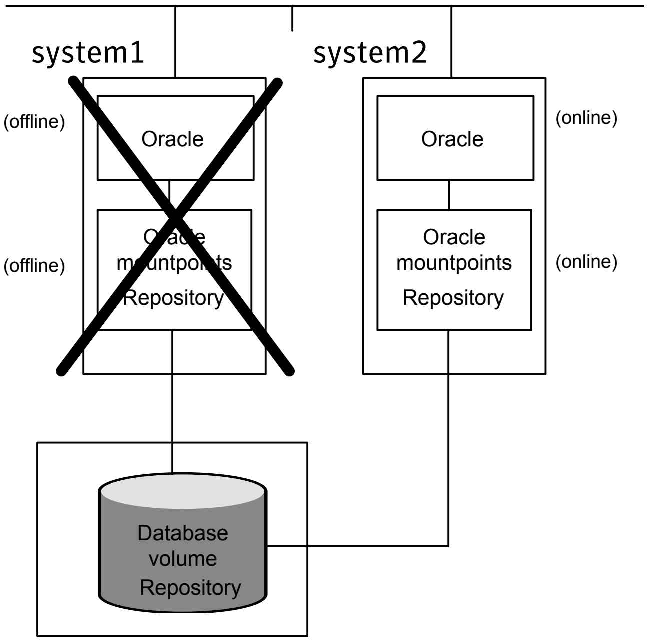
In the figures below the repository directory resides in the Oracle mountpoints.

Figure 2-4 SF Cluster File System or SF Cluster File System HA setup example



While the setup is the similar for SF Cluster File System and SF Cluster File System HA, failover to the backup system is automatic rather than manual for SF Cluster File System HA.

Figure 2-5 SFCFS HA failover example



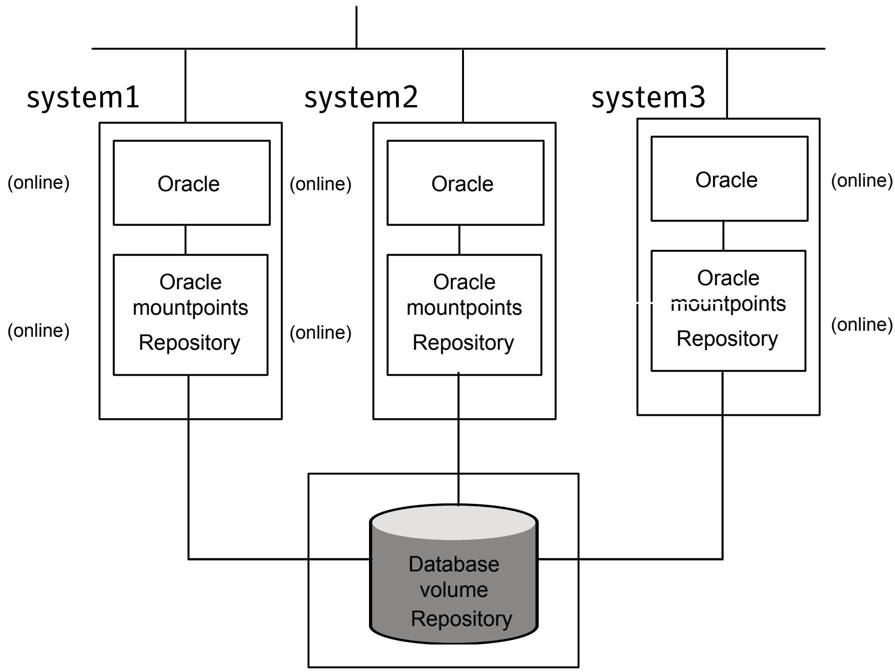
Deploying Storage Foundation for Databases (SFDB) tools with Storage Foundation for Oracle RAC

If you are deploying the SFDB tools with Storage Foundation for Oracle RAC (multiple instance Oracle) your setup configuration will reflect the following conditions:

- A highly available parallel cluster with a multiple instances of Oracle is set up on system1 and system2 with SF for Oracle RAC.
- The database is online on system1, system2, and system3.
- The datafiles are mounted and shared on system1, system2, and system3.
- The SFDB tools is mounted and shared on system1, system2, and system3.
- You can run the SFDB tools commands on system1, system2, and system3.
- Clustered ODM is supported.

In the figure below the repository directory resides in the Oracle mountpoints.

Figure 2-6 Deploying the database repository with Storage Foundation



For an SF Oracle RAC configuration, the systems are online in parallel and do not use failover mechanisms within the cluster.

Deploying in replicated environments

In an Oracle environment, there are two supported replication methods: Veritas Volume Replicator (VVR) and Oracle Data Guard. Storage Foundations for Databases (SFDB) tools support both methods.

Deploying with NetBackup

If you are deploying the Storage Foundations for Databases (SFDB) tools in an environment that includes Veritas NetBackup, your setup configuration will reflect the following considerations:

- NetBackup tools are supported
- Block-Level Incremental (BLI) Backup is supported
- You can back up Quick I/O and ODM files using NetBackup tools
- You can restore Quick I/O and ODM files using NetBackup tools
- You can back up and restore SF for Oracle RAC

Deploying with Storage Foundation Manager

Using the Storage Foundation Manager graphical user interface, you can only display database objects. You can not create Database Storage Checkpoints, Database FlashSnap snapshots, or Database Dynamic Storage Tiering policies.

Deploying with virtualization

If you are deploying the Storage Foundations for Databases (SFDB) tools in an environment that includes a Virtual Machine environment, the following are supported:

For Global Zone: Storage Foundation for Databases (SFDB) tools for the database environment are supported.

For Local Zone and Branded Zone, the following are supported:

- Quick I/O
- ODM

Solaris LDOM:

- Quick I/O
- ODM

RAC support depends on Oracle.

See Oracle documentation.

Deploying with SmartMove and Thin Provisioning

You can use SmartMove and Thin Provisioning with Storage Foundation products and your Oracle database. For information:

See the *Veritas Volume Manager Administrator's Guide*.

Chapter 3

Administering Storage Foundation for Databases (SFDB) tools

This chapter includes the following topics:

About administering the Storage Foundation for Databases (SFDB) tools

After you have installed and configured your base Storage Foundation product, you can set up and admininster the Storage Foundation for Databases tools for optimizing storage for your Oracle database. The following procedures apply for all Storage Foundation 5.1 Enterprise products unless specifically noted otherwise.

Setting up the Storage Foundation for Databases (SFDB) repository

The Storage Foundation for Database (SFDB) repository stores information used by SFDB tools. The repository contains Storage Checkpoint information, tablespace and datafile information, Database FlashSnap information, and Database DST parameters.

The SFDB repository:

- Uses SQLite3 for the repository.
- Is automatically created when you run `dbed_update` first time.

If the repository already exists, the command will refresh it. To use `dbed_update`:

- The database must be up and running.
- The ORACLE_SID and the ORACLE_HOME variables must be specified with and -S and -H options.

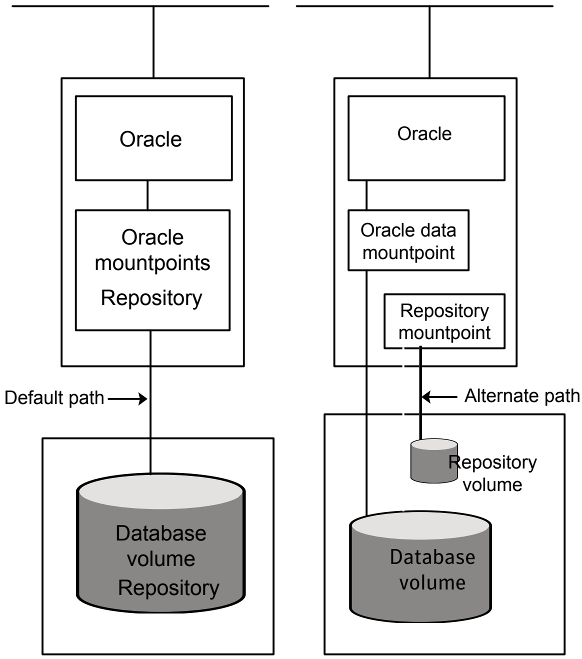
Locations for the SFDB repository

Locations for the repository:

- The default is on the volume where the SYSTEM tablespace resides.
- You can specify an alternate location using –R option of dbed_update.

In the figure below the repository directory resides in the Oracle mountpoints.

Figure 3-1 Locations for the SFDB repository



The alternate location must be:

- For Storage Foundation: on the local file system
- For Storage Foundation HA: on shared storage and the administrator must configure CFS or failover resource
- For Storage Foundation Cluster File System (HA): on shared storage and admin must configure CFS or failover resource
- For Storage Foundation for Oracle RAC: on shared storage and CFS mounted on all nodes

---

Note: Using the same alternate path for several databases is not supported.

Verifying the Storage Foundation for Databases tools are installed

To verify the Storage Foundation for Databases tools are installed

- Run the following:

  ```
  # pkginfo -l VRTSdbed
  ```

Preparing to set up the SFDB repository

Before you run any SF for Databases tools commands, you must the permissions correctly to prevent permisstion problems.

To prepare to set up the SFDB repository

- To set the permissions correctly for the SFDB directories, use the `sfua_db_config` command.

  For example:

  ```
  # /opt/VRTS/bin/sfua_db_config
  Welcome to the SFORA configuration script.
          This script sets the owner and group of various directories

  Are you ready to configure SFUA directories (y/n/q) [y]? y

  Enter login account name for DBA user: oracle

  Enter group name for DBA user: dba
  Owner and group changed for SFUA directories
  ```

Setting up the SFDB repository

Before you can create Database Storage Checkpoints, use Database FlashSnap, or create Database Dynamic Storage Tiering policies, you must set up the SF for Databases repository to maintain the information for them.

To set up the SFDB repository

- As Oracle DBA user, run the following to create or update the SFDB repository for the database.

  ```
  $ dbed_update -S $ORACLE_SID -H $ORACLE_HOME
  ```

  Options you can use:

Option	Sample value	Description
-S	ORACLE_SID	Specifies the name of the Oracle database whose information will be retrieved.
-H	ORACLE_HOME	The ORACLE_HOME setting for the ORACLE_SID database.
-G	SERVICE_GROUP	Specifies the VCS service group name for the ORACLE_SID database if it is under VCS control. This option is required in an HA environment.
-P	ORACLE_PFILE	Specifies the fully qualify path of the Oracle pfile if it is not located under ORACLE_HOME/dbs directory.
-R	REPOSITORY_PATH	Specifies a user-defined location for the repository. In a RAC or HA situation, it should be located on shared storage. The default location is on the mountpoint of the SYSTEM tablespace of the Oracle database. This way it is shared easily among cluster nodes. If REPOSITORY_PATH is slash (/), the repository location is switched back to the default path.
-I	N/A	An integrity check of the repository is performed, without refreshing from the Oracle database . If successful, the string "ok" is printed to stdout. In addition, the full path of the repository file is shown. Whenever refreshing from the database, an integrity check is automatically performed.
-o	list	The list of hosts and their internal ID is shown.

Option	Sample value	Description
-o	rename	Rename a host in the repository, by specifying old and new name. This is useful for DR after a failover, because the hostnames on the DR site differ from the original ones. For example: `-o rename,old_node=old_name,new_node=new_name`

Backing up and restoring the Storage Foundation for Databases (SFDB) repository

The SFDB repository should be backed up frequently after significant changes such as creating a new snapplan or creating checkpoints.

For the SFDB backup repository:

- The same default location is used as for the SFDB repository.
- The location can be specified by using the -d *directory* option.
- SFDB repositories for different databases must not share same backup location.
- The disk space at bacup location is automatically managed.
- Each SFDB repository backup file has a timestamp in its filename.
- The backup command preserves only last seven backups.

For high availability environments:

- Make sure the backup location is accessible by all nodes.
- The backup command can be run from any node.

Table 3-1 Options for `sfua_rept_util`

Option	Value	Description
-S	*ORACLE_SID*	Specifies the SID of the database for which the repository ie being backed up.
-o	backup	Specifies that the operation is to make a backup.

Table 3-1 Options for `sfua_rept_util` *(continued)*

Option	Value	Description
-o	list	Specifies that the operation is to list the backup files.
-o	restore	Specifies that the operation is a restoration of the selected backup file.
-d	*directory*	Specifies the directory location for the backup or restore of the file. This option is not required.
-f	*filename*	Specifies the file name for the repository backup which will be restored . This option is required for restoring a backup file.

To back up an SFDB repository

1. Verify that the database is online and that the backup location directory has write permission enabled for the DBA Oracle user.
2. Verify the SFDB repository is up to date:

```
# dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

3. As Oracle DBA user, run the backup command with the appropriate options.

   For example:

```
$ /opt/VRTS/bin/sfua_rept_util -S Prod -o backup
Backingup Repository for SID='Prod'
Copying Repository for SID=Prod to '/etc/vx/vxdba/Prod/.sfdb_rept'
Backup Repository for SID='Prod' Completed
```

To restore an SFDB repository

1. As Oracle DBA user, run the list command to display the SFDB repository backup ifles.

```
$ /opt/VRTS/bin/sfua_rept_util -S Prod -o list
Listing Repository Backups for SID='rac10g2'
TIME                  SIZE              NAME
2009-08-11@15:53 925696      rept.PROD.db_2009-08-11@15:53
2009-08-11@15:54 925696      rept.PROD.db_2009-08-11@15:54
```

2 As Oracle user, run the restore command with the appropriate options.

For example:

```
$ /opt/VRTS/bin/sfua_rept_util -S Prod -o restore -f rept.PROD.db_2009-08-11@15:54
Restoring Repository for SID='rac10g2'
Restore Repository for SID='rac10g2' Completed
```

Migrating the setup from IPV4 to Dual-Stack (IPV4/IPV6) for Storage Foundation HA

Use the following steps to migrate an IPV4 node which has SFHA and Veritas Cluster Server Agent for Oracle installed to IPV6. Oracle does not officially support pure IPV6 environment until Oracle 11gR2. There are issues for Oracle 11gR1 running in pure IPV6 mode on certain platforms. For more information:

See the Known issues section of the *Storage Foundation Release Notes*.

Assume the public NIC1 is on V4 network and private NIC3 is on V6 network.

To prepare to migrate a setup from IPV4 to Dual-Stack (IPV4/IPV6)

1 Modify system files to include both IPV4 and IPV6 entries.

- Modify /etc/resolv.conf to have both DNS entries for v6 network and DNS entries for V4 network.
- Also follow the instruction of your specific OS to modify the system network file to include the IPV6 address.
- Change the hostname to use ipv6 hostname.

2 You will need a Virtual IPV6 address with its hostname registered either in DNS or in /etc/hosts file.

3 Replace the NIC3 private heartbeat link with the V6 network connection and then run the below commands to make IPV6 online.

```
# ifconfig NIC3 down
# ifconfig NIC3 up down
```

Your node is in dual stack mode.

To migrate a setup from IPV4 to Dual-Stack (IPV4/IPV6)

1 Execute the steps below on the node where the Oracle Service Group is online to add two more resources (IP and NIC) for IPv6 to the existing Oracle Service Group. The Oracle Service Group will have two IP and two NIC resources one each for IPV4 and IPV6.

```
# haconf -makerw

# hares -add IP_ora_v6 IP Ora_VCS
  VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be se
  before agent monitors

# hares -modify IP_ora_v6 Device NIC3

# hares -modify IP_ora_v6 Address fd4b:454e:205a:
      111:211:43ff:fedf:0d65

# hares -modify IP_ora_v6 PrefixLen 64

# hares -add NIC_ora_v6 NIC Ora_VCS

# hares -modify NIC_ora_v6 Device NIC3

# hares -modify NIC_ora_v6 PingOptimize 0

# hares -link Listener_ora IP_ora_v6

# hares -link IP_ora_v6 NIC_ora_v6

# hagrp -enableresources Ora_VCS

# haconf -dump -makero
```

2 Bring the resources IP and NIC for V6 online.

```
# hares -online IP_ora_v6 -sys sys1
```

```
# hares -state | grep -i IP_ora_v6
IP_ora_v6      State               sys1     ONLINE
IP_ora_v6      State               sys2     OFFLINE

# hares -state | grep -i NIC_ora_v6
NIC_ora_v6     State               sys1     ONLINE
NIC_ora_v6     State               sys2     ONLINE
```

3 Bring the V4 resources Listener ora and IP resources offline. This will not have any impact on database accessiblity.

```
# hares -offline Listener_ora -sys sys1
# hares -offline IP_ora -sys sys1
```

4 Modifiy the listener.ora file by replacing the virtual V4 IP by virtual v6 IP.

For example, change 'swv80.location.sample.com' by 'swv80.ipv6.com'. An example of the listener.ora file might look something like:

```
$ cat /oracle/orabin/network/admin/listener.ora
# listener.ora Network Configuration
File: /oracle/orabin/network/admin/listener.ora
# Generated by Oracle configuration tools.

 LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = swv80.punipv6.com)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

5 Bring the listener resource online as below.

```
# hares -online Listener_ora -sys sys1
```

6 Remove the IP and NIC resources for V4 IP:

```
# haconf -makerw

# hares -unlink Listener_ora IP_ora

# hares -unlink IP_ora NIC_ora

# hares -delete IP_ora

# hares -delete NIC_ora

# haconf -dump -makero
```

7 Confirm that Oracle service group is online on V6 network.

8 Run the `dbed_update` command to start using the SFDB tools.

9 To verify if the dual-stack setup works on the other cluster nodes, fail over the service group when convenient.

To bring the system to a pure IPv6 evironment

1 Bring the setup in pure IPV6 setup by disabling the IPv4 network.

2 Verify that the host name contains a proper V6 entry using the output of the hostname command, which should display the proper v6 domain name.

3 Stop and start the vxdbd daemon by running the following commands on all the nodes in the cluster.

```
# /opt/VRTS/bin/vxdbdctrl stop
Stopping Veritas vxdbd

# /opt/VRTS/bin/vxdbdctrl start
Starting Veritas vxdbd
/opt/VRTSdbed/common/bin/vxdbd start SUCCESS

# /opt/VRTS/bin/vxdbdctrl stop
Stopping Veritas vxdbd

# /opt/VRTS/bin/vxdbdctrl start
Starting Veritas vxdbd
/opt/VRTSdbed/common/bin/vxdbd start SUCCESS
```

4 Run the `dbed_update` command to start using the SFDB tools.

5 To verify if the dual-stack setup works on the other cluster nodes, fail over the service group when convenient.

Updating the Storage Foundation for Databases (SFDB) repository after adding a node

After adding a node to a SF Enterprise products cluster, update the the SFDB repository to enable access for the new node.

To update the SFDB repository after adding a node

◆ Enter the following command:

For Storage Foundation HA:

```
# dbed_update -S $ORACLE_SID -H $ORACLE_HOME -G ORACLE_SERVICE_GROUP
```

For Storage Foundation for Cluster File System (HA):

```
# dbed_update -S $ORACLE_SID -H $ORACLE_HOME -G ORACLE_SERVICE_GROUP
```

For Storage Foundation for Orace RAC:

```
# dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

This completes the addition of the node to the SFDB repository.

Updating the Storage Foundation for Databases (SFDB) repository after removing a node

If you have already created an SFDB repository, you must update it to remove the reference for the node after removing the node from the cluster.

Note: If you have not created an SFDB repository, you do not need to perform the following steps.

To update the SFDB repository after removing a node

1. As Oracle user, list the nodes in the cluster:

   ```
   $ /opt/VRTSdbed/bin/dbed_rept_node -S $ORACLE_SID -o list
   ```

2. Run the following command after physically removing the node from the cluster.

   For example:

   ```
   $ /opt/VRTSdbed/bin/dbed_rept_node -S $ORACLE_SID -n NODE -o remove
   ```

   This completes the removal of the node from the SFDB repository.

Removing the Storage Foundation for Databases (SFDB) repository after removing the product

After removing the product, you can remove the SFDB repository file and any backups.

Removing the SFDB repository file will disable the SFDB tools.

To remove the SFDB repository

1. Change directories to the location of the local lookup information for the Oracle SID.

   For example:

   ```
   # cd /var/vx/vxdba/$ORACLE_SID
   ```

2. Identify the SFDB repository file and any associated links:

   For example:

   ```
   ls -al
   ```

```
lrwxrwxrwx   1 oracle   oinstall      26 Jul 21 13:58 .sfdb_rept -> \
/ora_data1/TEST/.sfdb_rept
cd /ora_data1/TEST
```

   Follow the symlink of .sfdb_rept.

3. Remove the repository directory containing the repository file and all backups.

   For example:

   ```
   # rm -rf .sfdb_rept
   ```

4. Remove the local lookup directory for the Oracle SID:

   ```
   # cd /var/vx/vxdba
   ```

   ```
   # rm -rf $ORACLE_SID
   ```

   This completes the removal of the SFDB repository.

Chapter 4

Using Veritas Quick I/O

This chapter includes the following topics:

About Quick I/O

Veritas Quick I/O is a VxFS feature included in Veritas Storage Foundation Standard and Enterprise products that lets applications access preallocated VxFS files as raw character devices. Quick I/O provides the administrative benefits of running databases on file systems without the typically associated degradation in performance.

Note: Veritas recommends that you use Veritas Extension for Oracle Disk Manager.

See "Setting up Veritas Extension for Oracle Disk Manager" on page 105.

How Quick I/O works

Veritas Quick I/O supports direct I/O and kernel asynchronous I/O and allows databases to access regular files on a VxFS file system as raw character devices.

The benefits of using Quick I/O are:

- Improved performance and processing throughput by having Quick I/O files act as raw devices.
- Ability to manage Quick I/O files as regular files, which simplifies administrative tasks such as allocating, moving, copying, resizing, and backing up datafiles.

  Note: Veritas recommends using Oracle Disk Manager.

  See "Converting Quick I/O files to Oracle Disk Manager files" on page 107.

How Quick I/O improves database performance

Quick I/O's ability to access regular files as raw devices improves database performance by:

- Supporting kernel asynchronous I/O
- Supporting direct I/O
- Avoiding kernel write locks on database files
- Avoiding double buffering

Creating database files as Quick I/O files using qiomkfile

The best way to preallocate space for tablespace containers and to make them accessible using the Quick I/O interface is to use the `qiomkfile`. You can use the `qiomkfile` to create the Quick I/O files for either temprory or permanent tablespaces.

Prerequisites	■ You can create Quick I/O files only on VxFS file systems. ■ If you are creating database files on an existing file system, run `fsadm` (or similar utility) to report and eliminate fragmentation. ■ You must have read/write permissions on the directory in which you intend to create Oracle Quick I/O files.
Usage notes	■ The `qiomkfile` command creates two files: a regular file with preallocated, contiguous space, and a file that is a symbolic link pointing to the Quick I/O name extension. ■ See the `qiomkfile`(1M) manual page for more information.

`-a`	Creates a symbolic link with an absolute path name for a specified file. Use the `-a` option when absolute path names are required. However, the default is to create a symbolic link with a relative path name.
`-e`	Extends a file by a specified amount to allow Oracle tablespace resizing. See "Extending a Quick I/O file" on page 76.
`-h`	Specifies the Oracle datafile header size. This option specifies a header that will be allocated in addition to the size specified because Oracle requires one additional database block for all its datafiles. If this option is used, the resulting file can be used as an Oracle datafile. When creating an Oracle datafile, the header size should be equal to the Oracle block size (as determined by the `DB_BLOCK_SIZE` parameter). If the header size is missing when the `-h` option is used, a 32K header will be allocated.
`-r`	Increases the file to a specified size to allow Oracle tablespace resizing. See "Extending a Quick I/O file" on page 76.
`-s`	Specifies the space to preallocate for a file in bytes, kilobytes, megabytes, gigabytes, or sectors (512 bytes) by adding a `k`, `K`, `m`, `M`, `g`, `G`, `s`, or `S` suffix. The default is bytes—you do not need to attach a suffix to specify the value in bytes. The size of the file that is preallocated is the total size of the file (including the header) rounded to the nearest multiple of the file system block size.

Warning: Exercise caution when using absolute path names. Extra steps may be required during database backup and restore procedures to preserve symbolic links. If you restore files to directories different from the original paths, you must change the symbolic links that use absolute path names to point to the new path names before the database is restarted.

To create a database file as a Quick I/O file using qiomkfile

1. Create a database file using the `qiomkfile` command:

   ```
   # /opt/VRTS/bin/qiomkfile -h headersize -s file_size
   /mount_point/filename
   ```

2. Change the owner and group permissions on the file:

   ```
   # chown oracle:dba .filename
   ```

   ```
   # chmod 660 .filename
   ```

3. Create tablespaces on this file using SQL*Plus statements.

   For example:

   ```
   $ sqlplus /nolog
   ```

   ```
   SQL> connect / as sysdba
   ```

   ```
   SQL> create tablespace ts1 datafile '/mount_point/filename.dbf'
   ```

   ```
   size 100M reuse;
   ```

   ```
   exit;
   ```

An example to show how to create a 100MB database file named `dbfile` on the VxFS file system `/db01` using a relative path name:

```
# /opt/VRTS/bin/qiomkfile -h 32k -s 100m /db01/dbfile
```

```
# ls -al
```

```
-rw-r--r--   1 oracle   dba   104890368   Oct 2 13:42    .dbfile
```

```
lrwxrwxrwx   1 oracle   dba          19   Oct 2 13:42    dbfile -> \
.dbfile::cdev:vxfs:
```

In the example, `qiomkfile` creates a regular file named `/db01/.dbfile`, which has the real space allocated. Then, `qiomkfile` creates a symbolic link named `/db01/dbfile`. This symbolic link is a relative link to the Quick I/O interface for `/db01/.dbfile`, that is, to the `.dbfile::cdev:vxfs:` file. The symbolic link allows `.dbfile` to be accessed by any database or application using its Quick I/O interface.

Preallocating space for Quick I/O files using the setext command

As an alternative to using the `qiomkfile` command, you can also use the VxFS `setext` command to preallocate space for database files.

Before preallocating space with `setext`, make sure the following conditions have been met:

Prerequisites	■ The `setext` command requires superuser (`root`) privileges.
Usage notes	■ You can use the `chown` command to change the owner and group permissions on the file after you create it. See the `setext` (1M) manual page for more information.

To create a Quick I/O database file using setext

1. Access the VxFS mount point and create a file:

```
# cd /mount_point
```

```
# touch .filename
```

2. Use the `setext` command to preallocate space for the file:

```
# /opt/VRTS/bin/setext -r size -f noreserve -f chgsize  \
.filename
```

3 Create a symbolic link to allow databases or applications access to the file using its Quick I/O interface:

```
# ln -s .filename::cdev:vxfs: filename
```

4 Change the owner and group permissions on the file:

```
# chown oracle:dba .filename
```

```
# chmod 660 .filename
```

An example to show how to access the mount point `/db01`, create a datafile, preallocate the space, and change the permissions:

```
# cd /db01
# touch .dbfile
# /opt/VRTS/bin/setext -r 100M -f noreserve -f chgsize .dbfile
# ln -s .dbfile::cdev:vxfs: dbfile
```

```
# chown oracle:dba .dbfile
```

```
# chmod 660 .dbfile
```

Accessing regular VxFS files as Quick I/O files

You can access regular VxFS files as Quick I/O files using the `::cdev:vxfs:` name extension.

While symbolic links are recommended because they provide easy file system management and location transparency of database files, the drawback of using symbolic links is that you must manage two sets of files (for instance, during database backup and restore).

Usage notes

- When possible, use relative path names instead of absolute path names when creating symbolic links to access regular files as Quick I/O files. Using relative path names prevents copies of the symbolic link from referring to the original file when the directory is copied. This is important if you are backing up or moving database files with a command that preserves the symbolic link.

  However, some applications require absolute path names. If a file is then relocated to another directory, you must change the symbolic link to use the new absolute path. Alternatively, you can put all the symbolic links in a directory separate from the data directories. For example, you can create a directory named `/database` and put all the symbolic links there, with the symbolic links pointing to absolute path names.

To access an existing regular file as a Quick I/O file on a VxFS file system

1. Access the VxFS file system mount point containing the regular files:

   ```
   $ cd /mount_point
   ```

2. Create the symbolic link:

   ```
   $ mv filename .filename
   $ ln -s .filename::cdev:vxfs: filename
   ```

   This example shows how to access the VxFS file `dbfile` as a Quick I/O file:

   ```
   $ cd /db01
   $ mv dbfile .dbfile
   $ ln -s .dbfile::cdev:vxfs: dbfile
   ```

   This example shows how to confirm the symbolic link was created:

   ```
   $ ls -lo .dbfile dbfile
   ```

   ```
   lrwxr-xr-x   1 oracle    18 Jul 22 06:01 dbfile ->
   .dbfile::cdev:vxfs:
   lrwxr-xr-x   1 oracle     22 Jul 22 06:27  dbfile.dbf ->
   .dbfile.dbf::cdev:vxfs:
   ```

   ```
   lrwxrwxrwx  1 oracle      19     Oct 2 13:42   dbfile ->
   .dbfile::cdev:vxfs:
   ```

Converting Oracle files to Quick I/O files

Special commands, available in the `/opt/VRTSdbed/bin` directory, are provided to assist you in converting an existing database to use Quick I/O. You can use the `qio_getdbfiles` command to extract a list of file names from the database system tables and the `qio_convertdbfiles` command to convert this list of database files to use Quick I/O.

Note: It is recommended that you create a Storage Checkpoint before converting to or from Quick I/O.

See "Creating Storage Checkpoints using dbed_ckptcreate" on page 282.

Before converting database files to Ouick I/O files, the following conditions must be met:

Prerequisites	■ Log in as the Database Administrator (typically, the user ID `oracle`) to run the `qio_getdbfiles` and `qio_convertdbfiles` commands. ■ You must predefine the Oracle environment variable `$ORACLE_SID`. Change to the ORACLE_SID environment variable must be defined. ■ Files you want to convert must be regular files on VxFS file systems or links that point to regular VxFS files

Usage notes

- Converting existing database files to Quick I/O files may not be the best choice if the files are fragmented. Use of the `-f` option to determine the fragmentation levels is not supported for 5.1.
- If you choose to create new files, they will be contiguous. You must then move data from the old files to the new files using the `dd`(1M) command or a database import facility, and then define the new files to the database.
- By default, `qio_getdbfiles` skips any tablespaces marked `TEMPORARY`. Tablespaces marked `TEMPORARY` can be sparse, which means that not all blocks in the file are allocated. Quick I/O files cannot be sparse, as Quick I/O provides a raw type interface to storage. If a sparse file is converted to a Quick I/O file, the Oracle instance can fail if Oracle attempts to write into one of these unallocated blocks.

  See "Handling Oracle temporary tablespaces and Quick I/O" on page 73.
- You may also want to consider creating Quick I/O files for temporary tablespaces.

  See "Creating database files as Quick I/O files using qiomkfile" on page 62.
- The `qio_convertdbfiles` command exits and prints an error message if any of the database files are not on a VxFS file system. If this happens, you must remove any non-VxFS files from the `mkqio.dat` file before running the `qio_convertdbfiles` command.
- Instead of using the `qio_getdbfiles` command, you can manually create the `mkqio.dat` file containing the Oracle database filenames that you want to convert to Quick I/O files.

The following options are available for the `qio_getdbfiles` command:

`-a`	Lets you include all datafiles, including those that are potentially sparse. (Use this option only for debugging purposes, as sparse files are not candidates for use with Quick I/O.)
`-T`	Lets you specify the type of database as `ora`.

The following options are available for the `qio_convertdbfiles` command:

`-a`	Changes regular files to Quick I/O files using absolute path names. Use this option when symbolic links need to point to absolute path names (for example, at a site that uses SAP).

-h	Displays a help message. Creates the extra links for all datafiles and log files in the `/dev` directory to support SAP's `brbackup`.
-T	Lets you specify the type of database as `ora`.
-u	Changes Quick I/O files back to regular files. Use this option to undo changes made by a previous run of the `qio_convertdbfiles` script.

To extract a list of Oracle files to convert

- With the database instance up and running, run the `qio_getdbfiles` command from a directory for which you have write permission:

```
$ cd /extract_directory
```

```
$ /opt/VRTSdbed/bin/qio_getdbfiles -T ora
```

The `qio_getdbfiles` command extracts the list file names from the database system tables and stores the file names and their size in bytes in a file called `mkqio.dat` under the current directory.

Note: Alternatively, you can manually create the `mkqio.dat` file containing the Oracle database file names that you want to convert to use Quick I/O. You can also manually edit the `mkqio.dat` file generated by `qio_getdbfiles`, and remove files that you do not want to convert to Quick I/O files.

Note: To run the `qio_getdbfiles` command, you must have permission to access the database and permission to write to the `/extract_directory`.

The `mkqio.dat` list file should look similar to the following:

```
/data11r1/VRTS11r1/redo01.log 52428800
/data11r1/VRTS11r1/redo02.log 52428800
/data11r1/VRTS11r1/redo03.log 52428800
/data11r1/VRTS11r1/sysaux01.dbf 632553472
/data11r1/VRTS11r1/system01.dbf 754974720
/data11r1/VRTS11r1/undotbs01.dbf 47185920
/data11r1/VRTS11r1/users01.dbf 5242880
/data11r1/nqio1.dbf 104857600
```

To convert the Oracle database files to Quick I/O files

1. Shut down the database.
2. Run the `qio_convertdbfiles` command from the directory containing the `mkqio.dat` file:

   ```
   $ cd /extract_directory
   ```

   ```
   $ /opt/VRTSdbed/bin/qio_convertdbfiles
   ```

   The list of files in the `mkqio.dat` file is displayed. For example:

   ```
   file1 --> .file1::cdev:vxfs:
   file2 --> .file2::cdev:vxfs:
   file3 --> .file3::cdev:vxfs:
   file4 --> .file4::cdev:vxfs:
   file5 --> .file5::cdev:vxfs:
   ```

   Run the `qio_convertdbfiles` command (with no options specified) to rename the file *filename* to `.filename` and creates a symbolic link to `.filename` with the Quick I/O extension. By default, the symbolic link uses a relative path name.

   The `qio_convertdbfiles` script exits and prints an error message if any of the database files are not on a VxFS file system. If this happens, you must remove any non-VxFS files from the `mkqio.dat` file before running the `qio_convertdbfiles` command again.
3. Start up the database.
4. You can now access these database files using the Quick I/O interface.

To undo the previous run of qio_convertdbfiles and change Quick I/O files back to regular VxFS files

1. If the database is running, shut it down.
2. Run the following command from the directory containing the `mkqio.dat` file:

   ```
   $ cd /extract_directory
   ```

   ```
   $ /opt/VRTSdbed/bin/qio_convertdbfiles -u
   ```

   The list of Quick I/O files in the `mkqio.dat` file is displayed. For example:

   ```
   .file1::cdev:vxfs: --> file1
   .file2::cdev:vxfs: --> file2
   .file3::cdev:vxfs: --> file3
   .file4::cdev:vxfs: --> file4
   .file5::cdev:vxfs: --> file5
   ```

   The `qio_convertdbfiles` command with the undo option (`-u`) specified renames the files from <.filename> to <filename> and undoes the symbolic link to `.filename` that was created along with the Quick I/O files.

About sparse files

Support for sparse files lets applications store information (in inodes) to identify data blocks that have only zeroes, so that only blocks containing non-zero data have to be allocated on disk.

For example, if a file is 10KB, it typically means that there are blocks on disk covering the whole 10KB. Assume that you always want the first 9K to be zeroes. The application can go to an offset of 9KB and write 1KB worth of data. Only a block for the 1KB that was written is allocated, but the size of the file is still 10KB.

The file is now sparse. It has a hole from offset 0 to 9KB. If the application reads any part of the file within this range, it will see a string of zeroes.

If the application subsequently writes a 1KB block to the file from an offset of 4KB, for example, the file system will allocate another block.

The file then looks like:

- 0-4KB - hole
- 4-5KB - data block
- 5-9KB - hole
- 9-10KB - data block

So a 1TB file system can potentially store up to 2TB worth of files if there are sufficient blocks containing zeroes. Quick I/O files cannot be sparse and will always have all blocks specified allocated to them.

Handling Oracle temporary tablespaces and Quick I/O

You can create a new temporary tablespace using Quick I/O files. However, you cannot convert existing temporary tablespaces which use regular files to Quick I/O with the `qio_getdbfiles` command on Oracle9.

By default, `qio_getdbfiles` skips any tablespaces marked `TEMPORARY` because they can be sparse, which means that not all blocks in the file are allocated. Quick I/O files cannot be sparse, as Quick I/O provides a raw-type interface to storage. If a sparse file is converted to a Quick I/O file, the Oracle instance can fail if Oracle attempts to write into one of these unallocated blocks. When you initially create a temporary tablespace on Quick I/O files, however, Oracle sees them as raw devices and does not create sparse files.

To convert a temporary tablespace using regular files to Quick I/O files, you can drop your existing temporary tablespaces which use regular files and recreate them using Quick I/O files. You can also leave the temporary tablespaces as regular files.

To obtain a list of file names that are not temporary

◆ Use the following SQL statements:

```
$ sqlplus /nolog
 SQL> connect / as sysdba;
 SQL> select file_name from dba_data_files a,
 dba_tablespaces b where a.tablespace_name =
 b.tablespace_name and b.contents <> 'TEMPORARY';
```

To drop an existing temporary tablespace and recreate using Quick I/O files

1. Drop the temporary tablespace, including its contents:

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> drop tablespace tablespace_name including contents;
```

2. Create a Quick I/O file on a VxFS file system:

```
# /opt/VRTS/bin/qiomkfile -h header_size -s size \
/mount_point/filename.dbf
```

3. Change the owner and group permissions on the file

```
# chown oracle:dba .filename
```

```
# chmod 660 .filename
```

4. Create a new temporary tablespace using the `create temporary tablespace` command.

   To use the `create temporary tablespace` command:

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> create temporary tablespace tablespace_name \
tempfile '/mount_point/new_filename.dbf' size size reuse;
```

This example shows how to drop tablespace `tempts`, create a Quick I/O file `temp01.dbf`, change permissions and then create a new temporary tablespace `tempts` using the `create temporary tablespace` command:

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> drop tablespace tempts including contents;
Tablespace dropped.
# /opt/VRTS/bin/qiomkfile -h 32k -s 100M /db01/temp01.dbf
```

```
# chown oracle:dba .temp01
# chmod 660 .temp01
```

```
$ sqlplus /nolog
SQL> connect / as dba;
SQL> create temporary tablespace tempts \
tempfile '/db01/temp01.dbf' \
```

```
size 100M reuse;
Tablespace created.
```

Displaying Quick I/O status and file attributes

You can obtain and display information about Quick I/O status and file attributes using various options of the `ls` command:

-al	Lists all files on a file system, including Quick I/O files and their links.
-1L	Shows if Quick I/O was successfully installed and enabled.
-a1L	Shows how a Quick I/O file name is resolved to that of a raw device.

To list all files on the current file system, including Quick I/O files and their links

◆ Use the `ls -al` command with the file names:

```
$ ls -al filename .filename
```

The following example shows how to use the `-a` option to display the absolute path name created using `qiomkfile`:

```
$ ls -al d* .d*

lrwxr-xr-x  1 root  sys  18 Jul 22 06:00 .dbfile -
> .dbfile::cdev:vxfs:
-rw-r-----   1 oracle   dba     104865792 Jul 22 06:21 .dbfile.dbf
lrwxr-xr-x   1 oracle   dba     18 Jul 22 06:01 dbfile -
> .dbfile::cdev:vxfs:
lrwxr-xr-x   1 oracle   dba     22 Jul 22 06:27 dbfile.dbf -
> .dbfile.dbf::cdev:vxfs:
```

To determine if a datafile has been converted to Quick I/O

◆ Use the `ls` command as follows:

```
$ ls -lL filename
```

The following example shows how to determine if Quick I/O is installed and enabled:

```
$ ls -lL dbfile

crw-r--r--  1 oracle  dba   45, 1        Oct 2 13:42  dbfile
```

To show a Quick I/O file resolved to a raw device

◆ Use the `ls` command with the file names as follows:

```
$ ls -alL filename .filename
```

The following example shows how the Quick I/O file name `dbfile` is resolved to that of a raw device:

```
$ ls -alL d* .d*

crw-r--r--   1 oracle   dba      45,  1        Oct 2 13:42    dbfile

-rw-r-----   1 oracle   dba     104865792  Jul 22 06:21 .dbfile.dbf
crw-r-----   1 oracle   dba      145 0x000001 Jul 22 06:21 dbfile.dbf
```

Extending a Quick I/O file

Although Quick I/O files must be preallocated, they are not limited to the preallocated sizes. You can grow or "extend" a Quick I/O file by a specific amount or to a specific size, using options to the `qiomkfile` command. Extending Quick I/O files is a fast, online operation and offers a significant advantage over using raw devices.

Before extending a Quick I/O file, make sure the following conditions have been met:

Prerequisites	■ You must have sufficient space on the file system to extend the Quick I/O file.
Usage notes	■ You can also grow VxFS file systems online (provided the underlying disk or volume can be extended) using the `fsadm` command. You can expand the underlying volume and the filesystem with the `vxresize` command. ■ You must have superuser (`root`) privileges to resize VxFS file systems using the `fsadm` command. ■ See the `fsadm_vxfs` (1M) and `qiomkfile` (1M) manual pages for more information.

The following options are available with the `qiomkfile` command:

`-e`	Extends the file by a specified amount to allow Oracle tablespace resizing.
`-r`	Increases the file to a specified size to allow Oracle tablespace resizing.

To extend a Quick I/O file

1 If required, ensure the underlying storage device is large enough to contain a larger VxFS file system (see the `vxassist`(1M) manual page for more information), and resize the VxFS file system using `fsadm` command:

where:

- `-b` is the option for changing size
- <newsize>is the new size of the file system in bytes, kilobytes, megabytes, blocks, or sectors
- <mount_point>is the file system's mount point

2 Extend the Quick I/O file using the `qiomkfile` command:

```
$ /opt/VRTS/bin/qiomkfile -e extend_amount /mount_point/filename
```

or

```
$ /opt/VRTS/bin/qiomkfile -r newsize /mount_point/filename
```

An example to show how to grow VxFS file system `/db01` to 500MB and extend the `emp.dbf` Quick I/O file by 20MB:

```
# /opt/VRTS/bin/fsadm -b 500M /db01
```

```
$ /opt/VRTS/bin/qiomkfile -e 20M /db01/emp.dbf
```

An example to show how to grow VxFS file system `/db01` to 500MB and resize the `emp.dbf` Quick I/O file to 300MB:

```
# /opt/VRTS/bin/fsadm -b 500M /db01
```

```
$ /opt/VRTS/bin/qiomkfile -r 300M /db01/emp.dbf
```

Using Oracle's AUTOEXTEND with Quick I/O files

Oracle supports an automatic extend feature that automatically grows a database file by a prespecified amount, up to a prespecified maximum size.

For regular file system files, `AUTOEXTEND` works transparently, provided the underlying file system has enough space. For example, suppose the current size of a database file `emp.dbf` is 100MB, but this file is expected to triple in size over time. To accommodate this growth using `AUTOEXTEND` feature, you can specify the `next` size at 20MB and `maxsize` at 300MB. This will automatically grow the file by 20MB until its size reaches 300MB. For example:

```
alter database datafile 'emp.dbf' autoextend on next 20m \
maxsize 300m;
```

(See the Oracle Server SQL Reference Guide for more information about the `alter database` command, as well as the `next` and `maxsize` parameters.)

Note: You must have sufficient space on the underlying file system to `AUTOEXTEND` a file, and the underlying storage device must be large enough to contain the new, larger file system.

For Quick I/O files or raw devices, `AUTOEXTEND` does not know how to grow the underlying Quick I/O files or devices. Therefore, the Quick I/O file size must be large enough to accommodate the new size before `AUTOEXTEND` can grow the datafile.

You can use AUTOEXTEND with Quick I/O files in the following ways:

- Preallocate the Quick I/O file to a size at least as big as the maximum growth size expected for this database file.
  Using this method, you would need to preallocate the Quick I/O file `emp.dbf` for the entire 300MB. The drawback is that this can unnecessarily lock up excess disk space. Raw devices have a similar requirement.
- Monitor the free space available in the Quick I/O file, and grow the file as necessary with the `qiomkfile` command.
  Unlike raw devices, you can easily extend Quick I/O files online. Using this method, you can monitor the free space available in the Oracle datafiles and use the `qiomkfile` command to grow the Quick I/O files online as and when needed (typically when the file is about 80 to 90 percent full). This method does not require you to lock out unused disk space for Quick I/O files. The free space on the file system is available for use by other applications.

The following options are available for the `qiomkfile` command:

`-e`	Extends the file by a specified amount to allow Oracle tablespace resizing.
`-r`	Increases the file to a specified size to allow Oracle tablespace resizing.

You can grow underlying VxFS file systems online (provided the underlying disk or volume can be extended) using the `fsadm` command. See the `fsadm_vxfs`(1M) manual page for more information.

To monitor the free space available in an Oracle tablespace

- Check the free space currently available in the Oracle tablespace using the following Oracle SQL command:

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> select * from dba_free_space where \
tablespace_name = ’tablespace_name’;
SQL> exit
```

To extend a Quick I/O file using qiomkfile

- If the datafile is running low on free blocks, use the `qiomkfile` command to extend the Quick I/O file:

```
$ /opt/VRTS/bin/qiomkfile -e extend_amount \
  /mount_point/filename
```

  The following example shows how to monitor the free space on the tablespace `EMP` on file system `/db01`:

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> select * from dba_free_space where tablespace_name = ’EMP’;
SQL> exit
```

  The following example shows how to extend the Oracle datafile `emp.dbf` by 20MB (the specified `next` size) using the `qiomkfile` command:

```
$ /opt/VRTS/bin/qiomkfile -e 20M /db01/emp.dbf
```

Recreating Quick I/O files after restoring a database

If you need to restore your database and were using Quick I/O files, you can use the `qio_recreate` command to automatically recreate the Quick I/O files after you have performed a full database recovery. The `qio_recreate` command uses the `mkqio.dat` file, which contains a list of the Quick I/O files used by the database and the file sizes.

For information on recovering your database, refer to the documentation that came with your database software.

Before recreating Quick I/O with the `qio_recreate` command, make sure the following conditions have been met:

Prerequisites	■ Recover your database before attempting to recreate the Quick I/O files. ■ You may be logged in as either the database administrator or `root` to run the `qio_recreate` command. ■ In the directory from which you run the `qio_recreate` command, you must have an existing `mkqio.dat` file. ■ The ORACLE_SID environment variable must be set. See "Converting Oracle files to Quick I/O files" on page 68.
Usage notes	■ The `qio_recreate` command supports only conventional Quick I/O files. ■ Refer to the `qio_recreate`(1M) manual page for more information.

To recreate Quick I/O files after recovering a database

◆ As Oracle DBA, use the `qio_recreate` command as follows:

```
$ /opt/VRTSdbed/bin/qio_recreate -T ora
```

You will not see any output if the command is successful.

When you run the `qio_recreate` command, the following actions occur:

If...	Then...
a Quick I/O file is missing	the Quick I/O file is recreated.
a symbolic link from a regular VxFS file to a Quick I/O file is missing	the symbolic link is recreated.
a symbolic link and its associated Quick I/O file are missing	both the link and the Quick I/O file are recreated.
a Quick I/O file is missing and the regular VxFS file that it is symbolically linked to is not the original VxFS file	the Quick I/O file is not recreated and a warning message is displayed.
a Quick I/O file is smaller than the size listed in the `mkqio.dat` file	the Quick I/O file is not recreated and a warning message is displayed.

Disabling Quick I/O

If you need to disable the Quick I/O feature, you first need to convert any Quick I/O files back to regular VxFS files. Then, remount the VxFS file system using a special mount option.

Before disabling Quick I/O, make sure the following condition has been met:

Prerequisite	The file system you are planning to remount must be located in the `/etc/vfstab` file.

To disable Quick I/O

1. If the database is running, shut it down.
2. To change Quick I/O files back to regular VxFS files, run the following command from the directory containing the `mkqio.dat` list:

   ```
   $ /opt/VRTSdbed/bin/qio_convertdbfiles -u
   ```

   The list of Quick I/O files in the `mkqio.dat` file is displayed. For example:

   ```
   .file1::cdev:vxfs: --> file1
   .file2::cdev:vxfs: --> file2
   .file3::cdev:vxfs: --> file3
   .file4::cdev:vxfs: --> file4
   .file5::cdev:vxfs: --> file5
   ```

   The `qio_convertdbfiles` command with the undo option (`-u`) renames the files from .filename to filename and removes the symbolic link to .filename that was created along with the Quick I/O files.
3. To remount the file system with Quick I/O disabled, use the `mount -o noqio` command as follows:

   ```
   # /opt/VRTS/bin/mount -F vxfs -o remount,noqio /mount_point
   ```

Creating Quick I/O files in Solaris local zone

Quick I/O files cannot be created in the Solaris local zones. This is because the VxFS command `qiomkfile` cannot run in the Solaris local zone.

You must use the following workaround procedure to create Quick I/O files in Solaris local zone. You must perform the following commands as **root** unless otherwise stated.

To create Quick I/O files in Solaris local zone

1. Install `VRTSodm` on the Solaris global zone.

   You have to perform this step once on your system. You do not have to repeat it for each local zone.

2. Install your Storage Foundation product license on the Solaris global zone.

   You need to perform this step once on your system. You do not need to repeat it for each local zone.

3. Export the license from the Solaris global zone to each Solaris local zone where you plan to use Quick I/O.

   To do so, enter the following sequence of commands:

   ```
   zonecfg -z <zone_name>
   ```

   For example, if the Solaris local zone name is **zone2**, then:

   ```
   # zonecfg -z zone2
     zonecfg:zone2> add fs
     zonecfg:zone2:fs> set dir=/etc/vx/licenses/lic
     zonecfg:zone2:fs> set special=/etc/vx/licenses/lic
     zonecfg:zone2:fs> set type=lofs
     zonecfg:zone2:fs> end
     zonecfg:zone2> verify
     zonecfg:zone2> commit
     zonecfg:zone2> exit
   ```

4 In each local zone, mount the **odm** device to `/dev/odm`.

To do so, use the following command in the Solaris global zone:

```
# mkdir /zones/<zone_name>/dev/odm
```

Then use the following command in the Solaris local zone:

```
# mount -F odm /dev/odm /dev/odm
```

For example, if the Solaris local zone name is **zone2**, you must use the following commands:

```
# mkdir /zones/zone2/dev/odm
# mount -F odm /dev/odm /dev/odm
```

5 As Oracle user, create a Quick I/O file in a Solaris local zone utilizing `/opt/VRTS/bin/odmmkfile`. Rename the file to a Quick I/O file.

To do so, enter the following sequence of commands:

```
# su - <oracle_dba>
# cd <target_dir>
# /opt/VRTS/bin/odmmkfile -s <size> filename
# mv filename .filename
# ln -s .filename::cdev:vxfs: filename
# chown <Oracle_dba> .filename
```

Chapter 5

Using Veritas Cached Quick I/O

This chapter includes the following topics:

About Cached Quick I/O

Veritas Cached Quick I/O maintains and extends the database performance benefits of Veritas Quick I/O by making more efficient use of large, unused system memory through a selective buffering mechanism. Cached Quick I/O also supports features that support buffering behavior, such as file system read-ahead.

How Cached Quick I/O works

Cached Quick I/O is a specialized external caching mechanism specifically suitable to 32-bit ports of the Oracle server. Cached Quick I/O can be used on 64-bit ports of the Oracle server, but the benefits are not as great. Cached Quick I/O can be selectively applied to datafiles that are suffering an undesirable amount of physical disk I/O due to insufficient Oracle System Global Area (SGA). Cached Quick I/O works by taking advantage of the available physical memory that is left over after the operating system reserves the amount it needs and the Oracle SGA disk block buffers cache has been sized to the maximum capacity allowed within a 32-bit virtual address space. This extra memory serves as a cache to store file data, effectively serving as a second-level cache backing the SGA.

For example, consider a system configured with 12GB of physical memory, an operating system using 1GB, and a total Oracle size of 3.5GB. Unless you have other applications running on your system, the remaining 7.5GB of memory is unused. If you enable Cached Quick I/O, these remaining 7.5GB become available for caching database files.

Note: You cannot allocate specific amounts of the available memory to Cached Quick I/O. When enabled, Cached Quick I/O takes advantage of available memory.

Cached Quick I/O is not beneficial for all files in a database. Turning on caching for all database files can degrade performance due to extra memory management overhead (double buffer copying). You must use file I/O statistics to determine which individual database files benefit from caching, and then enable or disable Cached Quick I/O for individual files.

If you understand the applications that generate load on your database and how this load changes at different times during the day, you can use Cached Quick I/O to maximize performance. By enabling or disabling Cached Quick I/O on a per-file basis at different times during the day, you are using Cached Quick I/O to dynamically tune the performance of a database.

For example, files that store historical data are not generally used during normal business hours in a transaction processing environment. Reports that make use of this historical data are generally run during off-peak hours when interactive database use is at a minimum. During normal business hours, you can disable Cached Quick I/O for database files that store historical data in order to maximize memory available to other user applications. Then, during off-peak hours, you can enable Cached Quick I/O on the same files when they are used for report generation. This will provide extra memory resources to the database server without changing any database configuration parameters. Enabling file system read-ahead in this manner and buffering read data can provide great performance benefits, especially in large sequential scans.

You can automate the enabling and disabling of Cached Quick I/O on a per-file basis using scripts, allowing the same job that produces reports to tune the file system behavior and make the best use of system resources. You can specify different sets of files for different jobs to maximize file system and database performance.

How Cached Quick I/O improves database performance

Enabling Cached Quick I/O on suitable Quick I/O files improves database performance by using the file system buffer cache to store data. This data storage speeds up system reads by accessing the system buffer cache and avoiding disk I/O when searching for information.

Having data at the cache level improves database performance in the following ways:

- For read operations, Cached Quick I/O caches database blocks in the system buffer cache, which can reduce the number of physical I/O operations and therefore improve read performance.
- For write operations, Cached Quick I/O uses a direct-write, copy-behind technique to preserve its buffer copy of the data. After the direct I/O is scheduled and while it is waiting for the completion of the I/O, the file system updates its buffer to reflect the changed data being written out. For online transaction processing, Cached Quick I/O achieves better than raw device performance in database throughput on large platforms with very large physical memories.
- For sequential table scans, Cached Quick I/O can significantly reduce the query response time because of the read-ahead algorithm used by Veritas File System. If a user needs to read the same range in the file while the data is still in cache, the system is likely to return an immediate cache hit rather than scan for data on the disk.

How to set up Cached Quick I/O

To set up and use Cached Quick I/O, you should do the following in the order in which they are listed:

- Enable Cached Quick I/O on the underlying file systems used for your database.
- Exercise the system in your production environment to generate file I/O statistics.
- Collect the file I/O statistics while the files are in use.
- Analyze the file I/O statistics to determine which files benefit from Cached Quick I/O.
- Disable Cached Quick I/O on files that do not benefit from caching.

Enabling Cached Quick I/O on a file system

Cached Quick I/O depends on Veritas Quick I/O running as an underlying system enhancement in order to function correctly. Follow the procedures listed here to ensure that you have the correct setup to use Cached Quick I/O successfully.

Prerequisites

- You must have permission to change file system behavior using the `vxtunefs` command to enable or disable Cached Quick I/O. By default, you need superuser (`root`) permissions to run the `vxtunefs` command, but other system users do not. Superuser (`root`) must specifically grant database administrators permission to use this command as follows:

```
# chown root:dba /opt/VRTS/bin/vxtunefs
```

```
# chmod 4550 /opt/VRTS/bin/vxtunefs
```

  where users belonging to the `dba` group are granted permission to run the `vxtunefs` command. We recommend this selective, more secure approach for granting access to powerful commands.
- You must enable Quick I/O on the file system. Quick I/O is enabled automatically at file system mount time.

If you have correctly enabled Quick I/O on your system, you can proceed to enable Cached Quick I/O as follows:

- Set the file system Cached Quick I/O flag, which enables Cached Quick I/O for all files in the file system.
- Setting the file system Cached Quick I/O flag enables caching for all files in the file system. You must disable Cached Quick I/O on individual Quick I/O files that do not benefit from caching to avoid consuming memory unnecessarily. This final task occurs at the end of the enabling process.

Usage notes

- Do not enable Cached Quick I/O if Oracle is using Oracle Disk Manager.

Enabling and disabling the qio_cache_enable flag

As superuser (`root`), set the `qio_cache_enable` flag using the `vxtunefs` command after you mount the file system.

To enable the qio_cache_enable flag for a file system

◆ Use the `vxtunefs` command as follows:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=1 /mount_point
```

For example:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=1 /db02
```

where `/db02` is a VxFS file system containing the Quick I/O files and setting the `qio_cache_enable` flag to “1” enables Cached Quick I/O. This command enables caching for all the Quick I/O files on this file system.

To disable the flag on the same file system

◆ Use the `vxtunefs` command as follows:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=0 /mount_point
```

For example:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=0 /db02
```

where `/db02` is a VxFS file system containing the Quick I/O files and setting the `qio_cache_enable` flag to “0” disables Cached Quick I/O. This command disables caching for all the Quick I/O files on this file system.

Making Cached Quick I/O settings persistent across reboots and mounts

You can make the Cached Quick I/O system setting persistent across reboots and mounts by adding a file system entry in the `/etc/vx/tunefstab` file.

Note: The `tunefstab` file is a user-created file. For information on how to create the file and add tuning parameters, see the `tunefstab` (4) manual page.

To enable a file system after rebooting

◆ Put the file system in the `/etc/vx/tunefstab` file and set the flag entry:

```
/dev/vx/dsk/dgname/volname qio_cache_enable=1
```

where:

- `/dev/vx/dsk/dgname/volname` is the name of a block device
- *dgname* is the name of the disk group
- *volname* is the name of the volume

For example:

```
/dev/vx/dsk/PRODdg/db01 qio_cache_enable=1
/dev/vx/dsk/PRODdg/db02 qio_cache_enable=1
```

where `/dev/vx/dsk/PRODdg/db01` is the block device on which the file system resides.

The `tunefstab` (4) manual pages contain information on how to add tuning parameters.

See the `tunefstab` (4) manual page.

Note: vxtunefs can specify a mount point or a block device; tunefstab must always specify a block device only.

Using vxtunefs to obtain tuning information

Check the setting of the qio_cache_enable flag for each file system using the vxtunefs command.

To obtain information on only the qio_cache_enable flag setting

- Use the grep command with vxtunefs:

```
# /opt/VRTS/bin/vxtunefs /mount_point | grep qio_cache_enable
```

For example:

```
# /opt/VRTS/bin/vxtunefs /db01 | grep qio_cache_enable
```

where /db01 is the name of the file system. This command displays only the qio_cache_enable setting as follows:

```
qio_cache_enable = 0
```

You can also use the vxtunefs command to obtain a more complete list of I/O characteristics and tuning statistics.

See the vxtunefs (1) manual page.

To obtain information on all vxtunefs system parameters

- Use the vxtunefs command without grep:

```
# /opt/VRTS/bin/vxtunefs /mount_point
```

For example:

```
# /opt/VRTS/bin/vxtunefs /db01
```

The vxtunefs command displays output similar to the following:

```
 Filesystem i/o parameters for /db01
read_pref_io = 2097152
read_nstream = 1
read_unit_io = 2097152
write_pref_io = 2097152
write_nstream = 1
write_unit_io = 2097152
```

```
pref_strength = 10
buf_breakup_size = 2097152
discovered_direct_iosz = 262144
max_direct_iosz = 1048576
default_indir_size = 8192
qio_cache_enable = 1
write_throttle = 0
max_diskq = 33554432
initial_extent_size = 8
max_seqio_extent_size = 2048
max_buf_data_size = 8192
hsm_write_prealloc = 0
read_ahead = 1
inode_aging_size = 0
inode_aging_count = 0
fcl_maxalloc = 222425088
fcl_keeptime = 0
fcl_winterval = 3600
fcl_ointerval = 600
oltp_load = 0
```

The `vxtunefs`(1) manual pages contain a complete description of `vxtunefs` parameters and the tuning instructions.

See the `vxtunefs`(1) manual page.

Determining candidates for Cached Quick I/O

Determining which files can benefit from Cached Quick I/O is an iterative process that varies with each application. For this reason, you may need to complete the following steps more than once to determine the best possible candidates for Cached Quick I/O.

Before determining candidate files for Quick I/O, make sure the following conditions have been met:

Prerequisites	■ You must enable Cached Quick I/O for the file systems. See "Enabling Cached Quick I/O on a file system" on page 87.
Usage notes	■ See the `qiostat` (1M) manual page for more information.

Collecting I/O statistics

Once you have enabled Cached Quick I/O on a file system, you need to collect statistics to determine and designate the files that can best take advantage of its benefits.

To collect statistics needed to determine files that benefit from Cached Quick I/O

1. Reset the `qiostat` counters by entering:

   ```
   $ /opt/VRTS/bin/qiostat -r /mount_point/filenames
   ```

2. Run the database under full normal load and through a complete cycle (24 to 48 hours in most cases) to determine your system I/O patterns and database traffic in different usage categories (for example, OLTP, reports, and backups) at different times of the day.
3. While the database is running, run `qiostat -l` to report the caching statistics as follows:

   ```
   $ /opt/VRTS/bin/qiostat -l /mount_point/filenames
   ```

   or, use the `-i` option to see statistic reports at specified intervals:

   ```
   $ /opt/VRTS/bin/qiostat -i n /mount_point/filenames
   ```

   where n is time in seconds

   For example:

   To collect I/O statistics from all database files on file system `/db01`:

   ```
   $ /opt/VRTS/bin/qiostat -l /db01/*.dbf
   ```

About I/O statistics

The output of the `qiostat` command is the primary source of information to use in deciding whether to enable or disable Cached Quick I/O on specific files. Statistics are printed in two lines per object.

The second line of information is defined as follows:

- `CREAD` is the number of reads from the VxFS cache (or total number of reads to Quick I/O files with cache advisory on)
- `PREAD` is the number of reads going to the disk for Quick I/O files with the cache advisory on

- `HIT RATIO` is displayed as a percentage and is the number of `CREADS` minus the number of `PREADS` times 100 divided by the total number of `CREADS`. The formula looks like this:

```
(CREADs - PREADs) * 100/ CREADs
```

The `qiostat -l` command output looks similar to the following:

```
                 OPERATIONS  FILE BLOCKS  AVG TIME(ms)
CACHE_STATISTICS

FILE NAME       READ  WRITE   READ WRITE    READ WRITE
CREAD  PREAD  HIT RATIO

/db01/cust.dbf   17128 9634   68509  38536   24.8  0.4
17124  15728  8.2

/db01/system.dbf   6     1      21    4      10.0  0.0
  6     6     0.0

/db01/stk.dbf    62552 38498  250213 153992  21.9  0.4
62567  49060  21.6

                 OPERATIONS  FILE BLOCKS  AVG TIME(ms)
CACHE_STATISTICS

FILE NAME       READ  WRITE   READ WRITE    READ WRITE
CREAD  PREAD  HIT RATIO

                 OPERATIONS  FILE BLOCKS  AVG TIME(ms)
CACHE_STATISTICS

FILE NAME       READ  WRITE   READ WRITE    READ WRITE
CREAD  PREAD  HIT RATIO
```

Analyze the output to find out where the cache-hit ratio is above a given threshold. A cache-hit ratio above 20 percent on a file for a given application may be sufficient to justify caching on that file. For systems with larger loads, the acceptable ratio may be 30 percent or above. Cache-hit-ratio thresholds vary according to the database type and load.

Using the sample output above as an example, the file `/db01/system.dbf` does not benefit from the caching because the cache-hit ratio is zero. In addition, the file receives very little I/O during the sampling duration.

However, the file `/db01/stk.dbf` has a cache-hit ratio of 21.6 percent. If you have determined that, for your system and load, this figure is above the acceptable threshold, it means the database can benefit from caching. Also, study the numbers reported for the read and write operations. When you compare the number of reads and writes for the /db01/stk.dbf file, you see that the number of reads is roughly twice the number of writes. You can achieve the greatest performance gains with Cached Quick I/O when using it for files that have higher read than write activity.

Based on these two factors, /db01/stk.dbf is a prime candidate for Cached Quick I/O.

See "Enabling and disabling Cached Quick I/O for individual files" on page 95.

Effects of read-aheads on I/O statistics

The number of CREADs in the qiostat output is the total number of reads performed, including Cached Quick I/O, and the number of PREADs is the number of physical reads. The difference between CREADs and PREADs (CREADS - PREADS) is the number of reads satisfied from the data in the file system cache. Thus, you expect that the number of PREADs would always be equal to or lower than the number of CREADs.

However, the PREADs counter also increases when the file system performs read-aheads. These read-aheads occur when the file system detects sequential reads. In isolated cases where cache hits are extremely low, the output from qiostat could show that the number of CREADs is lower than the number of PREADs. The cache-hit ratio calculated against these CREAD/PREAD values is misleading when used to determine whether Cached Quick I/O should be enabled or disabled.

Under these circumstances, you can make a more accurate decision based on a collective set of statistics by gathering multiple sets of data points. Consequently, you might want to enable Cached Quick I/O for all the data files in a given tablespace, even if just one of the files exhibited a high cache-hit ratio.

Other tools for analysis

While the output of the qiostat command is the primary source of information to use in deciding whether to enable Cached Quick I/O on specific files, we also recommend using other tools in conjunction with qiostat. For example, benchmarking software that measures database throughput is also helpful. If a benchmark test in which Cached Quick I/O was enabled for a certain set of data files resulted in improved performance, you can also use those results as the basis for enabling Cached Quick I/O.

Enabling and disabling Cached Quick I/O for individual files

After using `qiostat` or other analysis tools to determine the appropriate files for Cached Quick I/O, you need to disable Cached Quick I/O for those individual files that do not benefit from caching using the `qioadmin` command.

Prerequisites	■ Enable Cached Quick I/O for the file system before enabling or disabling Cached Quick I/O at the individual file level.
Usage notes	■ You can enable or disable Cached Quick I/O for individual files while the database is online. ■ You should monitor files regularly using `qiostat` to ensure that a file's cache-hit ratio has not changed enough to reconsider enabling or disabling Cached Quick I/O for the file. ■ Enabling or disabling Cached Quick I/O for an individual file is also referred to as setting the cache advisory on or off. ■ See the `qioadmin` (1) manual page.

Setting cache advisories for individual files

You can enable and disable Cached Quick I/O for individual files by changing the cache advisory settings for those files.

To disable Cached Quick I/O for an individual file

◆ Use the `qioadmin` command to set the cache advisory to `OFF` as follows:

```
$ /opt/VRTS/bin/qioadmin -S filename=OFF /mount_point
```

For example, to disable Cached Quick I/O for the file `/db01/system.dbf`, set the cache advisory to `OFF`:

```
$ /opt/VRTS/bin/qioadmin -S system.dbf=OFF /db01
```

To enable Cached Quick I/O for an individual file

- Use the `qioadmin` command to set the cache advisory to `ON` as follows:

```
$ /opt/VRTS/bin/qioadmin -S filename=ON /mount_point
```

For example, running `qiostat`shows the cache hit ratio for the file `/db01/system.dbf`reaches a level that would benefit from caching. To enable Cached Quick I/O for the file `/db01/system.dbf`, set the cache advisory to `ON`:

```
$ /opt/VRTS/bin/qioadmin -S system.dbf=ON /db01
```

Making individual file settings for Cached Quick I/O persistent

You can make the enable or disable individual file settings for Cached Quick I/O persistent across reboots and mounts by adding cache advisory entries in the `/etc/vx/qioadmin` file.

Cache advisories set using the `qioadmin` command are stored as extended attributes of the file in the inode. These settings persist across file system remounts and system reboots, but these attributes are not backed up by the usual backup methods, so they cannot be restored. Therefore, always be sure to reset cache advisories after each file restore. This is not necessary if you maintain the cache advisories for Quick I/O files in the `/etc/vx/qioadmin` file.

To enable or disable individual file settings for Cached Quick I/O automatically after a reboot or mount

- Add cache advisory entries in the `/etc/vx/qioadmin` file as follows:

```
device=/dev/vx/dsk/<diskgroup>/<volume>
```

filename,OFF

filename,OFF

filename,OFF

filename,ON

For example, to make the Cached Quick I/O settings for individual files in the `/db01` file system persistent, edit the `/etc/vx/qioadmin` file similar to the following:

```
#
# List of files to cache in /db01 file system
#
device=/dev/vx/dsk/PRODdg/db01

cust.dbf,OFF
system.dbf,OFF
stk.dbf,ON
```

Determining individual file settings for Cached Quick I/O using qioadmin

You can determine whether Cached Quick I/O is enabled or disabled for individual files by displaying the file's cache advisory setting using the `qioadmin` command.

Note: To verify caching, always check the setting of the flag `qio_cache_enable` using `vxtunefs`, along with the individual cache advisories for each file.

To display the current cache advisory settings for a file

- Use the `qioadmin` command with the `-P` option as follows:

  ```
  $ /opt/VRTS/bin/qioadmin -P filename /mount_point
  ```

  For example, to display the current cache advisory setting for the file `cust.dbf`in the `/db01`file system:

  ```
  $ /opt/VRTS/bin/qioadmin -P cust.dbf /db01
  ```

  ```
  cust.dbf,OFF
  ```

Chapter 6

Using Veritas Extension for Oracle Disk Manager

This chapter includes the following topics:

About Oracle Disk Manager

Veritas Extension for Oracle Disk Manager is specifically designed for Oracle10g or later to enhance file management and disk I/O throughput. The features of Oracle Disk Manager are best suited for databases that reside in a file system contained in Veritas File System. Oracle Disk Manager allows Oracle10g or later users to improve database throughput for I/O intensive workloads with special I/O optimization.

Veritas Extension for Oracle Disk Manager supports Oracle Resilvering. With Oracle Resilvering, the storage layer receives information from the Oracle database as to which regions or blocks of a mirrored datafile to resync after a system crash. Oracle Resilvering avoids overhead from the VxVM DRL, which increases performance.

Oracle Disk Manager reduces administrative overhead by providing enhanced support for Oracle Managed Files. Veritas Extension for Oracle Disk Manager has Quick I/O-like capabilities, but is transparent to the user. Unlike Veritas Quick I/O, files managed using Veritas Extension for Oracle Disk Manager do not require special file naming conventions. The Oracle Disk Manager interface uses regular database files. If you are upgrading to Oracle10g or later, you should convert from Quick I/O to Oracle Disk Manager.

Database administrators can choose the datafile type used with the Oracle product. Historically, choosing between file system files and raw devices was based on manageability and performance. The exception to this is a database intended for use with Oracle Parallel Server, which requires raw devices on most platforms. If performance is not as important as administrative ease, file system files are typically the preferred file type. However, while an application may not have substantial I/O requirements when it is first implemented, I/O requirements may change. If an application becomes dependent upon I/O throughput, converting datafiles from file system to raw devices is often necessary.

Oracle Disk Manager was designed to work with Oracle10g or later to provide both performance and manageability. Oracle Disk Manager provides support for Oracle's file management and I/O calls for database storage on VxFS file systems and on raw volumes or partitions. This feature is provided as a dynamically-loaded shared library with which Oracle binds when it is loaded. The Oracle Disk Manager library works with an Oracle Disk Manager driver that is loaded in the kernel to perform its functions.

If you are upgrading to Oracle10g or later, you should convert from Quick I/O to Oracle Disk Manager.

The benefits of using Oracle Disk Manager are as follows:

- True kernel asynchronous I/O for files and raw devices
- Reduced system call overhead
- Improved file system layout by preallocating contiguous files on a VxFS file system
- Performance on file system files that is equivalent to raw devices
- Transparent to users
- Contiguous datafile allocation

How Oracle Disk Manager improves database performance

Oracle Disk Manager improves database I/O performance to VxFS file systems by:

- Supporting kernel asynchronous I/O
- Supporting direct I/O and avoiding double buffering
- Avoiding kernel write locks on database files
- Supporting many concurrent I/Os in one system call
- Avoiding duplicate opening of files per Oracle instance
- Allocating contiguous datafiles

About kernel asynchronous I/O support

Asynchronous I/O performs non-blocking system level reads and writes, allowing the system to perform multiple I/O requests simultaneously. Kernel asynchronous I/O is better than library asynchronous I/O because the I/O is queued to the disk device drivers in the kernel, minimizing context switches to accomplish the work.

About direct I/O support and avoiding double buffering

I/O on files using read() and write() system calls typically results in data being copied twice: once between the user and kernel space, and the other between kernel space and the disk. In contrast, I/O on raw devices is copied directly between user space and disk, saving one level of copying. As with I/O on raw devices, Oracle Disk Manager I/O avoids the extra copying. Oracle Disk Manager bypasses the system cache and accesses the files with the same efficiency as raw devices. Avoiding double buffering reduces the memory overhead on the system. Eliminating the copies from kernel to user address space significantly reduces kernel mode processor utilization freeing more processor cycles to execute the application code.

About avoiding kernel write locks on database files

When database I/O is performed by way of the write() system call, each system call acquires and releases a kernel write lock on the file. This lock prevents simultaneous write operations on the same file. Because database systems usually implement their own locks for managing concurrent access to files, write locks unnecessarily serialize I/O writes. Oracle Disk Manager bypasses file system locking and lets the database server control data access.

About supporting many concurrent I/Os in one system call

When performing asynchronous I/O, an Oracle process may try to issue additional I/O requests while collecting completed I/Os, or it may try to wait for particular I/O requests synchronously, as it can do no other work until the I/O is completed. The Oracle process may also try to issue requests to different files. All this activity can be accomplished with one system call when Oracle uses the Oracle Disk Manager I/O interface. This interface reduces the number of system calls performed to accomplish the same work, reducing the number of user space/kernel space context switches.

About avoiding duplicate file opens

Oracle Disk Manager allows files to be opened once, providing a “file identifier.” This is called “identifying” the files. The same file identifiers can be used by any other processes in the Oracle instance. The file status is maintained by the Oracle Disk Manager driver in the kernel. The reduction in file open calls reduces processing overhead at process initialization and termination, and it reduces the number of file status structures required in the kernel.

About allocating contiguous datafiles

Oracle Disk Manager can improve performance for queries, such as sort and parallel queries, that use temporary tablespaces. Without Oracle Disk Manager, Oracle does not initialize the datafiles for the temporary tablespaces. Therefore, the datafiles become sparse files and are generally fragmented. Sparse or fragmented files lead to poor query performance. When using Oracle Disk Manager, the datafiles are initialized for the temporary tablespaces and are allocated in a contiguous fashion, so that they are not sparse.

About Oracle Disk Manager and Oracle Managed Files

Oracle10g or later offers a feature known as Oracle Managed Files (OMF). OMF manages datafile attributes such as file names, file location, storage attributes, and whether or not the file is in use by the database. OMF is only supported for databases that reside in file systems. OMF functionality is greatly enhanced by Oracle Disk Manager.

The main requirement for OMF is that the database be placed in file system files. There are additional prerequisites imposed upon the file system itself.

OMF is a file management feature that:

- Eliminates the task of providing unique file names

- Offers dynamic space management by way of the tablespace auto-extend functionality of Oracle10g or later

OMF should only be used in file systems that reside within striped logical volumes, which support dynamic file system growth. File systems intended for OMF use must also support large, extensible files in order to facilitate tablespace auto-extension. Raw partitions cannot be used for OMF.

By default, OMF datafiles are created with auto-extend capability. This attribute reduces capacity planning associated with maintaining existing databases and implementing new applications. Due to disk fragmentation that occurs as the tablespace grows over time, database administrators have been somewhat cautious when considering auto-extensible tablespaces. Oracle Disk Manager eliminates this concern.

When Oracle Disk Manager is used in conjunction with OMF, special care is given within Veritas Extension for Disk Manager to ensure that contiguous disk space is allocated to datafiles, including space allocated to a tablespace when it is auto-extended. The table and index scan throughput does not decay as the tablespace grows.

How Oracle Disk Manager works with Oracle Managed Files

The following example illustrates the relationship between Oracle Disk Manager and Oracle Managed Files (OMF). The example shows the `init.ora` contents and the command for starting the database instance. To simplify Oracle UNDO management, the new Oracle10g or later `init.ora` parameter `UNDO_MANAGEMENT` is set to `AUTO`. This is known as System-Managed Undo.

Note: Before building an OMF database, you need the appropriate `init.ora` default values. These values control the location of the `SYSTEM` tablespace, online redo logs, and control files after the `CREATE DATABASE` statement is executed.

```
$ cat initPROD.ora
UNDO_MANAGEMENT = AUTO
DB_CREATE_FILE_DEST = '/PROD'
DB_CREATE_ONLINE_LOG_DEST_1 = '/PROD'
db_block_size = 4096
db_name = PROD
$ sqlplus /nolog
SQL> connect / as sysdba
SQL> startup nomount pfile= initPROD.ora
```

The Oracle instance starts.

```
Total System Global Area 93094616 bytes
Fixed Size 279256 bytes
Variable Size 41943040 bytes
Database Buffers 50331648 bytes
Redo Buffers 540672 bytes
```

To implement a layout that places files associated with the `EMP_TABLE` tablespace in a directory separate from the `EMP_INDEX` tablespace, use the `ALTER SYSTEM` statement. This example shows how OMF handles file names and storage clauses and paths. The layout allows you to think of the tablespaces as objects in a file system as opposed to a collection of datafiles. Since OMF uses the Oracle Disk Manager file resize function, the tablespace files are initially created with the default size of 100MB and grow as needed. Use the `MAXSIZE` attribute to limit growth.

The following example shows the commands for creating an OMF database and for creating the `EMP_TABLE` and `EMP_INDEX` tablespaces in their own locale.

Note: The directory must exist for OMF to work, so the `SQL*Plus HOST` command is used to create the directories:

```
SQL> create database PROD;
```

The database is created.

```
SQL> HOST mkdir /PROD/EMP_TABLE;
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/PROD/EMP_TABLE';
```

The system is altered.

```
SQL> create tablespace EMP_TABLE DATAFILE AUTOEXTEND ON MAXSIZE \
500M;
```

A tablespace is created.

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/PROD/EMP_INDEX';
```

The system is altered.

```
SQL> create tablespace EMP_INDEX DATAFILE AUTOEXTEND ON MAXSIZE \
100M;
```

A tablespace is created.

Use the `ls` command to show the newly created database:

```
$ ls -lFR
total 638062
drwxr-xr-x 2 oracle10g dba 96 May  3 15:43 EMP_INDEX/
drwxr-xr-x 2 oracle10g dba 96 May  3 15:43 EMP_TABLE/
-rw-r--r-- 1 oracle10g dba 104858112 May 3 17:28 ora_1_BEhYgc0m.log
-rw-r--r-- 1 oracle10g dba 104858112 May 3 17:27 ora_2_BEhYu4NA.log
-rw-r--r-- 1 oracle10g dba 806912 May 3 15:43 ora_BEahlfUX.ctl
-rw-r--r-- 1 oracle10g dba 10489856 May 3 15:43 ora_sys_undo_BEajPSVq.dbf
-rw-r--r-- 1 oracle10g dba 104861696 May 3 15:4 ora_system_BEaiFE8v.dbf
-rw-r--r-- 1 oracle10g dba 186 May 3 15:03 PROD.ora

./EMP_INDEX:
total 204808
-rw-r--r-- 1 oracle10g dba 104861696 May 3 15:43
ora_emp_inde_BEakGfun.dbf

./EMP_TABLE:
total 204808
-rw-r--r-- 1 oracle10g dba 104861696 May 3 15:43
ora_emp_tabl_BEak1LqK.dbf
```

Setting up Veritas Extension for Oracle Disk Manager

Veritas Extension for Oracle Disk Manager is part of Veritas Storage Foundation Standard and Enterprise products. Veritas Extension for Oracle Disk Manager is enabled once your Veritas Storage Foundation Standard or Enterprise product and Oracle10g or later are installed. The Veritas Extension for Oracle Disk Manager library is linked to the library in the `{ORACLE_HOME}/lib` directory.

Before setting up Veritas Extension for Oracle Disk Manager, the following conditions must be met:

Prerequisites	■ A Veritas Storage Foundation Enterprise or Standard product must be installed on your system. ■ Oracle10g, or later, must be installed on your system. ■ If Cached Quick I/O is available, do not enable Oracle Disk Manager when Cached Quick I/O is enabled for datafiles.

Usage Notes

- When the Quick I/O feature is available, Oracle Disk Manager uses the Quick I/O driver to perform asynchronous I/O. Do not turn off the Quick I/O mount option, which is the default.
- Oracle uses default file access methods if Oracle10g or later or a Veritas Storage Foundation Standard or Enterprise product is not installed, or VxFS 5.0 or later is not available in the kernel.

Configuring Veritas Extension for Oracle Disk Manager

If `ORACLE_HOME` is on a shared file system, run the following commands from any node, otherwise run them on each node.

where *`ORACLE_HOME`* is the location where Oracle database binaries have been installed.

To configure Veritas Extension for Oracle Disk Manager

1. Log in as `oracle`.
2. If the Oracle database is running, then shutdown the Oracle database.
3. Verify that `/opt/VRTSodm/lib64/libodm.so` exists.
4. Link Oracle's ODM library present in `ORACLE_HOME` with Veritas Extension for Oracle Disk Manager library:

   For Oracle10g:

   - Change to the `$ORACLE_HOME/lib` directory, enter:

     ```
     # cd $ORACLE_HOME/lib
     ```

   - Take backup of `libodm10.so`, enter.

     ```
     # mv libodm10.so libodm10.so.oracle-`date '+%m_%d_%y-%H_%M_%S'`
     ```

   - Link `libodm10.so` with Veritas ODM library, enter:

     ```
     # ln -s /opt/VRTSodm/lib64/libodm.so libodm10.so
     ```

   For Oracle11g:

   - Change to the `$ORACLE_HOME/lib` directory, enter:

     ```
     # cd $ORACLE_HOME/lib
     ```

   - Take backup of `libodm11.so`, enter.

```
# mv libodm11.so libodm11.so.oracle-`date '+%m_%d_%y-%H_%M_%S'`
```

- Link `libodm11.so` with Veritas ODM library, enter:

```
# ln -s /opt/VRTSodm/lib64/libodm.so libodm11.so
```

5 Start the Oracle database.

6 To confirm that the Oracle database starts with Veritas Extension for ODM, the alert log will contain the following text:

```
Veritas <version> ODM Library
```

where *5.1.00.00* is the ODM library version shipped with the product.

How to prepare existing database storage for Oracle Disk Manager

Non-Quick I/O files in a VxFS file system work with Oracle Disk Manager without any changes. The files are found and identified for Oracle Disk Manager I/O by default. To take full advantage of Oracle Disk Manager datafiles, files should not be fragmented.

If you are using Quick I/O files in a VxFS file system and you want to move to Oracle Disk Manager, convert the Quick I/O files to normal files using the `qio_convertdbfiles -u` command.

You must be running Oracle10g or later to use Oracle Disk Manager.

Converting Quick I/O files to Oracle Disk Manager files

If you plan to run the Veritas product with Oracle10g or later, and you have been using Quick I/O files, Symantec recommends that you convert your Quick I/O files to regular files. This should be done after you upgrade.

Note: If you are running an earlier version of Oracle (Oracle 8.x or lower), you should not convert your Quick I/O files because Oracle Disk Manager is for Oracle10g or later only.

The Oracle Disk Manager uses the Quick I/O driver to perform asynchronous I/O, do not turn off the Quick I/O mount option, which is the default.

To convert Quick I/O files to Oracle Disk Manager files

1. As Oracle DBA, run `qio_getdbfiles` to retrieve a list of all datafiles.

   ```
   $ /opt/VRTS/bin/qio_getdbfiles -T ora -a
   ```

   The list is compiled in a file named `mkqio.dat`.

2. Shutdown the database.

3. As Oracle DBA, run `qio_convertdbfiles` in the directory containing the `mkqio.dat` file. The `qio_convertdbfiles` script converts all Quick I/O files to ODM files.

   ```
   $ /opt/VRTS/bin/qio_convertdbfiles -T ora -u
   ```

4. Restart the database instance.

Verifying that Oracle Disk Manager is configured

Before verifying that Oracle Disk Manager is configured, make sure that the following conditions are met:

Prerequisites	■ `/opt/VRTSodm/lib/libodm.so` must exist. ■ If you are using Oracle 10g, `$ORACLE_HOME/lib/libodm10.so` is linked to `/opt/VRTSodm/lib/sparcv9/libodm.so`. ■ If you are using Oracle 11g, `$ORACLE_HOME/lib/libodm11.so` is linked to `/opt/VRTSodm/lib/sparcv9/libodm.so`. ■ If you are using Oracle10g on Opteron Operating System, `$ORACLE_HOME/lib/libodm10.so` is linked to `/opt/VRTSodm/lib/amd64/libodm.so`.

To verify that Oracle Disk Manager is configured

1. Verify that the ODM feature is included in the license:

   ```
   # /opt/VRTS/bin/vxlicrep | grep ODM
   ```

   The output verifies that ODM is enabled.

   Note: Verify that the license key containing the ODM feature is not expired. If the license key has expired, you will not be able to use the ODM feature.

   ---

2. Check that the `VRTSodm` package is installed:

   ```
   # pkginfo VRTSodm
   system VRTSodm Veritas Oracle Disk Manager
   ```

3. Check that `libodm.so` is present.

   If you are running 32-bit Oracle9i, use the following command:

   ```
   # ls -lL /opt/VRTSodm/lib/libodm.so
   -rw-r--r-- 1 root sys 14336 Apr 25 18:42
   /opt/VRTSodm/lib/libodm.so
   ```

   If you are running 64-bit Oracle9i, use the following command:

   ```
   # ls -lL /opt/VRTSodm/lib/sparcv9/libodm.so
   -rw-r--r-- 1 root sys 14336 Apr 25 18:42
   /opt/VRTSodm/lib/sparcv9/libodm.so
   ```

To verify that Oracle Disk Manager is running

1. Start the Oracle database.
2. Check that the instance is using the Oracle Disk Manager function:

   ```
   # cat /dev/odm/stats
   # echo $?
   0
   ```

3 Verify that the Oracle Disk Manager is loaded:

```
# modinfo | grep ODM | grep VRTS
162 7b76c000  184a0  25   1  odm (VRTS ODM 5.1.10.00,REV=MP1u)
```

4 In the alert log, verify the Oracle instance is running. The log should contain output similar to the following:

```
Oracle instance running with ODM: Veritas 5.1.00.00 ODM Library,
Version 2.0
```

Disabling the Oracle Disk Manager feature

Since the Oracle Disk Manager feature uses regular files, you can access these files as regular VxFS files as soon as the feature is disabled.

The steps for disabling the Oracle Disk Manager feature are the same for both 32- and 64-bit Oracle10g.

Note: To convert to VxFS with Quick I/O, disable Oracle Disk Manager using the following procedure, then convert the files to Quick I/O files.

See "Converting Quick I/O files to Oracle Disk Manager files" on page 107.

Before disabling the Oracle Disk Manager feature, you may want to back up your files.

To disable the Oracle Disk Manager feature in an Oracle instance

1. Shut down the database instance.
2. Use the `rm` and `ln` commands to remove the link to the Oracle Disk Manager Library.

   For Oracle 11g, enter:

   ```
   # rm ${ORACLE_HOME}/lib/libodm11.so
   # ln -s ${ORACLE_HOME}/lib/libodmd11.so \
   ${ORACLE_HOME}/lib/libodm11.so
   ```

   For Oracle 10g, enter:

   ```
   # rm ${ORACLE_HOME}/lib/libodm10.so
   # ln -s ${ORACLE_HOME}/lib/libodmd10.so \
   ${ORACLE_HOME}/lib/libodm10.so
   ```

   For Oracle 10g on Opteron, enter:

   ```
   # rm ${ORACLE_HOME}/lib/libodm10.so
   # ln -s ${ORACLE_HOME}/lib/libodmd10.so \
   ${ORACLE_HOME}/lib/libodm10.so
   ```

3. Restart the database instance.

About Cached ODM

ODM I/O normally bypasses the file system cache and directly reads from and writes to disk. Cached ODM enables some I/O to use caching and read ahead, which can improve ODM I/O performance. Cached ODM performs a conditional form of caching that is based on per-I/O hints from Oracle. The hints indicate what Oracle does with the data. ODM uses these hints to perform caching and read ahead for some reads, but ODM avoids caching other reads, even for the same file.

You can enable cached ODM only for local mount files. Cached ODM does not affect the performance of files and file systems for which you did not enable caching.

See "Enabling Cached ODM for file systems" on page 112.

Cached ODM can be configured in two ways. The primary configuration method is to turn caching on or off for all I/O on a per-file basis. The secondary configuration method is to adjust the ODM cachemap. The cachemap maps file type and I/O type combinations into caching advisories.

See "Tuning Cached ODM settings for individual files" on page 112.

See "Tuning Cached ODM settings via the cachemap" on page 113.

Enabling Cached ODM for file systems

Cached ODM is initially disabled on a file system. You enable Cached ODM for a file system by setting the `odm_cache_enable` option of the `vxtunefs` command after the file system is mounted.

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

Note: The `vxtunefs` command enables conditional caching for all of the ODM files on the file system.

To enable Cached ODM for a file system

1. Enable Cached ODM on the VxFS file system `/database01`:

   ```
   # vxtunefs -s -o odm_cache_enable=1 /database01
   ```

2. Optionally, you can make this setting persistent across mounts by adding a file system entry in the file `/etc/vx/tunefstab`:

   ```
   /dev/vx/dsk/datadg/database01 odm_cache_enable=1
   ```

   See the `tunefstab`(4) manual page.

Tuning Cached ODM settings for individual files

You can use the `odmadm setcachefile` command to override the cachemap for a specific file so that ODM caches either all or none of the I/O to the file. The caching state can be ON, OFF, or DEF (default). The DEF caching state is conditional caching, meaning that for each I/O, ODM consults the cachemap and determines whether the specified file type and I/O type combination should be cached. The ON caching state causes the specified file always to be cached, while the OFF caching state causes the specified file never to be cached.

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

Note: The cache advisories operate only if Cached ODM is enabled for the file system. If the `odm_cache_enable` flag is zero, Cached ODM is OFF for all of the files in that file system, even if the individual file cache advisory for a file is ON.

To enable unconditional caching on a file

- Enable unconditional caching on the file `/mnt1/file1`:

```
# odmadm setcachefile /mnt1/file1=on
```

  With this command, ODM caches all reads from `file1`.

To disable caching on a file

- Disable caching on the file `/mnt1/file1`:

```
# odmadm setcachefile /mnt1/file1=off
```

  With this command, ODM does not cache reads from `file1`.

To check on the current cache advisory settings for a file

- Check the current cache advisory settings of the files `/mnt1/file1` and `/mnt2/file2`:

```
# odmadm getcachefile /mnt1/file1 /mnt2/file2
/mnt1/file1,ON
/mnt2/file2,OFF
```

To reset all files to the default cache advisory

- Reset all files to the default cache advisory:

```
# odmadm resetcachefiles
```

Tuning Cached ODM settings via the cachemap

You can use the `odmadm setcachemap` command to configure the cachemap. The cachemap maps file type and I/O type combinations to caching advisories. ODM uses the cachemap for all files that have the default conditional cache setting. Such files are those for which caching has not been turned on or off by the `odmadm setcachefile` command.

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

By default, the cachemap is empty, but you can add caching advisories by using the `odmadm setcachemap` command.

To add caching advisories to the cachemap

- Add a caching advisory to the cachemap:

  ```
  # odmadm setcachemap data/data_read_seq=cache,readahead
  ```

  With this example command, ODM uses caching and readahead for I/O to online log files (`data`) that have the `data_read_seq` I/O type. You can view the valid file type and I/O type values from the output of the `odmadm getcachemap` command.

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

Making the caching settings persistent across mounts

By default, the Cached ODM settings are not persistent across mounts. You can make the settings persistent by creating the `/etc/vx/odmadm` file and listing the caching advisory settings in the file

To make the caching setting persistent across mounts

- Create the `/etc/vx/odmadm` file to list files and their caching advisories. In the following example of the `/etc/vx/odmadm` file, if you mount the `/dev/vx/dsk/rootdg/vol1` device at `/mnt1`, odmadm turns off caching for `/mnt1/oradata/file1`:

  ```
  setcachemap data/read_data_header=cache
  setcachemap all/datapump=cache,readahead
  device /dev/vx/dsk/rootdg/vol1
  setcachefile oradata/file1=off
  ```

Chapter 7

Using Database Storage Checkpoints and Storage Rollback

This chapter includes the following topics:

About Storage Checkpoints and Storage Rollback in SF Enterprise products

The Veritas Storage Checkpoint feature is available with SF Enterprise products as part of the Veritas File System package and is used for the efficient backup and recovery of Oracle databases. Storage Checkpoints can also be mounted, allowing regular file system operations to be performed or secondary databases to be started. Review the following information on Storage Checkpoints and Storage Rollback and how to use these technologies through Storage Foundation.

Note: Veritas Storage Foundation Enterprise products only supports the SFDB features described in this guide. Additionally, the information in this chapter is only applicable for a Veritas Storage Foundation Enterprise products configuration. For information about single instance configurations and Storage Checkpoints and Storage Rollback, please refer to the appropriate Storage Foundation documentation.

Using Storage Checkpoints and Storage Rollback for backup and restore

Storage Checkpoints and Storage Rollback enable efficient backup and recovery of Oracle databases.

Storage Checkpoints

A Storage Checkpoint instantly creates an exact image of a database and provides a consistent image of the database from the point in time the Storage Checkpoint was created. The Storage Checkpoint image is managed and available through the Veritas Storage Foundation command line interface (CLI).

Note: A Storage Checkpoint persists after a system reboot.

Veritas NetBackup also makes use of Storage Checkpoints to provide a very efficient Oracle backup mechanism.

Storage Rollbacks

A direct application of the Storage Checkpoint facility is Storage Rollback.

Each Storage Checkpoint is a consistent, point-in-time image of a file system, and Storage Rollback is the restore facility for these on-disk backups. Storage Rollback rolls back changed blocks contained in a Storage Checkpoint into the primary file system for faster database restoration.

Storage Checkpoints and Storage Rollback process

A Storage Checkpoint is a disk and I/O efficient snapshot technology for creating a "clone" of a currently mounted file system (the primary file system). Like a snapshot file system, a Storage Checkpoint appears as an exact image of the snapped file system at the time the Storage Checkpoint was made. However, unlike a snapshot file system that uses separate disk space, all Storage Checkpoints share the same free space pool where the primary file system resides unless a Storage Checkpoint allocation policy is assigned.

Note: A Storage Checkpoint can be mounted as read only or read-write, allowing access to the files as if it were a regular file system. A Storage Checkpoint is created using the `dbed_ckptcreate` command.

Initially, a Storage Checkpoint contains no data. The Storage Checkpoint only contains the inode list and the block map of the primary fileset. This block map points to the actual data on the primary file system. Because only the inode list and block map are required and no data is copied, creating a Storage Checkpoint takes only a few seconds and very little space.

A Storage Checkpoint initially satisfies read requests by finding the data on the primary file system, using its block map copy, and returning the data to the requesting process. When a write operation changes a data block in the primary file system, the old data is first copied to the Storage Checkpoint, and then the primary file system is updated with the new data. The Storage Checkpoint maintains the exact view of the primary file system at the time the Storage Checkpoint was taken. Subsequent writes to block n on the primary file system do not result in additional copies to the Storage Checkpoint because the old data only needs to be saved once. As data blocks are changed on the primary file system, the Storage Checkpoint gradually fills with the original data copied from the primary file system, and less and less of the block map in the Storage Checkpoint points back to blocks on the primary file system.

You can set a quota to limit how much space a file system will give to all storage checkpoints, to prevent the checkpoints from consuming all free space.

See the command `dbed_ckptquota` for more information.

Storage Rollback restores a database, a tablespace, or datafiles on the primary file systems to the point-in-time image created during a Storage Checkpoint. Storage Rollback is accomplished by copying the "before" images from the appropriate Storage Checkpoint back to the primary file system. As with Storage Checkpoints, Storage Rollback restores at the block level, rather than at the file level. Storage Rollback is executed using the `dbed_ckptrollback` command.

Whenever you change the structure of the database (for example, by adding or deleting datafiles, converting PFILE to SPFILE, or converting SPFILE to PFILE), you must run the `dbed_update` command.

For example:

```
$ /opt/VRTS/bin/dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

Mountable Storage Checkpoints can be used for a wide range of application solutions including the following:

- Backups
- Investigations into data integrity
- Staging upgrades
- Database modifications
- Data replication solutions

If you mount a Storage Checkpoint as read-write, the command will not allow you to roll back to this Storage Checkpoint. This ensures that any Storage Checkpoint data that has been modified incorrectly cannot be a source of any database corruption. When a Storage Checkpoint is mounted as read-write, the `dbed_ckptmount` command creates a "shadow" Storage Checkpoint of and mounts this "shadow" Storage Checkpoint as read-write. This allows the database to still be rolled back to the original Storage Checkpoint.

For more information on mountable Storage Checkpoints:

See "Mounting Storage Checkpoints using dbed_ckptmount" on page 138.

Determining space requirements for Storage Checkpoints

To support Block-level Incremental (BLI) Backup and storage rollback, the file systems need extra disk space to store the Storage Checkpoints. The extra space needed depends on how the Storage Checkpoints are used. Storage Checkpoints that are used to keep track of the block changes contain only file system block maps, and therefore require very little additional space (less than 1 percent of the file system size).

If the database is online while the backup is running, the additional space required by each file system for Storage Checkpoints depends on the duration of the backup and the database workload. If workload is light during the backup or the backup window is relatively short (for example, for incremental backups), for most database configurations, an additional 10 percent of the file system size will be sufficient. If the database has a busy workload while a full backup is running, the file systems may require more space.

To support Storage Checkpoints and storage rollback, VxFS needs to keep track of the original block contents when the Storage Checkpoints were created. The additional space needed is proportional to the number of blocks that have been changed since a Storage Checkpoint was taken. The number of blocks changed may not be identical to the number of changes. For example, if a data block has been changed many times, only the first change requires a new block to be allocated to store the original block content. Subsequent changes to the same block require no overhead or block allocation.

If a file system that has Storage Checkpoints runs out of space, by default VxFS removes the oldest Storage Checkpoint automatically instead of returning an ENOSPC error code (UNIX errno 28- No space left on device), which can cause the Oracle instance to fail. Removing Storage Checkpoints automatically ensures the expected I/O semantics, but at the same time, eliminates a key recovery mechanism.

When restoring a file system that has data-full Storage Checkpoints from tape or other offline media, you need extra free space on the file system. The extra space is needed to accommodate the copy-on-write algorithm needed for preserving the consistent image of the Storage Checkpoints. The amount of free space required depends on the size of the restore and the number of Storage Checkpoints on the file system.

If you are restoring the entire file system, in most cases, you no longer need the existing Storage Checkpoint. You can simply re-make the file system using the `mkfs` command, and then restore the file system from tape or other offline media.

If you are restoring some of the files in the file system, you should first remove the data-full Storage Checkpoints that are no longer needed. If you have very limited free space on the file system, you may have to remove all data-full Storage Checkpoints in order for the restore to succeed.

To avoid unnecessary Storage Checkpoint removal, instead of using a low quota limit use the SFDB utility to set up a Monitoring Agent to monitor file system space usage. When file system space usage exceeds a preset threshold value (for example, 95 percent full), the Monitoring Agent alerts the system administrator and optionally grows the volume and the file system. Automatic notifications to the system administrator on the status of space usage and file system resizing are available through electronic mail, the `syslogd`(1M) program, or by logging messages to a simple log file.

Always reserve free disk space for growing volumes and file systems. You can also preallocate sufficient space for each file system when the file system is first created or manually grow the file system and logical volume where the file system resides.

For more information, refer to the `vxassist`(1) and `fsadm_vxfs`(1) manual pages.

Storage Checkpoint Performance

Veritas File System attempts to optimize the read and write access performance on both the Storage Checkpoint and the primary file system. Reads from a Storage Checkpoint typically perform at nearly the throughput of reads from a normal VxFS file system, allowing backups to proceed at the full speed of the VxFS file system.

Writes to the primary file system are typically affected by the Storage Checkpoints because the initial write to a data block requires a read of the old data, a write of the data to the Storage Checkpoint, and finally, the write of the new data to the primary file system. Having multiple Storage Checkpoints on the same file system, however, will not make writes slower. Only the initial write to a block suffers this penalty, allowing operations such as writes to the intent log or inode updates to proceed at normal speed after the initial write.

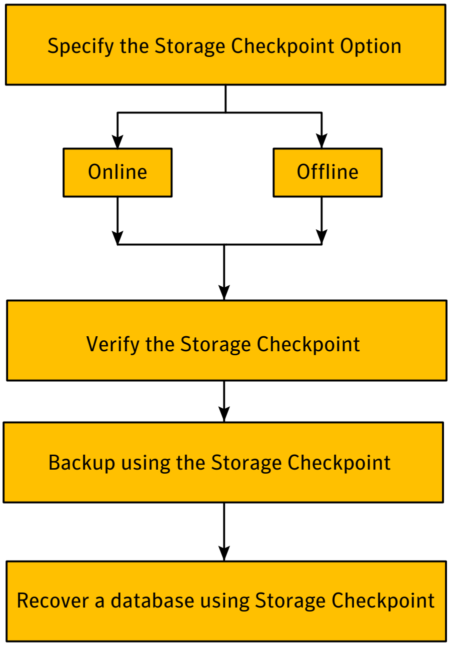
The performance impact of Storage Checkpoints on a database is less when the database files are Direct I/O files. A performance degradation of less than 5 percent in throughput has been observed in a typical OLTP workload when the Storage Checkpoints only keep track of changed information. For Storage Checkpoints that are used for storage rollback, higher performance degradation (approximately 10 to 20 percent) has been observed in an OLTP workload. The degradation should be lower in most decision-support or data-warehousing environments.

Reads from the Storage Checkpoint are impacted if the primary file system is busy, because the reads on the Storage Checkpoint are slowed by all of the disk I/O associated with the primary file system. Therefore, performing database backup when the database is less active is recommended.

Backing up and recovering the database using Storage Checkpoints

Figure 7-1 below describes the general process for backing up and recovering the database using Storage Checkpoints.

Figure 7-1 Backing up and recovering database using Storage Checkpoints



The following sections discuss this process:

- Specify the Storage Checkpoint option
- Verifying a Storage Checkpoint
- Backing up using a Storage Checkpoint
- Recovering a database using a Storage Checkpoint

Specify the Storage Checkpoint option

Storage Checkpoints can be created by specifying one of the following options:

- Online
- Offline
- Instant

Note: Instant Storage Checkpoint is not supported for Oracle RAC.

To create a Storage Checkpoint with the online option, the database should be online and you must enable ARCHIVELOG mode for the database.

Note: Refer to your Oracle documentation for information about enabling the archive log.

For the offline option, the database should be offline.

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. You can roll back the entire database or individual tablespaces or datafiles to an online or offline Storage Checkpoint. After the rollback is complete, you may roll the database forward to restore the database if you have used an online Storage Checkpoint.

For the instant option, the database should be online and it can be running in either ARCHIVELOG or NOARCHIVELOG mode. You can only roll back the entire database to an instant Storage Checkpoint. Rolling back individual tablespaces or datafiles to an instant Storage Checkpoint is not possible. After the rollback is complete, you need to perform database recovery. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

Note: To allow the easiest recovery, always keep ARCHIVELOG mode enabled, regardless of whether the database is online or offline when you create Storage Checkpoints.

Verifying a Storage Checkpoint

After creating a Storage Checkpoint and before using it to back up or restore a database, you can verify that the Storage Checkpoint is free of errors.

Usage notes	See the `dbed_ckptcreate`(1M) and `dbed_ckptmount`(1M) manual pages for more information.

See "Creating Storage Checkpoints using dbed_ckptcreate" on page 133.

See "Mounting Storage Checkpoints using dbed_ckptmount" on page 138.

Storage Checkpoints can only be used to restore from logical errors (for example, a human error). Storage Checkpoints cannot be used to restore files due to a media failure, because all the data blocks are on the same physical device.

A media failure requires a database restore from a tape backup or a copy of the database files kept on a separate medium. The combination of data redundancy (disk mirroring) and Storage Checkpoints is recommended for protecting highly critical data from both physical media failure and logical errors.

To verify that a Storage Checkpoint is error-free

1. As oracle user, create and mount a Storage Checkpoint by issuing the following commands:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD1 -H $ORACLE_HOME\
 -o online
```

```
Storage Checkpoint Checkpoint_1244130973 created.
```

```
$ mkdir /tmp/testckpt
```

```
$  /opt/VRTS/bin/dbed_ckptmount -S PROD1\
 -c Checkpoint_1244130973 -m /tmp/testckpt -o rw
```

```
Creating Storage Checkpoint on /tmp/testckpt/oradata with name
Checkpoint_1244130973_wr001
```

If the specified mount point directory does not exist, then the `dbed_ckptmount` command creates it before mounting the Storage Checkpoint, as long as the Oracle DBA user has permission to create it.

2. Examine the contents of the Storage Checkpoint:

```
$ ls -l /tmp/testckpt/oradata/PROD1
total 4438620
-rw-r-----   1 oracle     oinstall   18628608 Jun  4 22:07 control01.ctl
-rw-r-----   1 oracle     oinstall   18628608 Jun  4 22:07 control02.ctl
-rw-r-----   1 oracle     oinstall   18628608 Jun  4 22:07 control03.ctl
-rw-r-----   1 oracle     oinstall      3072 May 26 17:19 orapwPROD1
-rw-r-----   1 oracle     oinstall   52429824 Jun  4 22:05 redo01.log
-rw-r-----   1 oracle     oinstall   52429824 Jun  4 21:10 redo02.log
-rw-r-----   1 oracle     oinstall   52429824 Jun  4 22:05 redo03.log
-rw-r-----   1 oracle     oinstall   52429824 Jun  4 21:50 redo04.log
-rw-r-----   1 oracle     oinstall   52429824 Jun  4 22:07 redo05.log
-rw-r-----   1 oracle     oinstall   52429824 Jun  4 21:52 redo06.log
-rw-r-----   1 oracle     oinstall   1027547136 Jun  4 22:07 sysaux01.db
-rw-r-----   1 oracle     oinstall   734011392 Jun  4 22:07 system01.dbf
-rw-r-----   1 oracle     oinstall   20979712 Jun  4 22:02 temp01.dbf
-rw-r-----   1 oracle     oinstall   57679872 Jun  4 22:07 undotbs01.dbf
-rw-r-----   1 oracle     oinstall   26222592 Jun  4 22:07 undotbs02.dbf
-rw-r-----   1 oracle     oinstall   30416896 Jun  4 22:07 undotbs03.dbf
-rw-r-----   1 oracle     oinstall   5251072 Jun  4 22:07 users01.dbf
```

3 Run the dbv tool against the datafile. For example:

```
$ $ORACLE_HOME/bin/dbv file=/tmp/testckpt/oradata/\
PROD1/undotbs01.dbf

DBVERIFY: Release 11.1.0.6.0 - Production on Thu Jun 4 21:35:03 2009

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

DBVERIFY - Verification starting : FILE = /tmp/testckpt/oradata/PROD1\
/undotbs01.dbf


DBVERIFY - Verification complete

Total Pages Examined         : 7040
Total Pages Processed (Data) : 0
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 0
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 6528
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 512
Total Pages Marked Corrupt   : 0
Total Pages Influx           : 0
Total Pages Encrypted        : 0
Highest block SCN            : 6532192 (0.6532192)
$
```

Backing up using a Storage Checkpoint

You can back up a database by creating a Storage Checkpoint using the `dbed_ckptcreate` command, mount the Storage Checkpoint as read only using the `dbed_ckptmount` command, and then back it up using tools such as `tar` or `cpio`.

Usage notes	See the `dbed_ckptcreate`(1M), `dbed_ckptmount`(1M), `tar`(1), and `cpio`(1) manual pages for more information.

See “Creating Storage Checkpoints using dbed_ckptcreate” on page 133.

See “Mounting Storage Checkpoints using dbed_ckptmount” on page 138.

In the example procedure, all the database datafiles reside on one VxFS file system named /db01.

To back up a frozen database image using the command line

1 As an Oracle user, create a Storage Checkpoint using the `dbed_ckptcreate` command:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD -H /oracle/product \
-o online
```

```
Storage Checkpoint Checkpoint_903937870 created.
```

2 Mount the Storage Checkpoint using the `dbed_ckptmount` command:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_903937870 \
-m /tmp/ckpt_ro
```

If the specified mount point directory does not exist, then the `dbed_ckptmount` command creates it before mounting the Storage Checkpoint, as long as the Oracle DBA user has permission to create it.

3 Use tar to back up the Storage Checkpoint:

```
$ cd /tmp/ckpt_ro
$ ls
db01
$ tar cvf /tmp/PROD_db01_903937870.tar ./db01
```

Recovering a database using a Storage Checkpoint

Since Storage Checkpoints record the "before" images of blocks that have changed, you can use them to do a file-system-based storage rollback to the exact time when the Storage Checkpoint was taken. You can consider Storage Checkpoints as backups that are online, and you can use them to roll back an entire database, a tablespace, or a single database file. Rolling back to or restoring from any Storage Checkpoint is generally very fast because only the changed data blocks need to be restored.

Some database changes made after a Storage Checkpoint was taken may make it impossible to perform an incomplete recovery of the databases after Storage Rollback of an online or offline Storage Checkpoint using the current control files. For example, you cannot perform an incomplete recovery of the database to the point right before the control files have recorded the addition or removal of datafiles.

Warning: Use extreme caution when recovering your database using alternate control files.

Suppose a user deletes a table by mistake right after 4:00 p.m., and you want to recover the database to a state just before the mistake. You created a Storage Checkpoint (Checkpoint_903937870) while the database was running at 11:00 a.m., and you have ARCHIVELOG mode enabled.

To recover the database using a Storage Checkpoint

1. As root, freeze the VCS service group for the database.

   ```
   # hagrp -freeze Service_Group
   ```

2. Ensure that the affected datafiles, tablespaces, or database are offline.
3. Use storage rollback to roll back any datafiles in the database that contained the table data from the Storage Checkpoint you created at 11:00 a.m.

   For example:

   ```
   $ /opt/VRTS/bin/dbed_ckptrollback -S $ORACLE_SID -H\
   $ORACLE_HOME -c Checkpoint_903937870
   ```

   For other examples of this command (for a database, tablespace, or datafile):

   See "Performing Storage Rollback using dbed_ckptrollback" on page 139.
4. Start up the database instance if it is down.
5. Unfreeze the service group.

   ```
   # hagrp -unfreeze Service_Group
   ```

6. Re-apply archive logs to the point before the table was deleted to recover the database to 4:00 p.m. Use one of the following commands:

   ```
   SQL> recover database until cancel
   ```

   ```
   SQL> recover database until change
   ```

   ```
   SQL> recover database until time
   ```

7. Open the database with the following command:

   ```
   SQL> alter database open resetlogs
   ```

8 Delete the Storage Checkpoint you created at 11:00 a.m. and any other Storage Checkpoints created before that time.

9 Create a new Storage Checkpoint.

Guidelines for Oracle recovery

For an optimal Oracle recovery, the following steps should be taken:

- Back up all control files before Storage Rollback
- Ensure that the control files are not rolled back
- Ensure that all archived redo logs are available
- Media recovery procedures

Back up all control files before Storage Rollback

This guideline is recommended in case the subsequent Oracle recovery is not successful.

Oracle recommends that you keep at least two copies of the control files for each Oracle database and that you store the copies on different disks. Control files should also be backed up before and after making structural changes to databases.

Note: The `dbed_ckptcreate` command automatically saves control file and log information when you create a Storage Checkpoint.

See "Creating Storage Checkpoints using dbed_ckptcreate" on page 133.

Ensure that the control files are not rolled back

A control file is a small binary file that describes the structure of the database and must be available to mount, open, and maintain the database. The control file stores all necessary database file information, log file information, the name of the database, the timestamp of database creation, and synchronization information, such as the Storage Checkpoint and log-sequence information needed for recovery.

Rolling back the control file will result in an inconsistency between the physical database structure and the control file.

Ensure that all archived redo logs are available

A database backup with online and archived logs is required for a complete database recovery.

Query V$ARCHIVED_LOG to list all the archived log information and V$ARCHIVE_DEST to list the location of archive destinations.

Note: Refer to your Oracle documentation for information about querying archived information.

For Storage Foundation for Oracle RAC, the archive log destination must be on a Veritas cluster file system.

To restore the necessary archived redo log files, you can query V$LOG_HISTORY to list all the archived redo log history or query V$RECOVERY_LOG to list only the archived redo logs needed for recovery. The required archived redo log files can be restored to the destination specified in the LOG_ARCHIVE_DEST parameter or to an alternate location. If the archived redo logs were restored to an alternate location, use the `ALTER DATABASE RECOVER ... FROM` statement during media recovery.

After Storage Rollback, perform Oracle recovery, applying some or all of the archived redo logs.

Note: After rolling back the database (including control files and redo logs) to a Storage Checkpoint, you need to recover the Oracle database instance. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

Media recovery procedures

The following are the procedures for performing either a complete or incomplete media recovery.

Media recovery procedures

- To perform a complete media recovery:

  ```
  SQL> SET AUTORECOVERY ON;
  ```

  ```
  SQL> RECOVER DATABASE;
  ```

- To perform an incomplete media recovery, use one of the following:

```
SQL> RECOVER DATABASE UNTIL CANCEL;
```

or

```
SQL> RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss';
```

(You can confirm the time of error by checking the ../bdump/alert*.log file.)
or

```
SQL> RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss'\
using backup controlfile;
```

or

```
SQL> RECOVER DATABASE UNTIL CHANGE scn;
```

- To open the database after an incomplete media recovery, use the following:

  ```
  SQL> ALTER DATABASE OPEN RESETLOGS;
  ```

  RESETLOGS resets the log sequence. The RESETLOGS option is required after an incomplete media recovery. After opening the database with the RESETLOGS option, remove the Storage Checkpoint you just rolled back to as well as any Storage Checkpoints that were taken before that one. These earlier Storage Checkpoints can no longer be used for storage rollback. After removing these Storage Checkpoints, be sure to create a new Storage Checkpoint.

Warning: Attempting to roll back to the same Storage Checkpoint more than once can result in data corruption. After rolling back, be sure to delete the Storage Checkpoint that you rolled back to and then create a new one.

See your Oracle documentation for complete information on recovery.

Using the Storage Checkpoint Command Line Interface (CLI)

The Storage Foundation for Databases tools provide a command line interface to many key operations. The command line interface enables you to incorporate command operations into scripts and other administrative processes.

Note: The Storage Foundation for Databases (SFDB) tools command line interface depends on certain tablespace and container information that is collected and stored in a repository. Some CLI commands update the repository by default. It is also important to regularly ensure that the repository is up-to-date by using the `dbed_update` command.

Note: For SF Enterprise products database, when you issue the commands, replace `$ORACLE_SID` with `$ORACLE_SID=instance_name` and provide the instance name on which the instance is running.

Commands Overview

SF Enterprise products commands supported in the command line interface are located in the /opt/VRTS/bin directory.

The online manual pages for these commands are located in the /opt/VRTS/man directory.

Table 7-1 summarizes the commands available to you from the command line.

Table 7-1 Database Checkpoint commands

Command	Description
`dbed_update`	Command that creates or updates the SFDB repository. See "Creating or updating the repository using dbed_update" on page 132.
`dbed_ckptcreate`	Command that creates a Storage Checkpoint for an Oracle database. See "Creating Storage Checkpoints using dbed_ckptcreate" on page 133.

Table 7-1 Database Checkpoint commands *(continued)*

Command	Description
`dbed_ckptdisplay`	Command that displays the Storage Checkpoints associated with an Oracle instance. See "Displaying Storage Checkpoints using dbed_ckptdisplay" on page 134.
`dbed_ckptmount`	Command that mounts a Storage Checkpoint for an Oracle instance. See "Mounting Storage Checkpoints using dbed_ckptmount" on page 138.
`dbed_ckptquota`	Command that administers quotas for Storage Checkpoints. **Note:** This command only administers quotas for Storage Checkpoints for the local instance for SF Oracle RAC.
`dbed_ckptumount`	Command that unmounts a Storage Checkpoint for an Oracle instance. See "Unmounting Storage Checkpoints using dbed_ckptumount" on page 139.
`dbed_ckptrollback`	Command that rolls back an Oracle instance to a Storage Checkpoint point-in-time image. See "Performing Storage Rollback using dbed_ckptrollback" on page 139.
`dbed_ckptremove`	Command that removes a Storage Checkpoint for an Oracle instance. See "Removing Storage Checkpoints using dbed_ckptremove" on page 141.
`dbed_clonedb`	Command that creates a copy of an Oracle database by cloning all existing database files and recreating the control file accordingly. This cloned database can only be started on the same host as the existing database as long as it uses a different SID. See "Cloning the Oracle instance using dbed_clonedb" on page 141.

Command Line Interface examples

This section displays examples of SF Enterprise products commands that are used to perform administrative operations for Storage Checkpoints and Storage Rollbacks.

Note: For detailed information about these commands, their command syntax, and available options, see the individual manual pages.

Prerequisites

Review the prerequisites and usage notes listed below for each command before using that command.

Creating or updating the repository using dbed_update

You can use the `dbed_update` command to create or update the repository.

Note: Any time you change the structure of the database (for example, by adding or deleting datafiles, converting PFILE to SPFILE, or converting SPFILE to PFILE), you must run the `dbed_update` command.

Before creating or updating the repository, the following conditions must be met:

Prerequisites	■ As root, you must run `/opt/VRTSdbed/common/bin/sfua_db_config` before creating the repository. It will set the owner and group of various directories needed for SFDB funtionality. ■ You must be logged on as the database administrator (typically, the user ID oracle).
Usage notes	■ The `dbed_update` command creates a repository in the /etc/vx/vxdba/*$ORACLE_SID* directory where information used by SF Enterprise products is kept. If the repository already exists, the command will refresh the information. ■ The database must be up and running, and the *ORACLE_SID* and the *ORACLE_HOME* variable arguments must be specified with the `-S` and `-H` options, respectively. ■ See the `dbed_update`(1M) manual page for more information.

To update the repository

◆ Use the dbed_update command as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/ORA_HOME
```

To view the status of the repository

◆ Use the `dbed_update` command as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/ORA_HOME
```

Creating Storage Checkpoints using dbed_ckptcreate

You can use the `dbed_ckptcreate` command to create a Storage Checkpoint for an Oracle database from the command line.

Storage Checkpoints can be online, offline, or instant. By default, Storage Checkpoints are offline. If online is specified, the database is put into hot-backup mode when the Storage Checkpoint is created. If offline is specified, the database is expected to be down.

If instant is specified, the database must be online and a Storage Checkpoint will be taken for a "crash recovery"-type Storage Rollback.

Note: For Storage Foundation for Oracle RAC, instant and offline checkpoints are not supported.

Before creating a Storage Checkpoint, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID oracle). ■ For best recoverability, always keep ARCHIVELOG mode enabled when you create Storage Checkpoints.
Usage notes	■ dbed_ckptcreate stores Storage Checkpoint information in the SFDB repository. ■ See the `dbed_ckptcreate`(1M) manual page for more information.

To create Storage Checkpoints while the database is online

- Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product/ORA_HOME -o online
```

```
Storage Checkpoint Checkpoint_971672043 created.
```

To create Storage Checkpoints without updating the repository while the database is online

- Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product/ORA_HOME -o online -n
```

```
Storage Checkpoint Checkpoint_971672046 created.
```

To create Storage Checkpoints while the database is offline

- Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product/ORA_HOME -o offline
```

```
Storage Checkpoint Checkpoint_971672049 created.
```

Note: The default option is offline.

To assign a Storage Checkpoint allocation policy to a Storage Checkpoint

- Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product/ORA_HOME -o online -p ckpt_data,ckpt_metadata
```

```
Storage Checkpoint Checkpoint_971672055 created.
```

Displaying Storage Checkpoints using dbed_ckptdisplay

You can use the `dbed_ckptdisplay` command to display the Storage Checkpoints associated with an Oracle database from the command line.

You can also use it to display fileset quota values.

Before displaying Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage Notes	■ In addition to displaying the Storage Checkpoints created by SF Enterprise products, `dbed_ckptdisplay` also displays other Storage Checkpoints (for example, Storage Checkpoints created by the Capacity Planning Utility and NetBackup). ■ The Status field identifies if the Storage Checkpoint is partial (P), complete (C), invalid (I), mounted (M), read only (R), writable (W), or of type online (ON), offline (OF), instant (IN), or unknown (UN). Note that instant (IN) Storage Checkpoints are not supported in an SF Oracle RAC environment. ■ Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply. See "Mounting the snapshot volumes and backing up" on page 196. ■ See the `dbed_ckptdisplay`(1M) manual page for more information.

To display Database Checkpoints

◆ Use the `dbed_ckptdisplay` command as follows to display information for Storage Checkpoints created by SF Enterprise products:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME
```

```
Checkpoint_974428422_wr001Thu May 16 17:28:42 2005       C+R+ON
Checkpoint_974428423      Thu May 16 17:28:42 2004       P+R+ON
```

To display other Storage Checkpoints

◆ Use the `dbed_ckptdisplay` command as follows: :

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o other
```

```
NetBackup_incr_PROD_955187480        NBU       /db01
NetBackup_full_PROD_95518725   54      NBU       /db01
```

To display other Storage Checkpoints without updating the repository

◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o other -n

NetBackup_incr_PROD_955187480         NBU      /db01
NetBackup_full_PROD_95518725   54      NBU      /db01
```

To display all Storage Checkpoints

◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o all

Checkpoint_903937870   Fri May 13 22:51:10 2005        C+R+ON
Checkpoint_901426272   Wed May 11 16:17:52 2005        P+R+ON
NetBackup_incr_PROD_955133480         NBU          /db01
NetBackup_full_PROD_9551329   52      NBU          /db01
```

To display all Storage Checkpoints without updating the repository

◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o all -n

Checkpoint_903937870   Fri May 13 22:51:10 2005        C+R+ON
Checkpoint_901426272   Wed May 11 16:17:52 2005        P+R+ON
NetBackup_incr_PROD_955133480           NBU        /db01
NetBackup_full_PROD_9551329   52        NBU        /db01
```

To display fileset quota values

◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD -c \
Checkpoint_903937870 -Q
Checkpoint_903937870      Wed Mar 19 9:12:20 2005       C+R+ON
Filesystem                HardLim    SoftLim    CurrentUse
/oradata1/indx1_1         100000     50000      2028
/oradata1/user1_1         100000     50000      2028
/oradata1/temp            150000     80000      2142
/oradata1/system1         150000     70000      3092
```

Scheduling Storage Checkpoints using dbed_ckptcreate and cron

You can use the `dbed_ckptcreate` command to schedule Storage Checkpoint creation in a `cron` job or other administrative script.

Before scheduling Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID oracle).
Usage notes	■ Create a new crontab file or edit an existing crontab file to include a Storage Checkpoint creation entry with the following space-delimited fields: *minute hour day_of_month month_of_year day_of_week* \/opt/VRTS/bin/dbed_ckptcreate where: *minute* - numeric values from 0-59 or * *hour* - numeric values from 0-23 or * *day_of_month* - numeric values from 1-31 or * *month_of_year* - numeric values from 1-12 or * *day_of_week* - numeric values from 0-6, with 0=Sunday or * Each of these variables can either be an asterisk (meaning all legal values) or a list of elements separated by commas. An element is either a number or two numbers separated by a hyphen (meaning an inclusive range). ■ See the `dbed_ckptcreate`(1M), `cron`(1M), and `crontab`(1) manual pages for more information.

Scheduling Storage Checkpoint creation in a cron job

Depending on when you want to schedule Storage Checkpoint creation, make entries to the crontab file.

- To create a Storage Checkpoint at 1:00 a.m. every Sunday while the database is offline, include the following entry in your crontab file:

```
0 1 * * 0 /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product/ORA_HOME -o offline
```

Note: This is a crontab example for user oracle.

Mounting Storage Checkpoints using dbed_ckptmount

You can use the `dbed_ckptmount` command to mount a Storage Checkpoint for the database from the command line.

Before mounting Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage notes	■ The `dbed_ckptmount` command is used to mount a Storage Checkpoint into the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands. ■ Storage Checkpoints can be mounted as read only or read-write. By default, Storage Checkpoints are mounted as read only. ■ If the rw (read-write) option is used, _wrxxx, where xxx is an integer, will be appended to the Storage Checkpoint name. ■ If the specified mount point directory does not exist, then `dbed_ckptmount` creates it before mounting the Storage Checkpoint, as long as the Oracle database owner has permission to create it. ■ Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply. See "Mounting the snapshot volumes and backing up" on page 196. ■ See the `dbed_ckptmount`(1M) manual page for more information.

To mount Storage Checkpoints with the read/write option

◆ Use the `dbed_ckptmount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_971672042 \
-m /tmp/ckpt_rw -o rw
Creating Storage Checkpoint on /tmp/ckpt_rw/share/oradata with
name Checkpoint_971672042_wr001
```

To mount Storage Checkpoints with the read only option

◆ Use the dbed_ckptmount command as follows:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_971672042 \
-m /tmp/ckpt_ro -o ro
```

Unmounting Storage Checkpoints using dbed_ckptumount

You can use the `dbed_ckptumount` command to unmount a Storage Checkpoint for an Oracle database from the command line.

Before unmounting Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage notes	■ The `dbed_ckptumount` command is used to unmount a mounted Storage Checkpoint from the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands. When mounted Storage Checkpoints are not required, they can be unmounted. ■ See the `dbed_ckptumount`(1M) manual page for more information.

To unmount Storage Checkpoints

◆ Use the `dbed_ckptumount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptumount -S PROD \
-c Checkpoint_971672042
```

Performing Storage Rollback using dbed_ckptrollback

You can use the `dbed_ckptrollback` command to rollback an Oracle database to a Storage Checkpoint.

Before performing a Storage Rollback, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.

Usage notes	
	■ The dbed_ckptrollback command rolls an Oracle database back to a specified Storage Checkpoint. You can perform a Storage Rollback for the entire database, a specific tablespace, or list of datafiles. Database rollback for the entire database requires that the database be inactive before Storage Rollback commences. The dbed_ckptrollback command will not commence if the Oracle database is active. However, to perform a Storage Rollback of a tablespace or datafile, only the tablespace or datafile to be rolled back must be offline (not the entire database). ■ You must run the dbed_update command after upgrading from a previous release. This will allow you to roll back to a Storage Checkpoint that was created with an earlier version of this product. ■ See the dbed_ckptrollback(1M) manual page for more information.

To roll back an Oracle database to a Storage Checkpoint

◆ Use the dbed_ckptrollback command as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD \
-H /oracle/product/ORA_HOME -c Checkpoint_903937870
```

To rollback a tablespace to a Storage Checkpoint

◆ Use the dbed_ckptrollback command with the -T option as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD \
-H /oracle/product/ORA_HOME -T DATA01 -c Checkpoint_903937870
```

If the Oracle database is running, you must take the tablespace offline before running this command. If the tablespace is online, the command will fail.

To rollback datafiles to a Storage Checkpoint

◆ Use the dbed_ckptrollback command with the -F option as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD \
-H /oracle/product/ORA_HOME \
-F /share/oradata1/data01.dbf,/share/oradata2/index01.dbf \
-c Checkpoint_903937870
```

If the Oracle database is running, you must take the datafile offline before running this command. If the datafile is online, the command will fail.

Removing Storage Checkpoints using dbed_ckptremove

You can use the `dbed_ckptremove` command to remove a Storage Checkpoint for an Oracle database at the command line.

Before removing Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage notes	■ The `dbed_ckptremove` command is used to remove a Storage Checkpoint from the file system, or file systems, it is associated with. The Storage Checkpoint must have been created using the `dbed_ckptcreate`(1M) command. ■ You must unmount the Storage Checkpoint before you can remove it. ■ See the `dbed_ckptremove`(1M) manual page for more information.

To remove Storage Checkpoints

◆ Use the dbed_ckptremove command as follows:

```
$ /opt/VRTS/bin/dbed_ckptremove -S PROD \
-c Checkpoint_971672042_wr001
```

Cloning the Oracle instance using dbed_clonedb

You can use the `dbed_clonedb` command to clone an Oracle instance using a Storage Checkpoint.

Cloning an existing database using a Storage Checkpoint must be done on the same host.

You have the option to manually or automatically recover the database when using the `dbed_clonedb` command:

- Manual (interactive) recovery, which requires using the `-i` option, of the clone database allows the user to control the degree of recovery by specifying which archive log files are to be replayed.
- Automatic (non-interactive) recovery, which is the default usage of the command, recovers the entire database and replays all of the archive logs. You will not be prompted for any archive log names.

Before cloning the Oracle instance, the following conditions must be met:

Prerequisites	■ You must first create a Storage Checkpoint. See "Creating Storage Checkpoints using dbed_ckptcreate" on page 133. ■ You must be logged in as the database administrator. ■ Make sure you have enough space and system resources to create a clone database on your system. ■ A clone database takes up as much memory and machine resources as the primary database.
Usage notes	■ The `dbed_clonedb` command is used to create a copy of a database, cloning all existing database files to new locations. ■ The *ORACLE_SID* and *ORACLE_HOME* environment variables must be set to the primary database. ■ It is assumed that the user has a basic understanding of the database recovery process. ■ See the `dbed_clonedb`(1M) manual page for more information.
Limitations for SF Enterprise products	■ Note that the database cloning using Instant Checkpoint is not supported for SF Enterprise products. ■ When you clone the database by using Checkpoint, the node can be any node in the same SF Enterprise products cluster but the archive log destination is required to be on CFS file system. Otherwise, you must manually copy the archive log files.

Table 7-2 lists the options for the dbed_clonedb command.

Table 7-2 dbed_clonedb command options

Option	Description
`-S` *`CLONE_SID`*	Specifies the name of the new Oracle SID, which will be the name of the new database instance.
`-m` *`MOUNT_POINT`*	Indicates the new mount point of the Storage Checkpoint.
`-c` *`CKPT_NAME`*	Indicates the name of the Storage Checkpoint.
`-i`	Runs the command in interactive mode where you must respond to prompts by the system. The default mode is non-interactive. (Optional)
`-o umount`	Shuts down the clone database and unmounts the Storage Checkpoint file system.
`-o restartdb`	Mounts the Storage Checkpoint file system and starts the clone database. The `-o restartdb` option will not attempt to recover the clone database.

Table 7-2 dbed_clonedb command options *(continued)*

Option	Description
`-d`	Used with the `-o umount` option. If the `-d` option is specified, the Storage Checkpoint used to create the clone database will be removed along with the clone database.
`-p`	Specifies a file containing initialization parameters to be modified or added to the clone database's initialization parameter file prior to startup. The format of the pfile_modification_file is the same as that of the Oracle initialization parameter file.

To clone an Oracle instance with manual Oracle recovery

- Use the `dbed_clonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S new1 -m /tmp/new1 \
-c Checkpoint_1249624009 -i
Primary Oracle SID is FLAS11r1
New Oracle SID is new1
Cloning an online Storage Checkpoint Checkpoint_1249624009.
Mounting Checkpoint_1249624009 at /tmp/new1.
All redo-log files found.
Creating initnew1.ora
        from /tmp/vxdba.rel.tmp.28185/initFLAS11r1.ora.
Altering db_name parameter in initnew1.ora.
Editing remote_login_passwordfile in initnew1.ora.
Altering control file locations in initnew1.ora.
Copying initnew1.ora
        to /oracle/11gr1/dbs/initnew1.ora.
About to start up new database and begin reconfiguration.
Database new1 is being reconfigured.
Did not see error regarding database name.
Altering clone database archive log directory.
Updating log_archive_dest in clone database init file.
Found archive log destination at /snap_arch11r1/FLAS11r1.
The latest archive log(s) must now be applied. To apply
        the logs, open a new window and perform the following steps:
1. You must copy required archive log(s) from primary to clone:
       Primary archive logs in /snap_arch11r1/FLAS11r1
       Clone archive logs expected in /tmp/new1/snap_arch11r1/FLAS11r1.
2. ORACLE_SID=new1; export ORACLE_SID # sh and ksh, OR
     setenv ORACLE_SID new1 #csh
```

```
3. /oracle/11gr1/bin/sqlplus /nolog
4. CONNECT / AS SYSDBA
5. RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
6. enter the archive log(s) you would like to apply
7. EXIT
Press <Return> after you have completed the above steps.
<Return>
Resetting logs on new database NEW1
Database instance NEW1 is up and running
```

To clone an Oracle instance with automatic Oracle recovery

- Use the `dbed_clonedb` command as follows:

```
$/opt/VRTS/bin/dbed_clonedb -S new2 -m /tmp/new2 \
-c Checkpoint_1249624426
Primary Oracle SID is FLAS11r1
New Oracle SID is new2
Cloning an online Storage Checkpoint Checkpoint_1249624426.
Mounting Checkpoint_1249624426 at /tmp/new2.
All redo-log files found.
Creating initnew2.ora
        from /tmp/vxdba.rel.tmp.30929/initFLAS11r1.ora.
Altering db_name parameter in initnew2.ora.
Editing remote_login_passwordfile in initnew2.ora.
Altering control file locations in initnew2.ora.
Copying initnew2.ora
        to /oracle/11gr1/dbs/initnew2.ora.
About to start up new database and begin reconfiguration.
Database new2 is being reconfigured.
Did not see error regarding database name.
Starting automatic database recovery.
Shutting down clone database.
Altering clone database archive log directory.
Updating log_archive_dest in clone database init file.
Found archive log destination at /snap_arch11r1/FLAS11r1.
Mounting clone database.
Resetting logs on new database new2.
The sql script for adding tempfiles to new2 is at /tmp/add_tf.new2.sql.
Database instance new2 is up and running.
```

To shut down the clone database and unmount the Storage Checkpoint

- Use the `dbed_clonedb` command as follows:

```
$ opt/VRTS/bin/dbed_clonedb -S NEW -o umount
```

To mount a Storage Checkpoint file system and start the clone database

- Use the `dbed_clonedb` command as follows:

```
$/opt/VRTS/bin/dbed_clonedb -S NEW -o restartdb
Database instance NEW is up and running.
```

To delete a clone database and the Storage Checkpoint used to create it

- Use the dbed_clonedb command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW -o umount -d
```

Chapter 8

Using Database FlashSnap for backup and off-host processing

This chapter includes the following topics:

About Veritas Database FlashSnap

Database FlashSnap lets you capture an online image of an actively changing database at a given instant, known as a snapshot. You can then perform backups and off-host processing tasks on these snapshots while still maintaining continuous availability of your critical data. Database FlashSnap offers you a flexible way to efficiently manage multiple point-in-time copies of your data, and reduce resource contention on your business-critical servers.

A database snapshot can be used on the same host as the production database or on a secondary host sharing the same storage.

A database snapshot can be used for the following off-host processing applications:

- Data backup
- Data warehousing

- Decision-support queries

When the snapshot is no longer needed, the database administrator can import the original snapshot back to the primary host and resynchronize the snapshot to the original database volumes.

Database FlashSnap commands are executed from the command line interface.

Database FlashSnap also enables you to resynchronize your original database volumes from the data in the snapshot if the original volumes become corrupted. This is referred to as reverse resynchronization.

Reverse resynchronization is:

- Supported for single instance Oracle environments
- Not supported for Oracle RAC environments

Database FlashSnap significantly reduces the time it takes to backup your database, increase the availability of your production database, and still maintain your production database's performance.

Database FlashSnap requirements

The following are requirements for using Database FlashSnap:

- You must have a Storage Foundation Enterprise product installed on all systems on which you intend to use Database FlashSnap.
- To use Database FlashSnap, you must first configure the volumes used by the database. You must set the ORACLE_SID to your SID name.

Solving typical database problems with Database FlashSnap

Database FlashSnap allows database administrators to create a snapshot without root privileges.

Database FlashSnap is designed to enable you to use database snapshots to overcome the following types of problems encountered in enterprise database environments:

- In many companies, there is a clear separation between the roles of system administrators and database administrators. Creating database snapshots typically requires superuser (root) privileges, privileges that database administrators do not usually have.
- In some companies, database administrators are granted root privileges, but managing storage is typically neither central to their job function nor their core competency.

- Creating database snapshots is a complex process, especially in large configurations where thousands of volumes are used for the database. One mistake can render the snapshots useless.

Because root privileges are not required, Database FlashSnap overcomes these obstacles by enabling database administrators to easily create consistent snapshots of the database. The snapshots can be utilized for repetitive use.

Planning to use Database FlashSnap

Before using Database FlashSnap, you must first determine your intended application. You will then need to make the following decisions:

- Which snapshot mode is appropriate: online, offline, or instant?
- Will you need one or two hosts (off-host)?

Selecting the snapshot mode

If your purpose is to use the snapshot for backup or to recover the database after logical errors have occurred, choose the online option. In the event that your production database is offline, choose offline. If you intend to use the snapshot for decision-support analysis, reporting, development, or testing, choose instant. An instant snapshot is not suitable for recovery because it is not necessarily an exact copy of the primary database.

Selecting one or two hosts (off-host)

If maintaining the performance of your primary database is critical, you can offload processing of the snapshots to a secondary host. For off-host processing, storage must be shared between the primary and secondary hosts.

If cost savings is most important, you can choose to do the processing on the same host as the primary database to save on hardware costs.

Preparing hosts and storage for Database FlashSnap

Review the following details to prepare the hosts and storage for Database FlashSnap.

Setting up hosts

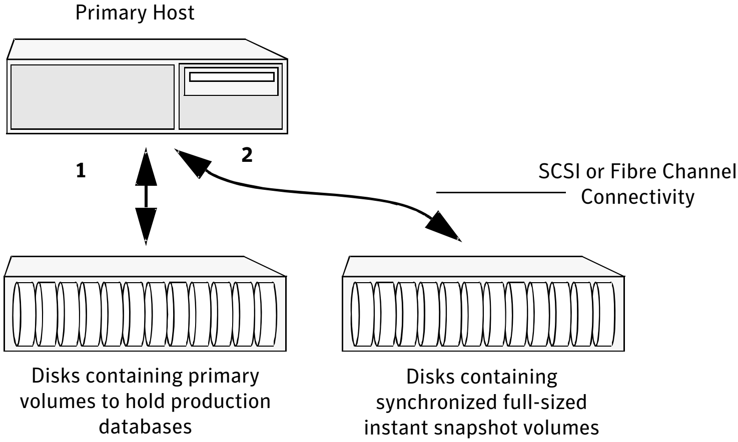
Database FlashSnap requires sufficient disk space in the disk group to add a mirror of equal size of the existing database.

Setting up a storage configuration for Database FlashSnap operations is a system administrator's responsibility and requires superuser (root) privileges. Database FlashSnap utilities do not address setting up an appropriate storage configuration.

Same-node configuration

Figure 8-1 shows the suggested arrangement for implementing Database FlashSnap solutions on the primary host to avoid disk contention.

Figure 8-1 Example of a Database FlashSnap solution on a primary host

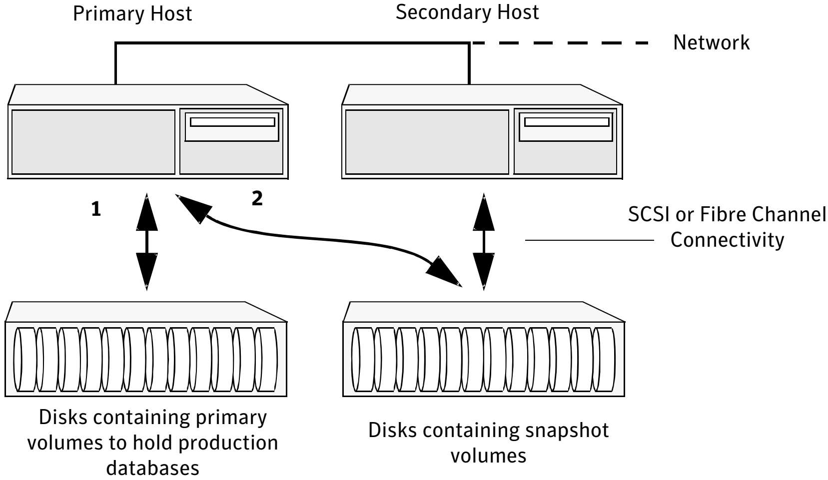


Database FlashSnap off-host configuration

A Database FlashSnap off-host configuration allows CPU- and I/O-intensive operations to be performed for online backup and decision support without degrading the performance of the primary host running the production database. Both the primary and secondary hosts share the storage in which the snapshot database is created. Both the primary and secondary hosts have access to the disks containing the snapshot volumes.

Figure 8-2 shows a Database FlashSnap off-host configuration.

Figure 8-2 Example of an off-host Database FlashSnap solution



Host and storage requirements

Before using Database FlashSnap, ensure that the following requirements are met:

- All files are on VxFS file systems over VxVM volumes. Raw devices are not supported.
- Symbolic links to datafiles are not supported.
- ORACLE_HOME is on a separate file system.
- Archive logs are on a separate VxFS file system and are separate from the VxFS file system containing Oracle data files or ORACLE_HOME.
- The database does not contain BFILES and external tables.
- Oracle datafiles, archive logs, redo logs, and control files are in a single or multiple disk groups.
- If your primary host is part of a VCS cluster, it is not advised to use one of the nodes in the cluster for off-host cloning.

In addition, before attempting to use Database FlashSnap with an off-host configuration, ensure that the following requirements are met:

- The product versions installed on the primary and secondary hosts are the same.

- The same version of Oracle is installed on both hosts, the Oracle binaries and datafiles are on different volumes and disks.
- The UNIX login for the database user and group must be the same on both hosts.
- You have an Enterprise license on both hosts.

Creating a snapshot mirror of a volume or volume set used by the database

With Database FlashSnap, you can mirror the volumes used by the database to a separate set of disks, and those mirrors can be used to create a snapshot of the database. These snapshot volumes can be split and placed in a separate disk group. This snapshot disk group can be imported on a separate host, which shares the same storage with the primary host. The snapshot volumes can be resynchronized periodically with the primary volumes to get recent changes of the datafiles. If the primary datafiles become corrupted, you can quickly restore them from the snapshot volumes. Snapshot volumes can be used for a variety of purposes, including backup and recovery, and creating a clone database.

You must create snapshot mirrors for all of the volumes used by the database datafiles before you can create a snapshot of the database. This section describes the procedure used to create snapshot mirrors of volumes.

Use the `vxsnap` command to create a snapshot mirror or synchronize a snapshot mirror.

Prerequisites

- You must be logged in as superuser (root).
- The disk group must be version 110 or later.
  For more information on disk group versions, see the vxdg(1M) online manual page.
- Be sure that a data change object (DCO) and a DCO log volume are associated with the volume for which you are creating the snapshot.
- Persistent FastResync must be enabled on the existing database volumes and disks must be assigned for the snapshot volumes.
  FastResync optimizes mirror resynchronization by tracking updates to stored data that have been missed by a mirror. When a snapshot mirror is reattached to its primary volumes, only the updates that were missed need to be re-applied to resynchronize it. FastResync increases the efficiency of the volume snapshot mechanism to better support operations such as backup and decision support.
  For detailed information about FastResync, see the *Veritas Volume Manager Administrator's Guide*.
- Snapshot mirrors and their associated DCO logs should be on different disks than the original mirror plexes, and should be configured correctly for creating snapshots by the system administrator.
- When creating a snapshot mirror, create the snapshot on a separate controller and separate disks from the primary volume.
- Allocate separate volumes for archive logs.
- Do not place any datafiles, including control files, in the $ORACLE_HOME/dbs directory.

Usage Notes

- Create a separate disk group for Oracle database-related files.
- Do not share volumes between Oracle database files and other software.
- ORACLE_HOME cannot be included in the snapshot mirror.
- Resynchronization speed varies based on the amount of data changed in both the primary and snapshot volumes during the break-off time.
- Do not share any disks between the original mirror and the snapshot mirror.
- Snapshot mirrors for datafiles and archive logs should be created so that they do not share any disks with the data of the original volumes. If they are not created in this way, the VxVM disk group cannot be split and, as a result, Database FlashSnap will not work.

Note: Database FlashSnap commands support third-mirror break-off snapshots only. The snapshot mirror must be in the SNAPDONE state.

The following sample procedure is for existing volumes without existing snapshot plexes or associated snapshot volumes. In this procedure, *volume_name* is the name of either a volume or a volume set.

Note: You must be logged in as superuser (root) to issue the commands in the following procedure.

To create a snapshot mirror of a volume or volume set

1. To prepare the volume for being snapshot, use the `vxsnap prepare` command:

   ```
   # vxsnap -g diskgroup prepare volume \
   alloc="storage_attribute ..."
   ```

   The `vxsnap prepare` command automatically creates a DCO and DCO volumes and associates them with the volume, and enables Persistent FastResync on the volume. Persistent FastResync is also set automatically on any snapshots that are generated from a volume on which this feature is enabled.

   For enabling persistent FastResync on a volume in VxVM 4.1 or 5.0, either from the command line or from within a script, use the `vxsnap prepare` command as described above.

2. To verify that FastResync is enabled on the volume, use the `vxprint` command:

   ```
   # vxprint -g diskgroup -F%fastresync volume_name
   ```

   This returns on if FastResync is on. Otherwise, it returns off.

3. To verify that a DCO and DCO log volume are attached to the volume, use the `vxprint` command:

   ```
   # vxprint -g diskgroup -F%hasdcolog volume_name
   ```

   This returns on if a DCO and DCO log volume are attached to the volume. Otherwise, it returns off.

4. Create a mirror of a volume:

   ```
   # vxsnap -g diskgroup addmir volume_name alloc=diskname
   ```

   Example of creating 3 mirrors for a particular volume:

   ```
   # vxsnap -g diskgroup addmir datavol \
   nmirror=3 alloc=disk1,disk2,disk3
   ```

5 List the available mirrors:

```
# vxprint -g diskgroup -F%name -e"pl_v_name in \"volume_name\""
```

6 Enable database FlashSnap to locate the correct mirror plexes when creating snapshots:

- Set the dbed_flashsnap tag for the data plex you want to use for breaking off the mirror. You can choose any tag name you like, but it needs to match the tag name specified in the snapplan.

  ```
  # vxedit -g diskgroup set putil2=dbed_flashsnap plex_name
  ```

- Verify that the dbed_flashsnap tag has been set to the desired data plex:

  ```
  # vxprint -g diskgroup -F%name -e"pl_v_name in \
  \"volume_name\" && p2 in \"dbed_flashsnap\""
  ```

If you require a backup of the data in the snapshot, use an appropriate utility or operating system command to copy the contents of the snapshot to tape or to some other backup medium.

Example procedure to create a snapshot mirror of a volume

This example shows the steps involved in creating a snapshot mirror for the volume data_vol belonging to the disk group PRODdg.

Note: You must be logged in as superuser (root) to issue the commands in the following procedure.

To create a snapshot mirror of the volume data_vol

1 Prepare the volume data_vol for mirroring:

```
# vxsnap -g PRODdg prepare data_vol alloc=PRODdg01,PRODdg02
```

2 Verify that FastResync is enabled:

```
# vxprint -g PRODdg -F%fastresync data_vol
on
```

3. Verify that a DCO and a DCO log are attached to the volume:

   ```
   # vxprint -g PRODdg -F%hasdcolog data_vol
   ```

   ```
   on
   ```

4. Create a snapshot mirror of data_vol:

   ```
   # vxsnap -g PRODdg addmir data_vol alloc=PRODdg02
   ```

5. List the data plexes:

   ```
   # vxprint -g PRODdg -F%name -e"pl_v_name in \"data_vol\""
   ```

   ```
   data_vol-01
   ```

   ```
   data_vol-02
   ```

6. Choose the plex that is in the SNAPDONE state. Use the `vxprint -g diskgroup` command to identify the plex that is in the SNAPDONE state.

7. Identify the plex name in the above step and set the dbed_flashsnap tag for it:

   ```
   # vxedit -g PRODdg set putil2=dbed_flashsnap data_vol-02
   ```

8 Verify that the dbed_flashsnap tag has been set to the desired data plex, data_vol-02:

```
# vxprint -g PRODdg -F%name -e"pl_v_name in \"data_vol\" \
&& p2 in \"dbed_flashsnap\""
data_vol-02
```

9 To verify that the snapshot volume was created successfully, use the `vxprint -g dg` command as follows:

```
# vxprint -g PRODdg
```

The following output appears on a system running the Solaris OS.

```
TY NAME          ASSOC         KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg        PRODdg        -        -        -        -        -       -
dm PRODdg01      c1t2d0s2      -        35358848 -        -        -       -
dm PRODdg02      c1t3d0s2      -        17674896 -        -        -       -
dm PRODdg03      c1t1d0s2      -        17674896 -        -        -       -
```

```
v  data_vol     fsgen
ENABLED        4194304            -        ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED        4194304            -        ACTIVE   -       -
sd PRODdg03-01  data_vol-01
ENABLED        4194304            0        -        -       -
pl data_vol-02  data_vol
ENABLED        4194304            -        SNAPDONE -       -
sd PRODdg02-01  data_vol-02
ENABLED        4194304            0        -        -       -
dc data_vol_dco data_vol
-              -                  -        -        -       -
v  data_vol_dcl gen
ENABLED        560                -        ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl ENABLED
560         -           ACTIVE   -       -
sd PRODdg01-01  data_vol_dcl-01 ENABLED
560         0           -        -       -
pl data_vol_dcl-02 data_vol_dcl DISABLED
560         -           DCOSNP   -       -
sd PRODdg02-02  data_vol_dcl-02 ENABLED
560         0           -        -       -
```

Identify that the specified plex is in the SNAPDONE state. In this example, it is data_vol-02.

The snapshot mirror is now ready to be used.

Upgrading existing volumes to use Veritas Volume Manager 5.1

The procedure in this section describes how to upgrade a volume created using a version older than VxVM 5.0, so that it can take advantage of Database FlashSnap.

Note the following requirements and caveats for this procedure:

- The plexes of the DCO volume require persistent storage space on disk to be available. To make room for the DCO plexes, you may need to add extra disks to the disk group, or reconfigure existing volumes to free up space in the disk group. Another way to add disk space is to use the disk group move feature to bring in spare disks from a different disk group.
- Existing snapshot volumes created by the `vxassist` command are not supported. A combination of snapshot volumes created by `vxassist` and `vxsnap` are also not supported.
- You must be logged in as superuser (root) to issue the commands in the following procedure. Additionally, all operations involving the creation or modification using the commands `vxassist` or `vxdg` require that the user perform the task on the master CVM node.

To upgrade an existing volume created with an earlier version of VxVM

1. Upgrade the disk group that contains the volume, to a version 120 or higher, before performing the remainder of the procedure described in this section. Use the following command to check the version of a disk group:

   ```
   # vxdg list diskgroup
   ```

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

   ```
   # vxdg upgrade diskgroup
   ```

2. If the volume to be upgraded has a DRL plex or subdisk from an earlier version of VxVM, use the following command to remove this:

   ```
   # vxassist [-g diskgroup] remove log volume [nlog=n]
   ```

   Use the optional attribute nlog=*n* to specify the number, *n*, of logs to be removed. By default, the `vxassist` command removes one log.

3. For a volume that has one or more associated snapshot volumes, use the following command to reattach and resynchronize each snapshot:

   ```
   # vxsnap [-g diskgroup] snapback snapvol
   ```

   If persistent FastResync was enabled on the volume before the snapshot was taken, the data in the snapshot plexes is quickly resynchronized from the original volume. If persistent FastResync was not enabled, a full resynchronization is performed.

4 Use the following command to turn off persistent FastResync for the volume:

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

5 Use the following command to dissociate a DCO object from an earlier version of VxVM, DCO volume and snap objects from the volume:

```
# vxassist [-g diskgroup] remove log volume logtype=dco
```

6 Use the following command on the volume to upgrade it:

```
# vxsnap [-g diskgroup] prepare volume \
alloc="disk_name1,disk_name2"
```

Provide two disk names to avoid overlapping the storage of the snapshot DCO plex with any other non-moving data or DCO plexes.

The `vxsnap prepare` command automatically enables persistent FastResync on the volume and on any snapshots that are generated from it. It also associates a DCO and DCO log volume with the volume to be snapshot.

7 To view the existing DCO plexes and see whether there are enough for the existing data plexes, enter:

```
# vxprint -g diskgroup
```

There needs to be one DCO plex for each existing data plex.

8 If there are not enough DCO plexes for the existing data plexes, create more DCO plexes:

```
# vxsnap [-g diskgroup] addmir dco_volume_name \
[alloc=disk_name]
```

where *dco_volume_name* is the name of the DCO volume you are creating.

9 If the plex is in a SNAPDONE state, convert it to an ACTIVE state:

```
# vxplex [-g diskgroup] convert state=ACTIVE data_plex
```

10 Convert the data plexes to a SNAPDONE state and associate a DCO plex with the data plex that will be used for snapshot operations:

```
# vxplex [-g diskgroup] -o dcoplex=dco_plex_name convert \state=SNAPDONE data_plex
```

where *dco_plex_name* is the name of the DCO plex you are creating.

Example procedure to upgrade existing volumes to use Veritas Volume Manager 5.1

Note: You must be logged in as superuser (root) to issue the commands in the following procedure. Additionally, all operations involving the creation or modification using the commands `vxassist` or `vxdg` require that the user perform the task on the master CVM node.

In this example, the volume, `data_vol`, is upgraded to make use of VxVM 5.0 features.

To upgrade an existing volume created with an earlier version of VxVM

1. Upgrade the disk group, PRODdg.

   ```
   # vxdg upgrade PRODdg
   ```

2. Remove the DRL plexes or subdisks, belonging to an earlier version of VxVM, from the volume to be upgraded.

   ```
   # vxassist -g PRODdg remove log data_vol logtype=drl
   ```

3. Reattach any snapshot volume back to the primary volume to be upgraded.

   ```
   # vxsnap -g PRODdg snapback SNAP-data_vol
   ```

4. Turn off FastResync on the volume to be upgraded.

   ```
   # vxvol -g PRODdg set fastresync=off data_vol
   ```

5. Disassociate and remove any older DCO object and DCO volumes.

   ```
   # vxassist -g PRODdg remove log data_vol logtype=dco
   ```

6. Upgrade the volume by associating a new DCO object and DCO volume.

   ```
   # vxsnap -g PRODdg prepare data_vol alloc="PRODdg01,PRODdg02"
   ```

7. View the existing DCO plexes and plex state.

   Scenario 1

   In this scenario, there are enough DCO plexes for the data plexes. Also, no data plex is associated with a DCO plex.

   ```
   # vxprint -g PRODdg
   ```

The following output appears on a system running the Solaris OS.

```
TY NAME          ASSOC         KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg        PRODdg        -        -        -        -        -       -
dm PRODdg01      c1t2d0s2      -        35358848 -        -        -       -
dm PRODdg02      c1t3d0s2      -        17674896 -        -        -       -
dm PRODdg03      c1t1d0s2      -        17674896 -        -        -       -
```

```
v  data_vol     fsgen
ENABLED        4194304         -        ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED        4194304         -        ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED        4194304         0        -        -       -
pl data_vol-04  data_vol
ENABLED        4194304         -        SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED        4194304         0        -        -       -
dc data_vol_dco data_vol
-              -               -        -        -       -
v  data_vol_dcl gen
ENABLED       560              -        ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED       560              -        ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED       560              0        -        -       -
pl data_vol_dcl-02 data_vol_dcl
ENABLED       560              -        ACTIVE   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED       560              0        -        -       -
```

- Convert the data plex state from SNAPDONE to ACTIVE.

  ```
  # vxplex -g PRODdg convert state=ACTIVE data_vol-04
  ```

- Associate the data plex with a new DCO plex and convert it back to a SNAPDONE state.

  ```
  # vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 \
  convert state=SNAPDONE data_vol-04
  ```

  ```
  # vxprint -g PRODdg
  ```

  The following output appears on a system running the Solaris OS.

```
TY NAME          ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg        PRODdg       -        -        -        -        -       -
dm PRODdg01      c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02      c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03      c1t1d0s2     -        17674896 -        -        -       -

pl data_vol-03  -
DISABLED       4194304          -         -        -       -
sd PRODdg02-01  data_vol-03
ENABLED        4194304          0         -        -       -

v  data_vol     fsgen
ENABLED        4194304          -         ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED        4194304          -         ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED        4194304          0         -        -       -
pl data_vol-04  data_vol
ENABLED        4194304          -         SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED        4194304          0         -        -       -
dc data_vol_dco data_vol
-              -                -         -        -       -
v  data_vol_dcl gen
ENABLED        560              -         ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED        560              -         ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED        560              0         -        -       -
pl data_vol_dcl-02 data_vol_dcl
DISABLED       560              -         DCOSNP   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED        560              0         -        -       -
```

Scenario 2

In this scenario, there are fewer DCO plexes than data plexes.

```
# vxprint -g PRODdg
```

The following output appears on a system running the Solaris OS.

```
TY NAME          ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg        PRODdg       -        -        -        -        -       -
dm PRODdg01      c1t2d0s2     -        35358848 -        -        -       -
```

```
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -
```

```
pl data_vol-03   -
DISABLED        4194304       -        -        -       -
sd PRODdg02-01  data_vol-03
ENABLED         4194304       0        -        -       -
v  data_vol     fsgen
ENABLED         4194304       -        ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED         4194304       -        ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED         4194304       0        -        -       -
pl data_vol-04  data_vol
ENABLED         4194304       -        ACTIVE   -       -
sd PRODdg02-03  data_vol-04
ENABLED         4194304       0        -        -       -
dc data_vol_dco data_vol
-               -             -        -        -       -
v  data_vol_dcl gen
ENABLED         560           -        ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED         560           -        ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED         560           0        -        -       -
```

- Add a DCO plex to the DCO volume using the vxassist mirror command.

  ```
  # vxsnap -g PRODdg addmir data_vol_dcl alloc=PRODdg02
  ```

- Associate the data plex with the new DCO plex and convert it to a SNAPDONE state.
  The following command is used for a system running the Solaris OS.

  ```
  # vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 \
    convert state=SNAPDONE data_vol-04
  ```

  The following output appears on a system running the Solaris OS.

```
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -
```

```
pl data_vol-03   -
DISABLED        4194304       -        -        -       -
v  data_vol      fsgen
ENABLED         4194304       -        ACTIVE   -       -
pl data_vol-01   data_vol
ENABLED         4194304       -        ACTIVE   -       -
sd PRODdg01-01   data_vol-01
ENABLED         4194304       0        -        -       -
pl data_vol-04   data_vol
ENABLED         4194304       -        SNAPDONE -       -
sd PRODdg02-03   data_vol-04
ENABLED         4194304       0        -        -       -
dc data_vol_dco data_vol
-               -             -        -        -       -
v  data_vol_dcl gen
ENABLED         560           -        ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED         560           -        ACTIVE   -       -
sd PRODdg01-02   data_vol_dcl-01
ENABLED         560           0        -        -       -
pl data_vol_dcl-02 data_vol_dcl
DISABLED        560           -        DCOSNP   -       -
sd PRODdg02-02   data_vol_dcl-02
ENABLED         560           0        -        -       -
```

About creating database snapshots

A snapshot can be a source for backing up the database or creating a clone database for decision-support purposes. You can use Database FlashSnap commands to create a snapshot of your entire database on the same host (node) or on a different one.

Online database snapshots

Table 8-1 describes the three types of snapshots that can be created.

Table 8-1 Database snapshot types

Database snapshot type	Description
online	If the SNAPSHOT_MODE specified in the snapplan is set to online, the `dbed_vmsnap` command first puts the tablespaces to be snapshot into backup mode. After the snapshot is created, the tablespaces are taken out of backup mode, the log files are switched to ensure that the extra redo logs are archived, and a snapshot of the archive logs is created. Both online and offline snapshots provide a valid backup copy of the database.
offline	If the SNAPSHOT_MODE is set to offline, the database must be shut down before the snapshot is created. Online redo logs and control files are required and will be used to ensure a full database recovery.
instant	If the SNAPSHOT_MODE is set to instant, tablespaces are not put into and out of backup mode. Online redo logs and control files are required and will be used to ensure a full database recovery. Instant snapshots do not represent a valid backup copy for point-in-time recovery.

Note: For Storage Foundation for Oracle RAC, only the online snapshot mode is supported.

Database FlashSnap supports online database snapshot types.

When the SNAPSHOT_MODE specified in the snapplan is set to online, the `dbed_vmsnap`command first puts the tablespaces to be snapshot into backup mode. After the snapshot is created, the tablespaces are taken out of backup mode, the log files are switched to ensure that the extra redo logs are archived, and a snapshot of the archive logs is created.

Online snapshots provide a valid backup copy of the database

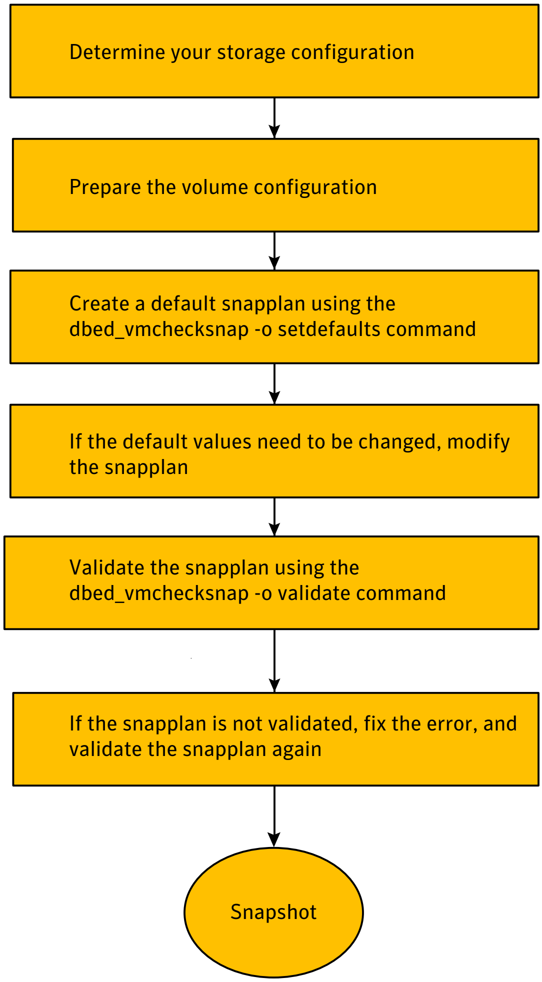
Tasks before creating a snapshot

Review the details on how to create snapshots of all volumes on a database using the snapplan.

Optionally, you can use the VxVM command (`vxsnap`) to create volume snapshots. However, unlike the Database FlashSnap commands, the `vxsnap` command does not automate disk group content reorganization functions.

Figure 8-3 depicts the sequence of steps leading up to taking a snapshot using Database FlashSnap.

Figure 8-3 Prerequisites for creating a snapshot of your database



Creating a snapshot

Make sure the volumes used by the database are configured properly before attempting to take a snapshot. This database configuration requires superuser (root) privileges.

Note: Database FlashSnap commands must be run by the Oracle database administrator.

Whenever you change the structure of the database (for example, by adding or deleting datafiles, converting PFILE to SPFILE, or converting SPFILE to PFILE), you must run `dbed_update`. For example:

```
$ /opt/VRTS/bin/dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

To create a snapshot image of a database

1. Create a snapshot mirror of a volume or volume set.

   See "To create a snapshot mirror of a volume or volume set" on page 154.

2. Use the `dbed_vmchecksnap` command to create a snapplan template and check the volume configuration to ensure that it is valid for creating volume snapshots of the database.

   The snapplan contains detailed database and volume configuration information that is needed for snapshot creation and resynchronization. You can modify the snapplan template with a text editor.

   The `dbed_vmchecksnap` command can also be used to:

List all snapplans associated with a specific ORACLE_SID	`dbed_vmchecksnap -o list`
Remove the snapplan from the SFDB repository	`dbed_vmchecksnap -o remove -f SNAPPLAN`
Copy a snapplan from the SFDB repository to your local directory	`dbed_vmchecksnap -o copy -f SNAPPLAN`

   See "Creating a snapplan (dbed_vmchecksnap)" on page 175.

3. Use the `dbed_vmsnap` command to create snapshot volumes for the database.

   See "Creating a snapshot (dbed_vmsnap)" on page 191.

4. On the secondary host, use the `dbed_vmclonedb` command to create a clone database using the disk group deported from the primary host. For more information:

   See "Cloning a database (dbed_vmclonedb)" on page 199.

   If the primary and secondary hosts specified in the snapplan are different, the `dbed_vmclonedb` command takes the following actions:

   - Imports the disk group that was deported from the primary host

- Recovers the snapshot volumes
- Mounts the file systems
- Recovers the database
- Brings the database online with a different Oracle SID name than the primary host.

You can use the `-o recoverdb` option to let `dbed_vmclonedb` perform an automatic database recovery, or you can use the `-o mountdb` option to perform your own point-in-time recovery and bring up the database manually. For a point-in-time recovery, the snapshot mode must be online.

You can also create a clone on the primary host. Your snapplan settings specify whether a clone should be created on the primary or secondary host.

5 You can now use the clone database to perform database backup and other off-host processing work.

6 For single instance Oracle, the clone database can be used to reverse resynchronize the original volume from the data in the snapshot, or can be discarded by rejoining the snapshot volumes with the original volumes (that is, by resynchronizing the snapshot volumes) for future use.

7 The clone database can be discarded by rejoining the snapshot volumes with the original volumes (that is, by resynchronizing the snapshot volumes) for future use.

Tasks after creating a snapshot

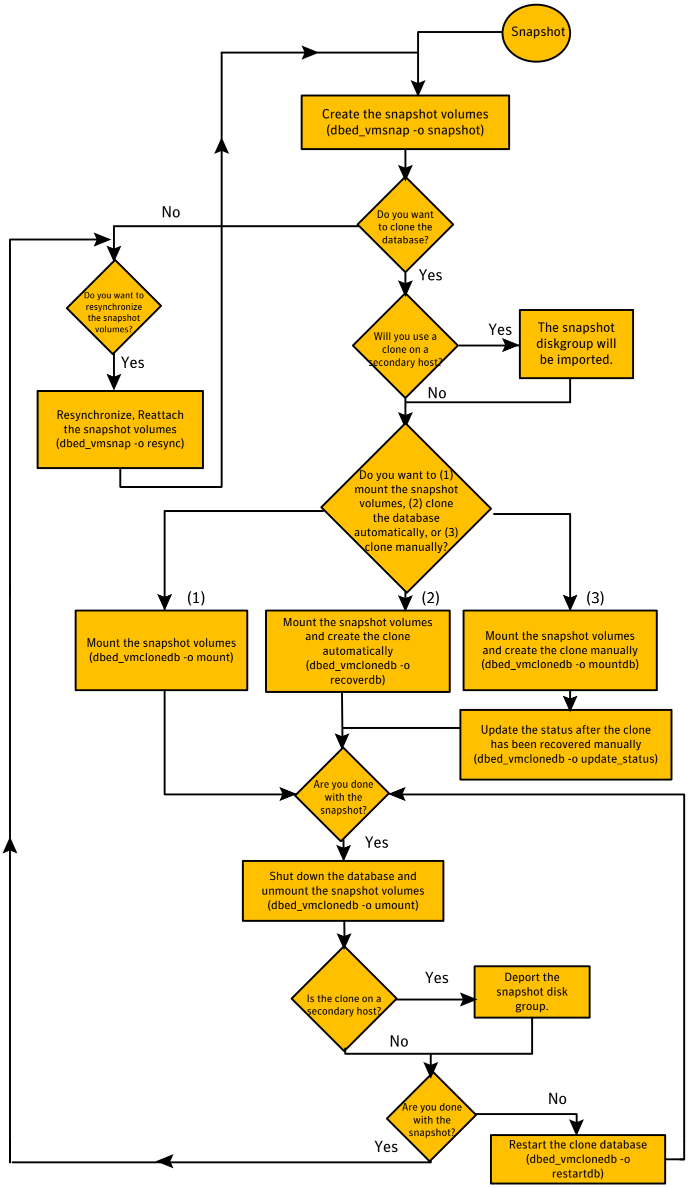
There are many actions you can take after creating a snapshot of your database using Database FlashSnap. You can create a clone of the database for backup and off-host processing purposes. You can resynchronize the snapshot volumes with the primary database.

For single instance Oracle, in the event of primary database failure, you can recover it by reverse resynchronizing the snapshot volumes in the event of primary database failure.

The following flow chart depicts the actions you can perform after creating a snapshot of your database using Database FlashSnap, and involve the following four questions:

Figure 8-4 is a flow chart that depicts the actions you can perform after creating a snapshot of your database using Database FlashSnap.

Figure 8-4 Actions you can perform after creating a snapshot of your database

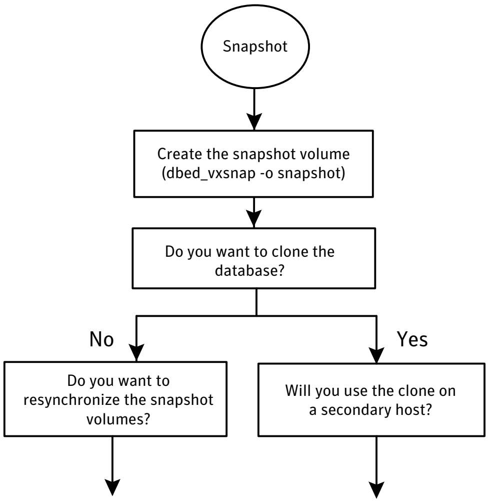


Do you want to clone the database?

After creating the snapshot volumes, you can proceed to clone the database and use the clone on a secondary host or resynchronize the snapshot volumes.

Figure 8-5 is a flow chart of this process.

Figure 8-5 Clone the database

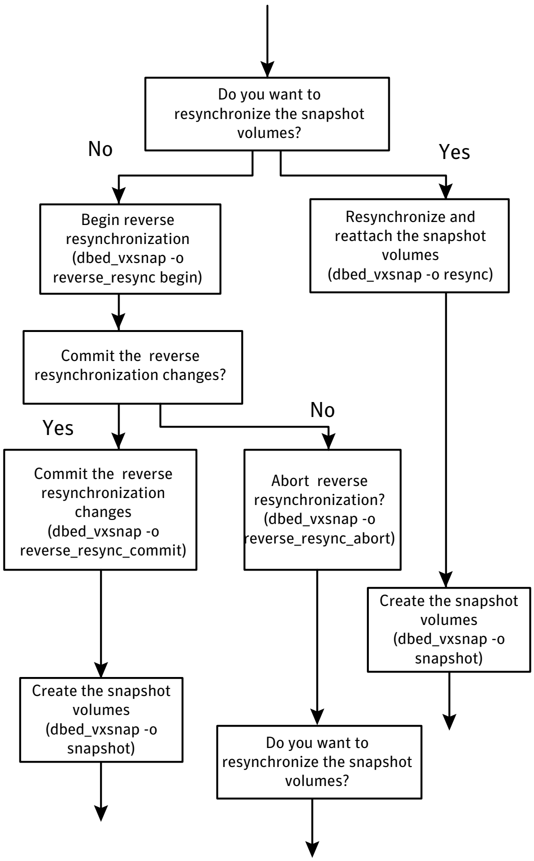


Do you want to resynchronize the snapshot volumes?

If you decide not to clone the database, then you can proceed to resynchronize and reattach the snapshot volumes or begin a reverse resynchronization process.

Figure 8-6 is a flow chart of this process.

Figure 8-6 Resynchronize the database



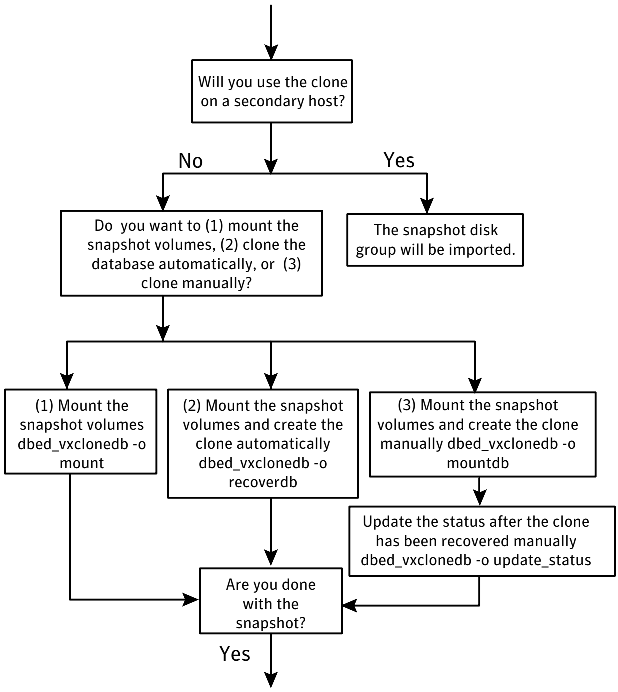
Will you use the clone on a secondary host?

If you decide to clone the database and use the clone on a secondary host, then the snapshot diskgroup will be imported. If not, then proceed with one of the following three options:

- Mount the snapshot volumes
- Mount the snapshot volumes and create the clone automatically
- Mount the snapshot volumes and create the clone manually

Figure 8-7 is a flow chart of this process.

Figure 8-7 Clone on a secondary host?

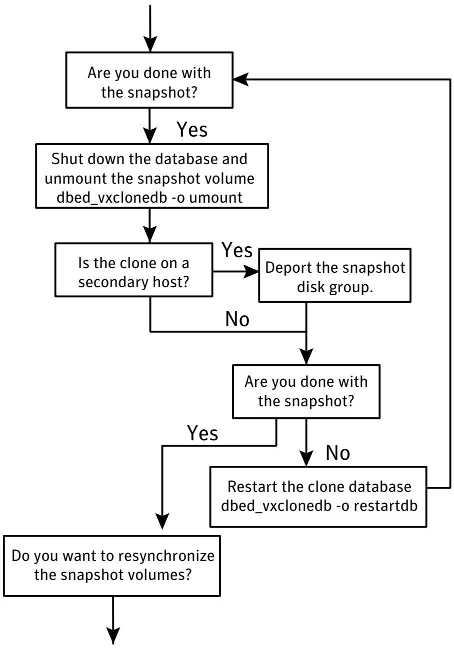


Are you done with the snapshot?

If you are done with the snapshot, then shut down the database and unmount the snapshot volumes. If the clone is on a secondary host, then deport the snapshot diskgroup.

Figure 8-8 is a flow chart of these processes.

Figure 8-8 Finished with the snapshot



FlashSnap commands

Database FlashSnap actions can be performed by using the following FlashSnap commands described in the following sections:

- Creating a snapplan (dbed_vmchecksnap)
- Validating a snapplan (dbed_vmchecksnap)
- Displaying, copying, and removing a snapplan (dbed_vmchecksnap)
- Creating a snapshot (dbed_vmsnap)
- Backing up the database from snapshot volumes (dbed_vmclonedb)
- Cloning a database (dbed_vmclonedb)
- Resynchronizing the snapshot to your database
- Removing a snapshot volume

Creating a snapplan (dbed_vmchecksnap)

The `dbed_vmchecksnap` command creates a snapplan that `dbed_vmsnap` uses to create a snapshot of an Oracle database.

The snapplan specifies snapshot scenarios: online, offline, or instant.

You can name a snapplan file whatever you choose. Each entry in the snapplan file is a line in parameter=argument format.

Table 8-2 describes the parameters that can be set when using the `dbed_vmchecksnap` command to create or validate a snapplan.

Table 8-2 Parameter values for dbed_vmchecksnap

Parameter	Value
SNAPSHOT_VERSION	Specifies the snapshot version for this major release of SF Enterprise products.
PRIMARY_HOST	The name of the host where the primary database resides.
SECONDARY_HOST	The name of the host where the database will be imported.
PRIMARY_DG	Colon-seperated list of names of the VxVM disk group's used by the primary database
SNAPSHOT_DG_PREFIX	The name of the prefix attached to the disk group name. A snapshot disk group's name is a concatination of SNAPSHOT_DG_PREFIX and the corresponding primary dg name. Its default value will be "SNAP_". The snapshot volumes will be put into this disk group on the primary host and deported. The secondary host will import this disk group to start a clone database. SNAPSHOT_DG is not supported in release 5.1. It is replaced by SNAPSHOT_DG_PREFIX.
ORACLE_SID	The name of the Oracle database. By default, the name of the Oracle database is included in the snapplan.

Table 8-2 Parameter values for dbed_vmchecksnap *(continued)*

Parameter	Value
ARCHIVELOG_DEST	The full path of the archive logs. There are several archive log destinations that can be used for database recovery if you are multiplexing the archive logs. You must specify which archive log destination to use. It is recommended that you have the archive log destination on a separate volume if SNAPSHOT_ARCHIVE_LOG is yes.
SNAPSHOT_ARCHIVE_LOG	yes or no Specifies whether to create a snapshot of the archive log volumes. Specify yes to split the archive log volume mirrors and deport them to the secondary host. When using the Oracle remote archive log destination feature to send the archive logs to the secondary host, you can specify no to save some space. Because the archive logs may not always be delivered to the secondary host reliably, it is recommended that you specify yes.

Table 8-2 Parameter values for dbed_vmchecksnap *(continued)*

Parameter	Value
SNAPSHOT_MODE	Specifies the database snapshot mode. Values can be online, offline, or instant. ■ If the snapshot is created while the database is online, the `dbed_vmsnap` command will put the tablespaces into backup mode. After `dbed_vmsnap` finishes creating the snapshot, it will take the tablespaces out of backup mode, switch the log files to ensure that the extra redo logs are archived, and create a snapshot of the archived logs. ■ If the database if offline, it is not necessary to put the tablespaces into backup mode. The database must be shut down before creating an offline snapshot. ■ If the database snapshot is instant, `dbed_vmsnap` will skip putting the tablespace into backup mode.doctype_adv_ora **Note:** If SNAPSHOT_MODE is set to offline or instant, an off-host configuration is required and the `-r` *`relocate_path`* option is not allowed. **Note:** The offline and instant snapshot modes are not supported for Oracle RAC.
SNAPSHOT_PLAN_FOR	The default value is database and cannot be changed. Specifies the database object for which you want to create a snapshot.
SNAPSHOT_PLEX_TAG	Specifies the snapshot plex tag. Use this variable to specify a tag for the plexes to be snapshot. The maximum length of the `plex_tag` is 15 characters. The default plex tag is `dbed_flashsnap`.
SNAPSHOT_VOL_PREFIX	Specifies the snapshot volume prefix. Use this variable to specify a prefix for the snapshot volumes split from the primary disk group. A volume name cannot be more than 32 characters. You should consider the length of the volume name when assigning the prefix.

Table 8-2 Parameter values for dbed_vmchecksnap *(continued)*

Parameter	Value
ALLOW_REVERSE_RESYNC	yes or no By default, reverse resynchronization is off (set equal to no). If it is set to yes, data from the snapshot volume can be used to update the primary volume. **Note:** This parameter must be set to no for Oracle RAC.
SNAPSHOT_MIRROR	Specifies the number of plexes to be snapshot. The default value is 1.
DG:*VOL*	Optional These entries are created by default if the `dbed_vmchecksnap` command is run with the `-o setdefaults -m` option. These entries specify the mount path for the associated snapshot volumes when we perform a clone operation using dbed_vmclondb. The values for these fields can be left blank, if they are blank then the `-r relocate_path` needs to be specified when performing a `dbed_vmclonedb` operation.

When you first run `dbed_vmchecksnap`, use the `-o setdefaults` option to create a snapplan using default values for variables. You may then edit the file manually to set the variables for different snapshot scenarios.

Before creating a snapplan, make sure the following conditions have been met:

Prerequisites	■ Storage must be configured as specified: See "Preparing hosts and storage for Database FlashSnap" on page 149. ■ You must be the Oracle database administrator. ■ The disk group must be version 110 or later. For more information on disk group versions, see the vxdg(1M) manual page. ■ Be sure that a DCO and DCO volume are associated with the volume for which you are creating the snapshot. ■ Snapshot plexes and their associated DCO logs should be on different disks than the original plexes, and should be configured correctly for creating snapshots by the system administrator. ■ Persistent FastResync must be enabled on the existing database volumes and disks must be assigned for the snapshot volumes. ■ The database must be running in archive log mode. Archive log mode is set in the Oracle initialization parameter file. ■ The Oracle database must have at least one mandatory archive destination. ■ ORACLE_HOME cannot reside on disk which will be used for snapshot. ■ The Oracle database files and archive log files should use different volumes with unique disks in same disk group.
Usage Notes	■ The snapplan must be created on the primary host. ■ After creating the snapplan using the dbed_vmchecksnap command, you can use a text editor to review and update the file, if necessary. ■ It is recommended that you create a local working directory to store your snapplans in. ■ See the dbed_vmchecksnap (1M) online manual page for more information. ■ If the SNAPSHOT_MODE for the database is set to online, the primary and secondary hosts can be the same. ■ If the SNAPSHOT_MODE is set to offline or instant, the primary and secondary hosts must be different.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To create a snapplan

1. Change directories to the working directory you want to store your snapplan in.

   ```
   $ cd /working_directory
   ```

2. Create a snapplan with default values using the `dbed_vmchecksnap` command:

   ```
   $ /opt/VRTS/bin/dbed_vmchecksnap -S  ORACLE_SID \
   -H ORACLE_HOME -f SNAPPLAN -o setdefaults -t  host_name \
   [-p PLEX_TAG] [-m]
   ```

   Example output for using `multi-dg` and for when when you specify `-m` option for mapped mount point.

   ```
   $ /opt/VRTS/bin/dbed_vmchecksnap -S slave -H $ORACLE_HOME -f snapplan3 -o s
   -t mensa -m
   Snapplan snapplan3 for slave.
   =====================================================
   SNAPSHOT_VERSION=5.0
   PRIMARY_HOST=mensa.veritas.com
   SECONDARY_HOST=mensa
   PRIMARY_DG=archdg1:datadg1:datadg2
   ORACLE_SID=slave
   ARCHIVELOG_DEST=/standby/oraarch1_1/slave
   SNAPSHOT_ARCHIVE_LOG=yes
   SNAPSHOT_MODE=online
   SNAPSHOT_PLAN_FOR=database
   SNAPSHOT_PLEX_TAG=dbed_flashsnap
   SNAPSHOT_DG_PREFIX=SNAP_
   SNAPSHOT_VOL_PREFIX=SNAP_
   ALLOW_REVERSE_RESYNC=no
   SNAPSHOT_MIRROR=1
   archdg1:arch1vol1=
   datadg1:data1vol2=
   datadg1:data1vol1=
   datadg2:data1vol1=
   datadg2:data1vol2=
   ```

3. Open the snapplan file in a text editor and modify it as needed.

Example snapplans created for a snapshot image

In this example, a snapplan, snap1, is created for a snapshot image in a same-node configuration and default values are set. The host is named host1 and the working directory is /export/snap_dir.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$ cd /export/snap_dir
```

```
$ $ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/orahome -f snap1 -o setdefaults -t host1
```

Example ouput for a single disk group environment:

```
Snapplan snap1 for PROD.
=====================================================
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/prod_ar
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Example output for an environment with multiple disk groups:

```
Snapplan snap1 for PROD.
=====================================================
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg1:PRODdg2
ORACLE_SID=PROD
ARCHIVELOG_DEST=/prod_ar
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
```

```
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

In this second example, a snapplan, snap2, is created for a snapshot image in a two-node in the cluster configuration, and default values are set. The primary host is host1, the secondary host is host2, and the working directory is /export/snap_dir.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$cd /export/snap_dir
```

```
$/opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/orahome -f snap2 -o setdefaults -t host2
```

Example ouput for a single disk group environment:

```
Snapplan snap2 for PROD.
=====================================================
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/mytest/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Example output for an environment with multiple disk groups:

```
Snapplan snap2 for PROD.
=====================================================
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
```

```
PRIMARY_DG=PRODdg1:PRODdg2
ORACLE_SID=PROD
ARCHIVELOG_DEST=/mytest/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

By default, a snapplan's SNAPSHOT_PLEX_TAG value is set as dbed_flashsnap. You can use the `-p` option to assign a different tag name. Make use of the `-p` option when creating the snapplan with the `setdefaults` option.

In the following example, the `-p` option is used with `setdefaults` to assign my_tag as the SNAPSHOT_PLEX_TAG value.

```
$ dbed_vmchecksnap -S PROD -H $ORACLE_HOME -O setdefaults \
-p my_tag -f snap1 -t host2
```

Example ouput for a single disk group environment:

```
Snapplan snap1 for PROD
=====================================================
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch_data
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=my_tag
SNAPSHOT_DG_PREFIX=SNAP
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Example output for an environment with multiple disk groups:

```
Snapplan snap1 for PROD
=====================================================
```

```
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
PRIMARY_DG=PRODdg1:PRODdg2
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch_data
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=my_tag
SNAPSHOT_DG_PREFIX=SNAP
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Creating multi-mirror snapshots

To make the Database Snapshots highly available, the snapped snapshot volume should contain more than one mirror. This makes the snapshot volumes available even if one of the mirrors becomes disabled. Snapshot volumes can be mounted and the entire database snapshot is usable even if one of the mirrors becomes disabled. The multi-mirror snapshots are enabled by `SNAPSHOT_MIRROR=<n>` in the snapplan.

Note: There are no changes to the Command Line usage or arguments for the Flashsnap tools.

Before taking the snapshot, make sure all tagged snapshot mirrors are in SNAPDONE state.

For information about snapshot mirrors, refer to the *Veritas Volume Manager Administrator's Guide*.

Validating a snapplan (dbed_vmchecksnap)

After creating a snapplan, the next steps are to validate the snapplan parameters and check whether the snapshot volumes have been configured correctly for creating snapshots. If validation is successful, the snapplan is copied to the repository. The snapplan is validated using the `dbed_vmchecksnap` command with the `-o validate` option.

Consider the following prerequisites and notes before validating a snapplan:

Prerequisites	■ The database must be up and running while executing the `dbed_vmchecksnap` command.
Usage Notes	■ The `dbed_vmchecksnap` command must be run as the Oracle database administrator. ■ When using `dbed_vmchecksnap -o validate` to validate the snapplan and storage, you can save the validation output. The system administrator can use this information to adjust the storage setup if the validation fails. ■ If a snapplan is updated or modified, you must re-validate it. It is recommended that snapplans are revalidated when changes are made in the database disk group. ■ See the `dbed_vmchecksnap`(1M) manual page for more information.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To validate a snapplan

1 Change directories to the working directory your snapplan is stored in:

```
$ cd /working_directory
```

2 Validate the snapplan using the `dbed_vmchecksnap` command:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S  ORACLE_SID \
-H ORACLE_HOME -f SNAPPLAN -o validate
```

Example to validate a snapplan snap1 for a snapshot image

In the following example, a snapplan, snap1, is validated for a snapshot image in a same-node configuration. The primary host is host1 and the working directory is /export/snap_dir.

Note: You must issue commands as an Oracle database administrator in the following procedure.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/orahome \
-f snap1 -o validate
```

Example output for a single disk group environment:

```
PRIMARY_HOST is host1
SECONDARY_HOST is host1
The version of PRIMARY_DG-PRODdg is 110.
The primary diskgroup PRODdg is a shared disk group
SNAPSHOT_DG is SNAP_PRODdg

SNAPSHOT_MODE is online

The database is running in archivelog mode.

ARCHIVELOG_DEST is /prod_ar
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/prod_ar is mount on /dev/vx/dsk/PRODdg/prod_ar.

Examining Oracle volume and disk layout for snapshot

Volume prod_db on PRODdg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODdg01.
Snapshot plex and DCO log for prod_db is on PRODdg02.
SNAP_PRODdg for snapshot will include: PRODdg02
ALLOW_REVERSE_RESYNC is no

The snapplan snap1 has been created.
```

Example output for an environment with multiple disk groups:

```
PRIMARY_HOST is host1
SECONDARY_HOST is host1
The version of PRIMARY_DG-PRODdg1 is 140.
SNAPSHOT_DG is SNAP_PRODdg1
The version of PRIMARY_DG-PRODdg2 is 140.
SNAPSHOT_DG is SNAP_PRODdg2
SNAPSHOT_MODE is online
The database is running in archivelog mode.
ARCHIVELOG_DEST is /archvol
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
```

```
ARCHIVELOG_DEST=/archvol is mount on /dev/vx/dsk/archdg/archvol.
Examining Oracle volume and disk layout for snapshot.
Volume prodvol1 on PRODdg1 is ready for snapshot.
Original plex and DCO log for prodvol1 is on PRODdisk11.
Snapshot plex and DCO log for prodvol1 is on PRODdisk12.
SNAP_PRODdg1 for snapshot will include: PRODdisk11 PRODdisk12
Examining Oracle volume and disk layout for snapshot.
Volume prodvol2 on PRODdg2 is ready for snapshot.
Original plex and DCO log for prodvol2 is on PRODdisk21.
Snapshot plex and DCO log for prodvol2 is on PRODdisk22.
SNAP_PRODdg2 for snapshot will include: PRODdisk21 PRODdisk22
ALLOW_REVERSE_RESYNC is no
The snapplan snap1 has been created.
```

In the following example, a snapplan, snap2, is validated for a snapshot image in a off-host configuration. The primary host is host1, the secondary host is host2, and the working directory is /export/snap_dir.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H \
/oracle/product/orahome -f snap2 -o validate
```

Example output for a single disk group environment:

```
PRIMARY_HOST is host1
SECONDARY_HOST is host2
The version of PRIMARY_DG-PRODdg is 110.
The primary diskgroup PRODdg is a shared disk group
SNAPSHOT_DG is SNAP_PRODdg
SNAPSHOT_MODE is online

The database is running in archivelog mode.

ARCHIVELOG_DEST is /mytest/arch
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/mytest/arch is mount on
/dev/vx/dsk/PRODdg/arch.

Examining Oracle volume and disk layout for snapshot.
Volume arch on PRODdg is ready for snapshot.
Original plex and DCO log for arch is on PRODdg01.
```

```
Snapshot plex and DCO log for arch is on PRODdg02.

Volume prod_db on PRODdg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODdg01.
Snapshot plex and DCO log for prod_db is on PRODdg04.

SNAP_PRODdg for snapshot will include: PRODdg02

ALLOW_REVERSE_RESYNC is no

The snapplan snap2 has been created.
```

Example output for an environment with multiple disk groups:

```
PRIMARY_HOST is host1
SECONDARY_HOST is host2
The version of PRIMARY_DG-PRODdg1 is 140.
SNAPSHOT_DG is SNAP_PRODdg1
The version of SECONDARY_DG-PRODdg2 is 140.
SNAPSHOT_DG is SNAP_PRODdg2
SNAPSHOT_MODE is online
The database is running in archivelog mode.
ARCHIVELOG_DEST is /archvol
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/archvol is mount on /dev/vx/dsk/archdg/archvol.
Examining Oracle volume and disk layout for snapshot.
Volume prodvol1 on PRODdg1 is ready for snapshot.
Original plex and DCO log for prodvol1 is on PRODdisk11.
Snapshot plex and DCO log for prodvol1 is on PRODdisk12.
SNAP_PRODdg1 for snapshot will include: PRODdisk11 PRODdisk12
Examining Oracle volume and disk layout for snapshot.
Volume prodvol2 on PRODdg2 is ready for snapshot.
Original plex and DCO log for prodvol2 is on PRODdisk21.
Snapshot plex and DCO log for prodvol2 is on PRODdisk22.
SNAP_PRODdg2 for snapshot will include: PRODdisk21 PRODdisk22
ALLOW_REVERSE_RESYNC is no
The snapplan snap1 has been created.
```

Displaying, copying, and removing a snapplan (dbed_vmchecksnap)

Consider the following usage notes before listing all snapplans for a specific Oracle database, displaying a snapplan file, or copying and removing snapplans.

Usage Notes	■ If the local snapplan is updated or modified, you must revalidate it. ■ If the database schema or disk group is modified, you must revalidate it after running `dbed_update`.

Displaying a snapplan

You can use the `dbed_vmchecksnap` command to list all available snapplans and to display detailed information for a particular snapplan.

To list all available snapplans for a specific Oracle database

◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -o list
```

In the following example, all available snapplans are listed for the database PROD.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -o list
The following snapplan(s) are available for PROD:

SNAP_PLAN                           SNAP_STATUS
DB_STATUS                           SNAP_READY
snap1                               init_full
init                                yes
snap2                               init_full
init                                yes
```

The command output displays all available snapplans, their snapshot status (SNAP_STATUS), database status (DB_STATUS), and whether a snapshot may be taken (SNAP_READY).

For Database FlashSnap status information:

See "About Database FlashSnap status information" on page 335.

To display detailed information for a snapplan

- Use the `dbed_vmchecksnap` command as follows:

  ```
  $ /opt/VRTS/bin/dbed_vmchecksnap -S \
  ORACLE_SID -f SNAPPLAN -o list
  ```

  In the following example, the snapplan snap1 is displayed.

  ```
  $ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o list

  SNAPSHOT_VERSION=5.0
  PRIMARY_HOST=host1
  SECONDARY_HOST=host1
  SNAPSHOT_DG_PREFIX=SNAP_
  ORACLE_SID=PROD
  ARCHIVELOG_DEST=/prod_ar
  SNAPSHOT_ARCHIVE_LOG=yes
  SNAPSHOT_MODE=online
  SNAPSHOT_PLAN_FOR=database
  SNAPSHOT_PLEX_TAG=dbed_flashsnap
  SNAPSHOT_VOL_PREFIX=SNAP_
  ALLOW_REVERSE_RESYNC=no
  SNAPSHOT_MIRROR=1

  STORAGE_INFO
  PRODdg02
  SNAP_PLEX=prod_ar-02
  STATUS_INFO
  SNAP_STATUS=init_full
  DB_STATUS=init
  ```

Copying a snapplan

If you want to create a snapplan similar to an existing snapplan, you can simply create a copy of the existing snapplan and modify it. To copy a snapplan from the SFDB repository to your current directory, the snapplan must not already be present in the current directory.

To copy a snapplan from the SFDB repository to your current directory

- Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S  ORACLE_SID \
-f SNAPPLAN -o copy
```

  In the following example, the snapplan, snap1, is copied from the VxDBA repository to the current directory.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o  copy
Copying 'snap1' to '/export/snap_dir'
```

Removing a snapplan

A snapplan can be removed from a local directory or repository if the snapplan is no longer needed.

To remove a snapplan from the SFDB repository

- Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -f\
SNAPPLAN -o remove
```

  In the following example, the snapplan, snap1, is removed from the SFDB repository.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o  remove
The snapplan snap1 has been removed.
```

Creating a snapshot (dbed_vmsnap)

The `dbed_vmsnap` command creates a snapshot of an Oracle database by splitting the mirror volumes used by the database into a snapshot database. You can use the snapshot image on either the same host as the database or on a secondary host provided storage is shared by the two hosts.

The snapshot image created by `dbed_vmsnap` is a frozen image of an Oracle database's datafiles. The `dbed_vmsnap` command ensures that a backup control file is created when the snapshot database is created, which allows for complete data recovery, if needed.

For Database FlashSnap status information:

See "About Database FlashSnap status information" on page 335.

Prerequisites	■ You must be logged in as the Oracle database administrator. ■ You must create and validate a snapplan using `dbed_vmchecksnap` before you can create a snapshot image with `dbed_vmsnap`.
Usage Notes	■ The `dbed_vmsnap` command can only be used on the primary host. ■ Do not share volumes between Oracle database files and other software. ■ When creating a snapshot volume, create the snapshot on a separate controller and on separate disks from the primary volume. ■ Make sure your archive log destination is separate from your Oracle database volumes. ■ Do not place any datafiles, including control files, in the $ORACLE_HOME/dbs directory. ■ Resynchronization speed varies based on the amount of data changed in both the primary and secondary volumes when the mirror is broken off. ■ See the `dbed_vmsnap(1M)` manual page for more information.

Note the following points:

- To force snapshot creation, use the `-F` option. The `-F` option can be used after a snapshot operation has failed and the problem was fixed without using SFDB commands. (That is, the volumes were synchronized using VxVM commands.) In this situation, the status of the snapplan will appear as unavailable for creating a snapshot. The `-F` option ignores the unavailable status, checks for the availability of volumes, and creates the snapshot after the volumes pass the availability check.
- After the snapshot is created, `dbed_vmsnap` returns values you will need to run `dbed_vmclonedb`. These values include the snapshot disk group, the snapplan name, and the SFDB repository volume for an off-host configuration. Make a note of these values so you have them when running `dbed_vmclonedb`.
- You can also use the command `dbed_vmchecksnap -f snapplan -o list` to access the information regarding the snapshot disk group, the snapplan name, and the SFDB repository.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To create a snapshot

1 Change directories to the working directory in which your snapplan is stored:

```
$ cd /working_directory
```

2 Create the snapshot image using the `dbed_vmsnap` command.

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \
-o snapshot [-F]
```

The snapshot volumes now represent a consistent backup copy of the database. You can backup the database by copying the snapshot volumes to tape or other backup media.

See "Backing up the database from snapshot volumes (dbed_vmclonedb)" on page 194.

3 You can also create another Oracle database for decision-support purposes.

See "Cloning a database (dbed_vmclonedb)" on page 199.

Example to create a snapshot image of the database PROD

In this example, a snapshot image of the database, PROD, is created for a same-node configuration. In this case, the SECONDARY_HOST parameter is set the same as the PRIMARY_HOST parameter in the snapplan.

Note: You must issue commands as an Oracle database administrator in the following procedure.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot

dbed_vmsnap started at 2006-03-02 14:15:27
VxDBA repository is up to date.
The database is running in archivelog mode.
A snapshot of ORACLE_SID PROD is in DG SNAP_PRODdg.
Snapplan snap1 is used for the snapshot.

If -r <relocate_path> is used in dbed_vmclonedb, make sure
<relocate_path> is created and owned by Oracle DBA. Otherwise,
the following mount points need to be created and owned by
Oracle DBA:

      /prod_db.
```

```
      /prod_ar.

dbed_vmsnap ended at 2006-03-02 14:16:11
```

In this example, a snapshot image of the primary database, PROD, is created for an off-host configuration. In this case, the SECONDARY_HOST parameter specifies a different host name than the PRIMARY_HOST parameter in the snapplan.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap2 -o snapshot

dbed_vmsnap started at 2005-03-02 23:01:10
VxDBA repository is up to date.
The database is running in archivelog mode.
A snapshot of ORACLE_SID PROD is in DG SNAP_PRODdg.
Snapplan snap2 is used for the snapshot.
VxDBA repository volume is SNAP_arch.

If -r <relocate_path> is used in dbed_vmclonedb, make sure
<relocate_path> is created and owned by Oracle DBA. Otherwise,
the following mount points need to be created and owned by

Oracle DBA:

      /prod_db.
      /prod_ar.

dbed_vmsnap ended at 2005-03-02 23:02:58
```

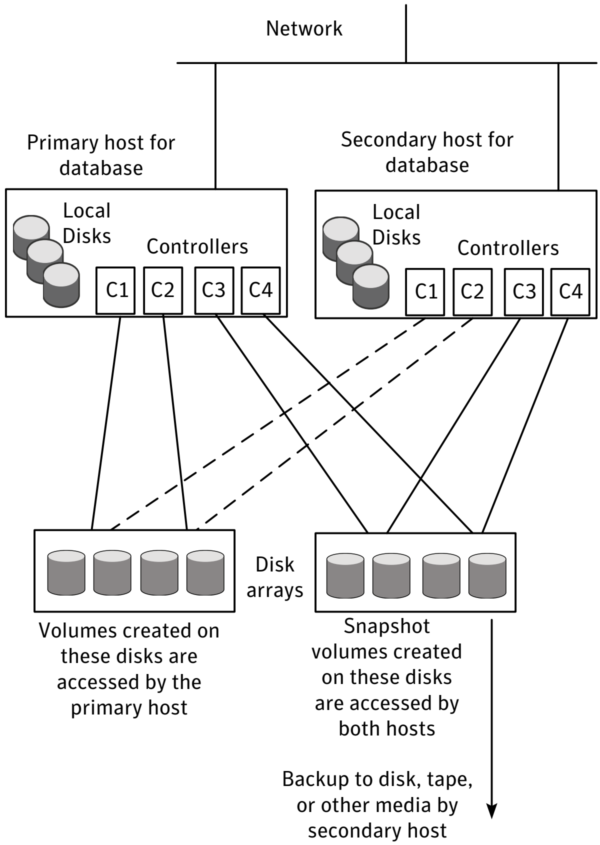
Backing up the database from snapshot volumes (dbed_vmclonedb)

Snapshots are most commonly used as a source for backing up a database. The advantage of using snapshot volumes is that the backup will not contest the I/O bandwidth of the physical devices. Making the snapshot volumes available on a secondary host will eliminate the extra loads put on processors and I/O adapters by the backup process on the primary host.

A clone database can also serve as a valid backup of the primary database. You can back up the primary database to tape using snapshot volumes.

Figure 8-9 shows a typical configuration when snapshot volumes are used on a secondary host.

Figure 8-9 Example system configuration for database backup on a secondary host



Prerequisites	■ You must be logged in as the Oracle database administrator to use `dbed_vmclonedb` command. ■ Before you can use the `dbed_vmclonedb` command, you must validate a snapplan and create a snapshot. See "About creating database snapshots" on page 165. See "Validating a snapplan (dbed_vmchecksnap)" on page 184. See "Creating a snapshot (dbed_vmsnap)" on page 191. ■ The volume snapshot must contain the entire database. ■ Before you can use the `dbed_vmclonedb` command with the `-r` *`relocate_path`* option (which specifies the initial mount point for the snapshot image), the system administrator must create the mount point and then change the owner to the Oracle database administrator. ■ In case of mapped mounts, the mapped mount points need to be created by the System Administrator.
Usage Notes	■ The `dbed_vmclonedb` command can be used on the secondary host. ■ In a same-node configuration, the primary and secondary hosts are the same. ■ In a same-node configuration, `-r` *`relocate_path`* is required if no mapped mounts were specified or some of the mapped mount entries were left blank in the snapplan. ■ If SNAPSHOT_MODE is set to offline or instant, an off-host configuration is required and `-r` *`relocate_path`* is not allowed. ■ See the `dbed_vmclonedb`(1M) manual page for more information.

Mounting the snapshot volumes and backing up

Before using the snapshot volumes to do a backup, you must first mount them.

Note: You must issue commands as an Oracle database administrator in the following procedure.

Note: If you use the Oracle online backup method, you must also back up all the archived log files in order to do a complete restore and recovery of the database.

To mount the snapshot volumes

◆ Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID \
  -o mount,new_sid=new_sid,server_name=svr_name -f SNAPPLAN [-H ORACLE_HOME] \
  [-r relocate_path]
```

You can now back up an individual file or a group of files under a directory onto the backup media.

In this example, snapshot volumes are mounted.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
  -o mount,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone/single
```

```
dbed_vmclonedb started at 2004-04-02 15:35:41
Mounting /clone/single/prod_db on
/dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
Mounting /clone/single/prod_ar on
/dev/vx/dsk/SNAP_PRODdg/SNAP_prod_ar.
dbed_vmclonedb ended at 2004-04-02 15:35:50
```

The following is an example of creating a snapshot on the same host (host nobody):

```
$  /opt/VRTS/bin/dbed_vmclonedb -S PROD \
 -o mount,new_sid=NEWPROD,server_name=nobody -f snap1 -r /clone/single
```

```
dbed_vmclonedb started at 2006-10-24 10:44:54
Mounting /clone/single/archivelogs on /dev/vx/dsk/SNAP_PRODdg/SNAP_archvol.
Mounting /clone/single/oradata on /dev/vx/dsk/SNAP_PRODdg/
SNAP_ora_data_vol.
dbed_vmclonedb ended at 2006-10-24 10:45:49
```

Note: A usage error is displayed if the server_name is not given in the above command.

To mount a Storage Checkpoint carried over from the snapshot volumes to a secondary host

1. On the secondary host, list the Storage Checkpoints carried over from the primary database using the `dbed_ckptdisplay` command.

   For example:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S ORACLE_SID -n
```

2. You can mount one of the listed Storage Checkpoints using the `dbed_ckptmount` command.

   For example:

```
$ /opt/VRTS/bin/dbed_ckptmount -S ORACLE_SID -c CKPT_NAME \
  -m MOUNT_POINT
```

   Note the following limitations:

   - Any mounted Storage Checkpoints must be unmounted before running the following commands:

```
$ /opt/VRTS/bin/dbed_ckptumount -S ORACLE_SID -c CKPT_NAME
```

   - It is only possible to mount a Storage Checkpoint carried over with the snapshot volumes in an off-host configuration if the snapshot volumes were mounted with the `dbed_vmclonedb` command with the `-o mount` option without the use of `-r relocate_path`.
   - Storage Checkpoints carried over with the snapshot volumes can be mounted before a clone database is created using `dbed_vmclonedb` with the `-o mount` option. After a clone database is created using `dbed_vmclonedb` with the `-o recoverdb` option, however, Storage Checkpoints are no longer present.

To back up the database using the snapshot

◆ Copy the snapshot volumes to tape or other appropriate backup media.

Restoring from backup

For single instance Oracle environments, backup copies are used to restore volumes lost due to disk failure, or data destroyed due to human error. If a volume's data is corrupted and you know that you need to restore it from backup, you can use Database FlashSnap's reverse resynchronization function to restore the database.

Cloning a database (dbed_vmclonedb)

The SFDB commands enable you to create a clone database using snapshot volumes. You can use snapshots of a primary database to create a clone of the database at a given point in time. You can then implement decision-support analysis and report generation operations that take their data from the database clone rather than from the primary database to avoid introducing additional burdens on the production database.

A clone database can also serve as a valid backup of the primary database.

See "Backing up the database from snapshot volumes (dbed_vmclonedb)" on page 194.

You can also back up the primary database to tape using snapshot volumes.

The resynchronization functionality of Database FlashSnap allows you to quickly refresh the clone database 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.

Using Database FlashSnap to clone a database

In a same-node configuration, the `dbed_vmclonedb` command creates a clone database on the same host. The command can also be used to shut down the clone database and unmount its file systems. When creating or unmounting the clone database in a same-node configuration, `-r` *`relocate_path`* is required so that the clone database's file systems use different mount points than those used by the primary database.

When used in an off-host configuration, the `dbed_vmclonedb` command imports the snapshot disk groups, mounts the file systems on the snapshot, and starts a clone database. It can also reverse the process by shutting down the clone database, unmounting the file systems, and deporting the snapshot disk group.

Warning: When creating a clone database, all Storage Checkpoints in the original database are discarded.

Prerequisites	■ You must be logged in as the Oracle database administrator. ■ Before you can use the dbed_vmclonedb command, you must validate a snapplan and create a snapshot. See "About creating database snapshots" on page 165. See "Validating a snapplan (dbed_vmchecksnap)" on page 184. See "Creating a snapshot (dbed_vmsnap)" on page 191. ■ The volume snapshot must contain the entire database. ■ The system administrator must provide the database administrator with access to the necessary volumes and mount points. ■ Before you can use the `dbed_vmclonedb` command with the `-r` *`relocate_path`* option (which specifies the initial mount point for the snapshot image), the system administrator must create the mount point and then change the owner to the Oracle database administrator. ■ If SNAPSHOT_MODE is set to offline or instant, an off-host configuration is required and `-r` *`relocate_path`* is not allowed. ■ The Oracle database must have at least one mandatory archive destination. ■ In case of mapped mounts, the mapped mount points need to be created by the System Administrator.
Usage Notes	■ The `dbed_vmclonedb` command can be used on the secondary host. ■ In a same-node configuration, -r relocate_path is required if no mapped mounts were specified or some of the mapped mount entries were left blank in the snapplan. ■ The initialization parameters for the clone database are copied from the primary database. This means that the clone database takes up the same memory and machine resources as the primary database. If you want to reduce the memory requirements for the clone database, shut down the clone database and then start it up again using a different init.ora file that has reduced memory requirements. If the host where `dbed_vmclonedb` is run has little available memory, you may not be able to start up the clone database and the cloning operation may fail. ■ See the `dbed_vmclonedb`(1M) manual page for more information.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To mount a database and recover it manually

1. Start and mount the clone database to allow manual database recovery:

   ```
   $ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID \
     -o mountdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN \
     [-H ORACLE_HOME] [-r relocate_path]
   ```

2. Follow the Oracle recovery procedure to recover the database manually.
3. Update the snapshot status information for the clone database in the SFDB repository:

   ```
   $ /opt/VRTS/bin/dbed_vmclonedb -o update_status,\
     new_sid=new_sid,server_name=svr_name -f SNAPPLAN [-r relocate_path]
   ```

Example: Mounting the file systems without bringing up the clone database

In this example, file systems are mounted without bringing up the clone database. The clone database must be manually created and recovered before it can be used. This example is for a clone created on the same host as the primary database.

Note: You must issue commands as an Oracle database administrator in the following procedure.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
  -o mountdb,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 15:34:41
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
Mounting /clone/prod_ar on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_ar.
All redo-log files found.
Altering instance_name paramter in initabc.ora.
Altering instance_number paramter in initabc.ora.
Altering thread paramter in initabc.ora.Starting automatic database recover
Database NEWPROD (SID=NEWPROD) is in recovery mode.
If the database NEWPROD is recovered manually, you must run
dbed_vmclonedb -o update_status to change the snapshot status.
dbed_vmclonedb ended at 2006-03-02 15:34:59
```

The database status (database_recovered) needs to be updated for a clone database on the primary host after manual recovery has been completed.

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,\
  new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone

dbed_vmclonedb started at 2006-03-02 15:35:16
The snapshot status has been updated.
dbed_vmclonedb ended at 2006-03-02 15:35:42
```

Example: Mounting the file systems without recovering the clone database

Note: You must issue commands as an Oracle database administrator in the following procedure.

In this example, file systems are mounted without recovering the clone database. The clone database must be manually recovered before it can be used. This example is for a clone created on a secondary host.

```
dbed_vmclonedb started at 2006-03-09 23:26:50
Mounting /clone/arch on /dev/vx/dsk/SNAP_PRODdg/SNAP_arch.
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
All redo-log files found.
Altering instance_name paramter in initabc.ora.
Altering instance_number paramter in initabc.ora.
Altering thread paramter in initabc.ora.
Starting automatic database recovery.
Database NEWPROD (SID=NEWPROD) is in recovery mode.

If the database NEWPROD is recovered manually, you must run
dbed_vmclonedb -o update_status to change the snapshot status.
dbed_vmclonedb ended at 2006-03-09 23:27:17
```

The database is recovered manually.

The snapshot status (database_recovered) is updated for a clone database on a secondary host after manual recovery has been completed.

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,\
  new_sid=NEWPROD,server_name=host2 -f snap2

dbed_vmclonedb started at 2006-03-09 23:34:01
The snapshot status has been updated.
dbed_vmclonedb ended at 2006-03-09 23:34:35
```

To clone the database automatically

- ◆ Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID \
  -o recoverdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN \
  [-H ORACLE_HOME] [-r relocate_path]
```

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID \
  -o recoverdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN \
  [-H ORACLE_HOME] [-r relocate_path]
```

Where:

ORACLE_SID	Represents the name of the Oracle database used to create the snapshot.
snap_dg	Represents the name of the diskgroup that contains all the snapshot volumes.
new_sid	Specifies the ORACLE_SID for the clone database.
server_name	Specifies the server name as svr_name.
SNAPPLAN	Represents the name of the snapplan file.
ORACLE_HOME	Represents the ORACLE_HOME setting for the ORACLE_SID database.
relocate_path	Represents the name of the initial mount point for the snapshot image.

When cloning a database on a secondary host, ensure that PRIMARY_HOST and SECONDARY_HOST parameters in the snapplan file are different.

When the `-o recoverdb` option is used with `dbed_vmclonedb`, the clone database is recovered automatically using all available archive logs. If the `-o recoverdb` option is not used, you can perform point-in-time recovery manually.

In the following example, a clone of the primary database is automatically created on the same host as the primary database.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
  -o recoverdb,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 14:42:10
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
```

```
Mounting /clone/prod_ar on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_ar.
All redo-log files found.
Altering instance_name paramter in initabc.ora.
Altering instance_number paramter in initabc.ora.
Altering thread paramter in initabc.ora.
Starting automatic database recovery.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2006-03-02 14:43:05
```

In the following example, a clone of the primary database is automatically created on a secondary host.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
  -o recoverdb,new_sid=NEWPROD,server_name=svr_name -f snap2
```

```
dbed_vmclonedb started at 2006-03-09 23:03:40
Mounting /clone/arch on /dev/vx/dsk/SNAP_PRODdg/SNAP_arch.
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
All redo-log files found.
Altering instance_name paramter in initabc.ora.
Altering instance_number paramter in initabc.ora.
Altering thread paramter in initabc.ora.
Starting automatic database recovery.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2006-03-09 23:04:50
```

In the following example, a clone of the primary database is automatically created using mapped mounts. The Promary database mount points are located at /prod_db abd /prod_ar these were specified in the snapplan to be mapped to /tmp/datadst and /tmp/archdst.

```
$/opt/VRTS/bin/dbed_vmclonedb -S PROD -o recoverdb,new_sid=NEWPROD,server_n
dbed_vmclonedb started at 2009-08-01 16:12:00
PROD_dg:archvol (/prod_db) will be mapped to (/tmp/archdst)
PROD_dg:datavol (/prod_ar) will be mapped to (/tmp/datadst)
Editing remote_login_passwordfile in initcl2.ora.
All redo-log files found.
Altering instance_name parameter in initcl2.ora.
Altering instance_number parameter in initcl2.ora.
Altering thread parameter in initcl2.ora.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2009-08-01 16:15:05
```

Shutting down the clone database and unmounting file systems

When you are done using the clone database, you can shut it down and unmount all snapshot file systems with the `dbed_vmclonedb -o umount` command. If the clone database is used on a secondary host that has shared disks with the primary host, the `-o umount` option also deports the snapshot disk group.

Note: Any mounted Storage Checkpoints mounted need to be unmounted before running `dbed_vmclonedb -o umount`.

To shut down the clone database and unmount all snapshot file systems

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD,server_name=svr_name \
-f snap1 -r /clone
dbed_vmclonedb started at 2006-03-02 15:11:22
umounting /clone/prod_db
umounting /clone/arch
dbed_vmclonedb ended at 2006-03-02 15:11:47
```

In this example output, the clone database is shut down, file systems are unmounted, and the snapshot disk group is deported for a clone on a secondary host (two node configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD,server_name=svr_name \
-f snap2
dbed_vmclonedb started at 2006-03-09 23:09:21
Umounting /prod_db
Umounting /arch
dbed_vmclonedb ended at 2006-03-09 23:09:50
```

In the following example output, the clone database is shutdown and the file systems are unmounted for a clone created using mapped mounts.

```
dbed_vmclonedb -o umount,new_sid=NEWPROD,server_name=svr_name -f snap1
dbed_vmclonedb started at 2009-08-01 16:25:52
PROD_dg:archvol (/prod_db) will be mapped to (/tmp/archdst)
PROD_dg:datavol (/prod_ar) will be mapped to (/tmp/datadst)
Umounting /tmp/dadst.
Umounting /tmp/ardst.
dbed_vmclonedb ended at 2009-08-01 16:25:57
```

Restarting a Clone Database

If the clone database is down as a result of using `dbed_vmclonedb -o umount` or rebooting the system, you can restart it with the `-o restartdb` option.

Note: This option can only be used when a clone database is created successfully. If the clone database is recovered manually, `-o update_status` must be run to update the status before `-o restartdb` will work.

To start the clone database

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID \
  -o restartdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN [-H ORACLE_HOME] \
  [-r relocate_path]
```

In this example, the clone database is re-started on the same host as the primary database (same-node configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
  -o restartdb,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 15:14:49
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
Mounting /clone/prod_ar on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_ar.
Oracle instance NEWPROD successfully started.
dbed_vmclonedb ended at 2006-03-02 15:15:19
```

In this example, the clone database is re-started on the secondary host (two node configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
  -o restartdb,new_sid=NEWPROD,server_name=svr_name -f snap2
```

```
dbed_vmclonedb started at 2006-03-09 23:03:40
Mounting /clone/arch on /dev/vx/dsk/SNAP_PRODdg/SNAP_arch.
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODdg/SNAP_prod_db.
Oracle instance NEWPROD successfully started.
dbed_vmclonedb ended at 2006-03-09 23:04:50
```

Recreating Oracle tempfiles

After a clone database is created and opened, the tempfiles are added if they were residing on the snapshot volumes. If the tempfiles were not residing on the same file systems as the datafiles, `dbed_vmsnap` does not include the underlying volumes in the snapshot. In this situation, `dbed_vmclonedb` issues a warning message and you can then recreate any needed tempfiles on the clone database as described in the following procedure.

To recreate the Oracle tempfiles

1 If the tempfiles were not residing on the same file systems as the datafiles, the `dbed_vmclonedb` command displays WARNING and INFO messages similar to the following:

```
WARNING: Not all tempfiles were included in snapshot for
$ORACLE_SID, there is no snapshot volume for
/clone_path/temp02.dbf.
WARNING: Could not recreate tempfiles for $ORACLE_SID due to
lack of free space.INFO: The sql script for adding tempfiles to $ORACLE_
```

where $ORACLE_SID is the name of the clone database.

2 A script named `add_tf.`*`$ORACLE_SID`*`.sql` is provided in the /tmp directory for the purpose of recreating Oracle tempfiles. This script contains the SQL*Plus commands to recreate the missing tempfiles.

3 Make a copy of the /tmp/add_tf.*$ORACLE_SID*.sql script and open it to view the list of missing tempfiles.

An example of the add_tf.*$ORACLE_SID*.sql script is shown below:

```
$ cat /tmp/add_tf.$ORACLE_SID.sql
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE
'/clone_path/temp01.dbf'
SIZE 4194304 REUSE AUTOEXTEND ON NEXT 1048576 MAXSIZE 33554432 ;
ALTER TABLESPACE TEMP ADD TEMPFILE
'/clone_path/temp02.dbf' REUSE;
ALTER DATABASE TEMPFILE '/clone_path2/temp02.dbf'
OFFLINE;
```

4 Evaluate whether you need to recreate any temp files. If you want to recreate tempfiles, proceed to the next step.

5. In the add_tf.*$ORACLE_SID*.sql file, edit the sizes and default path names of the tempfiles as needed to reside on cloned volumes configured for database storage.

   Warning: Do not run the script without first editing it because path names may not exist and the specified mount points may not contain sufficient space.

   ---

6. After you have modified the `add_tf.$ORACLE_SID.sql` script, execute it against your clone database.
7. After you have successfully run the script, you may delete it.

Resynchronizing the snapshot to your database

When you have finished using a clone database or want to refresh it, you can resynchronize it with the original database. This is also known as refreshing the snapshot volume or merging the split snapshot image back to the current database image. After resynchronizing, the snapshot can be retaken for backup or decision-support purposes.

When resynchronizing the data in a volume:

- Resynchronize the snapshot from the original volume. This procedure is explained in this section.
- Resynchronizing the original volume from the snapshot. This choice is known as reverse resynchronization. Reverse resynchronization may be necessary to restore a corrupted database and is usually much quicker than using alternative approaches such as full restoration from backup media.

  Note: The reverse resynchronization option is not available for Oracle RAC environments.

You can resynchronize the snapshot from the original volume.

Prerequisites	■ You must be logged in as the Oracle database administrator. ■ Before you can resynchronize the snapshot image, you must validate a snapplan and create a snapshot. See "About creating database snapshots" on page 165. See "Validating a snapplan (dbed_vmchecksnap)" on page 184. See "Creating a snapshot (dbed_vmsnap)" on page 191. ■ If a clone database has been created, shut it down and unmount the file systems using the `dbed_vmclonedb -o umount` command. This command also deports the disk group if the primary and secondary hosts are different. See "Shutting down the clone database and unmounting file systems" on page 205. ■ The Oracle database must have at least one mandatory archive destination.
Usage Notes	■ The `dbed_vmsnap` command can only be executed on the primary host. ■ In an off-host configuration, the `dbed_vmsnap` command imports the disk group that was deported from the secondary host and joins the disk group back to the original disk group. The snapshot volumes again become plexes of the original volumes. The snapshot is then resynchronized. ■ See the `dbed_vmsnap`(1M) manual page for more information.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To resynchronize the snapshot image

◆ Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN -o resync
```

In this example, the snapshot image is resynchronized with the primary database.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o resync
dbed_vmsnap started at 2006-03-02 16:19:05
The option resync has been completed.
dbed_vmsnap ended at 2006-03-02 16:19:26
```

Now, you can again start creating snapshots.

Removing a snapshot volume

If a snapshot volume is no longer needed, you can remove it and free up the disk space for other uses by using the `vxedit rm` command.

Prerequisites	■ You must be logged in as superuser. ■ If the volume is on a mounted file system, you must unmount it before removing the volume.

To remove a snapplan and snapshot volume

1 To remove the snapshot and free up the storage used by it:

- If the snapshot has been taken, remove the snapshot as follows:

```
# vxsnap -g diskgroup dis snapshot_volume
```

```
# vxvol -g diskgroup stop snapshot_volume
```

```
# vxedit -g diskgroup -rf rm snapshot_volume
```

- If the snapshot has not been taken and the snapshot plex (mirror) exists, remove the snapshot as follows:

```
# vxsnap -g diskgroup rmmir volume
```

2 Remove the DCO and DCO volume:

```
# vxsnap -g diskgroup unprepare volume
```

3 Remove the snapplan.

```
# /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snapplan -o remove
```

For example, the following commands will remove a snapshot volume from disk group PRODdg:

```
# vxsnap -g PRODdg dis snap_v1
# vxvol -g PRODdg stop snap_v1
# vxedit -g PRODdg -rf rm snap_v1
```

Chapter 9

Using Database Dynamic Storage Tiering

This chapter includes the following topics:

About Database Dynamic Storage Tiering

Database Dynamic Storage Tiering (DST) matches data storage with data usage requirements. After data matching, the data can then be relocated based upon data usage and other requirements determined by the database administrator (DBA).

As more and more data is retained over a period of time, eventually, some of that data is needed less frequently. The data that is needed less frequently still requires a large amount of disk space. DST enables the database administrator to manage data so that less frequently used data can be moved to slower, less expensive disks. This also permits the frequently accessed data to be stored on faster disks for quicker retrieval.

Tiered storage is the assignment of different types of data to different storage types to improve performance and reduce costs. With DST, storage classes are used to designate which disks make up a particular tier. There are two common ways of defining storage classes:

- Performance, or storage, cost class: The most-used class consists of fast, expensive disks. When data is no longer needed on a regular basis, the data can be moved to a different class that is made up of slower, less expensive disks.
- Resilience class: Each class consists of non-mirrored volumes, mirrored volumes, and n-way mirrored volumes.
  For example, a database is usually made up of data, an index, and logs. The data could be set up with a three-way mirror because data is critical. The index could be set up with a two-way mirror because the index is important, but can be recreated. The logs are not required on a daily basis and could be set up without mirroring.

Dynamic Storage Tiering policies control initial file location and the circumstances under which existing files are relocated. These policies cause the files to which they apply to be created and extended on specific subsets of a file systems's volume set, known as placement classes. The files are relocated to volumes in other placement classes when they meet specified naming, timing, access rate, and storage capacity-related conditions.

In addition to preset policies, you can manually move files to faster or slower storage with DST, when necessary. You can also run reports that list active policies, display file activity, display volume usage, or show file statistics.

Database Dynamic Storage Tiering building blocks

To use Database Dynamic Storage Tiering, your storage must be managed using the following features:

- VxFS multi-volume file system
- VxVM volume set
- Volume tags
- Dynamic Storage Tiering policies

About VxFS multi-volume file systems

Multi-volume file systems are file systems that occupy two or more virtual volumes. The collection of volumes is known as a volume set, and is made up of disks or disk array LUNs belonging to a single Veritas Volume Manager (VxVM) disk group. A multi-volume file system presents a single name space, making the existence of multiple volumes transparent to users and applications. Each volume retains a separate identity for administrative purposes, making it possible to control the locations to which individual files are directed. This feature is available only on file systems meeting the following requirements:

- The minimum Diskgroup version is 140.
- The minimum filesystem version is 7.

To convert your existing VxFS file system to a VxFS multi-volume file system, you must convert a single volume to a volume set. See "Converting a VxFS file system to a VxFS multi-volume file system" on page 219.

The VxFS volume administration utility (fsvoladm utility) can be used to administer VxFS volumes. The fsvoladm utility performs administrative tasks, such as adding, removing, resizing, encapsulating volumes, and setting, clearing, or querying flags on volumes in a specified Veritas File System.

See the fsvoladm (1M) manual page for additional information about using this utility.

About VxVM volume sets

Volume sets allow several volumes to be represented by a single logical object. Volume sets cannot be empty. All I/O from and to the underlying volumes is directed via the I/O interfaces of the volume set. The volume set feature supports the multi-volume enhancement to Veritas File System (VxFS). This feature allows file systems to make best use of the different performance and availability characteristics of the underlying volumes. For example, file system metadata could be stored on volumes with higher redundancy, and user data on volumes with better performance.

About volume tags

You make a VxVM volume part of a placement class by associating a volume tag with it. For file placement purposes, VxFS treats all of the volumes in a placement class as equivalent, and balances space allocation across them. A volume may have more than one tag associated with it. If a volume has multiple tags, the volume belongs to multiple placement classes and is subject to allocation and relocation policies that relate to any of the placement classes.

Warning: Multiple tagging should be used carefully.

A placement class is a Dynamic Storage Tiering attribute of a given volume in a volume set of a multi-volume file system. This attribute is a character string, and is known as a volume tag.

About Dynamic Storage Tiering policies

Dynamic Storage Tiering allows administrators of multi-volume VxFS file systems to manage the placement of files on individual volumes in a volume set by defining placement policies that control both initial file location and the circumstances under which existing files are relocated. These placement policies cause the files to which they apply to be created and extended on specific subsets of a file system's volume set, known as placement classes. The files are relocated to volumes in other placement classes when they meet the specified naming, timing, access rate, and storage capacity-related conditions.

Database Dynamic Storage Tiering in a High Availability (HA) environment

Veritas Cluster Server does not provide a bundled agent for volume sets. If issues arise with volumes or volume sets, the issues can only be detected at the DiskGroup and Mount resource levels.

The DiskGroup agent brings online, takes offline, and monitors a Veritas Volume Manager (VxVM) disk group. This agent uses VxVM commands. When the value of the StartVolumes and StopVolumes attributes are both 1, the DiskGroup agent onlines and offlines the volumes during the import and deport operations of the disk group. When using volume sets, set StartVolumes and StopVolumes attributes of the DiskGroup resource that contains the volume set to 1. If a file system is created on the volume set, use a Mount resource to mount the volume set.

The Mount agent brings online, takes offline, and monitors a file system or NFS client mount point.

If you are using any of the Database Dynamic Storage Tiering commands in a high availability (HA) environment, the time on each system in the cluster must be synchronized. Otherwise, the scheduled task may not be executed at the expected time after a service group failover.

For additional information, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

Configuring Database Dynamic Storage Tiering

To use database Dynamic Storage Tiering, the following requirements must be met:

- An Oracle database must be up and running.
- Only the Oracle database administrator can run Database Dynamic Storage Tiering commands.

To use Database Dynamic Storage Tiering, the following tasks must be performed:

- Review the Database Dynamic Storage Tiering command requirements.
- Define the database parameters.
- Set up storage classes.
- Convert an existing VxFS database file system to a VxFS multi-volume file system for use with Database Dynamic Storage Tiering.
- Classify, or tag, volumes so that the tags indicate the quality of the underlying disk.
- Display the free space on each class.
- Add or remove volumes as necessary.

Database Dynamic Storage Tiering command requirements

Before defining your database parameters, review the following command requirements:

- Run the `dbed_update` command before running any of the Database Dynamic Storage Tiering commands. You should also run the `dbed_update` command if any of the database files change.
  The repository must be up to date, since the Database Dynamic Storage Tiering commands retrieve database information from the repository.
- You do not need to set the environment variable LD_LIBRARY_PATH to use the SFDB commands. However, if you set this environment variable for another reason, Symantec recommends including the library path /opt/VRTSdbed/common/lib before other library paths so the SFDB commands do not mistakenly link with libraries with same name in the other library paths.
- If you are using any of the Database Dynamic Storage Tiering commands in a high availability (HA) environment, the time on each system in the cluster must be synchronized.
- Create the volumes that you want to add to the multi-volume file system in the same disk group as the file system volume. As root, use the following command to change the owner of each volume:

  ```
  # /opt/VRTS/bin/vxedit -g disk_group \
  set user=oracle volume
  ```
- Change the owner of the mount point on which you want to implement Database Dynamic Storage Tiering to oracle.

Defining database parameters

Running the `dbdst_admin` command defines parameters for the entire database. You must run this command at least once to define the database parameters for Database Dynamic Storage Tiering. Three pre-defined storage classes will be created (PRIMARY, SECONDARY, and BALANCE). Parameter values are stored in the SFDB repository.

Set at least one of the parameters in maxclass, minclass, statinterval, sweeptime, sweepinterval, purgetime, or purgeinterval, to enable default values. Add at least one class to enable the default classes.

Table 9-1 lists the options for the `dbdst_admin` command:

Table 9-1 dbdst_admin command options

Command option	Description
`-S $ORACLE_SID`	Specifies the `ORACLE_SID`, which is the name of the Oracle instance.
list	Lists all of the Database Dynamic Storage Tiering parameters of the database, including class name and description. This option should be used exclusively from the other options.
maxclass=	Maximum number of storage classes allowed in the database. The default value is 4.
minclass=	Minimum number of storage classes allowed in the database. The default value is 2.
sweepinterval=	Interval for file sweeping for file relocation. Default value is 1, which means one per day. If this value is set to 0, all scheduled sweep tasks will become unscheduled.
sweeptime=	Time per day for the file sweep to take place. Times are entered in 24-hour periods and should list hour: minute. For example, 8:30 AM is represented as 08:30 and 10:00 PM is represented as 22:00. Default value is 22:00.
statinterval=	Interval in minutes for gathering file statistics. Default value is 30, which represents every 30 minutes. If this value is set to 0, all scheduled tasks will become unscheduled.

Table 9-1 dbdst_admin command options *(continued)*

Command option	Description
purgeinterval=	Number of days after which the file statistics in the repository will be summarized and purged. Default value is 30. It is recommended that you set your purge interval sooner because you will not be able to view any statistics until the first 30-day interval is over, if you use the default.
purgetime=	Time per day for the file purge to take place. Times are entered in 24-hour periods and should list hour: minute. For example, 8:30 AM is represented as 08:30 and 8:00 PM is represented as 20:00. Default value is 20:00.
addclass=	Parameter that allows you to add a class to a database. The information should be entered as class:"description", where the class represents the class name and description is a string of up to 64 characters enclosed by double quotes used to describe the class.
rmclass=	Parameter that allows you to remove a class from a database. Enter the class name as it appears in the database.
-o definechunk= <classname>: {128k|256k|512k|1m}	Defines a chunksize in bytes for the given storage class. Valid chunksizes are 128k, 256k, 512k or 1m bytes. When a chunksize is specified for a storage class, the files in this storage class will be extent-balanced. Each chunk of the file will be in a separate volume of the storage class. A given file will have approximately equal number of chunks on each component volumes of the stortage class. When a new volume is added or an existing volume is removed from the storage class (using dbdst_addvol or dbdst_rmvol), the files are automatically balanced again.

Note: If you do not want to change specific default values, you can omit those parameters when you run the dbdst_admin command. You only need to enter the parameters that need to be changed.

To define database parameters

- Use the `dbdst_admin` command as follows:

```
dbdst_admin -S ORACLE_SID -o setup-parameters\
[,storage_class operations]
        setup-parameters
                maxclass=number,minclass=number,\
                        statinterval=minutes
                sweeptime=HH:MM,sweepinterval=days
                purgetime=HH:MM,purgeinterval=days
        storage_class operations
                addclass=classname:"description"
                rmclass=classname
                definechunk=classname:128k | 256k | 512k | 1m
```

For example, to add a class called tier1 for database PROD, and to set up a purge interval of one, meaning that the file statistics will be gathered for one day and then summarized and purged, use the dbdst_admin command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=tier1:"Fast Storage",\
purgeinterval=1
```

Setting up storage classes

When you define your database parameters, three pre-defined storage classes are created. You will need to add or remove storage classes to meet your needs.

Adding storage classes

In addition to the default storage classes, you can add storage classes to better manage your data.

Before adding a storage class, review the following information:

- Use the `dbdst_admin` command as follows:

  ```
  $ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=classname:\
  "description"
  ```

For example, to create a storage class named "FAST" for an EMC array, use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=FAST:\
"fast EMC array"
```

Removing storage classes

If you no longer require a specific storage class, you can remove it.

Note: You cannot remove the pre-defined storage classes (PRIMARY, SECONDARY, and BALANCE).

Before removing a storage class, review the following information:

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID rmclass=classname
```

For example, to remove a storage class called "SLOW," use the dbdst_admin command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID rmclass=SLOW
```

Displaying storage classes

You can display a list of Database Dynamic Storage Tiering properties and storage classes using the dbdst_admin command.

Before displaying your storage classes, review the following information:

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o list
```

Converting a VxFS file system to a VxFS multi-volume file system

To convert your existing VxFS file system to a VxFS multi-volume file system, you must convert a single volume to a volume set.

Converting a single volume to a volume set

When you convert to a volume set using the `dbdst_convert` command, the original volume will be renamed to a new volume name. The mount device name will become the new volume set name. Creating the new volume set name with the mount device name nullifies the need to rename the mount device in various locations.

Before converting to a volume set, make sure the following conditions have been met:

Prerequisites	■ The Oracle database must not be active. ■ Create at least one additional volume.

Usage Notes	■ You must convert the single-volume file system on which you plan to implement Database Dynamic Storage Tiering. ■ The file system has to be unmounted when you run the `dbdst_convert` command. ■ If the file system has n volumes, volumes 1 through n–1 will be placed in the storage class "PRIMARY" and volume n will be placed in the storage class "SECONDARY." ■ The volumes specified when running the conversion must be in the same disk group as the mount device.

To convert a mount device from a single volume device to a volume set

1 Use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S $ORACLE_SID -M mount_device -v \
volume_name,volume_name
```

2 Bring the database objects online.

For example, to convert a volume-based oradata file system to a Database Dynamic Storage Tiering-ready volume set file system on mount device /dev/vx/dsk/oradg/oradata, use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD -M /dev/vx/dsk/oradg/oradata -v \
new_vol1,new_vol2
```

After conversion, you will have a volume set named oradata containing three volumes (oradata_b4vset, new_vol1, and new_vol2). The file system will have two storage classes defined as PRIMARY and SECONDARY. The volumes will be assigned as follows:

- PRIMARY storage class will contain volumes oradata_b4vset and new_vol1.
- SECONDARY storage class will contain volume new_vol2.

Classifying volumes into a storage class

Before creating a DST policy or manually moving data, assign classes to your volumes.

Before assigning classes to volumes, review the following information:

Usage notes	■ You must convert your VxFS file system to a multi-volume file system first. ■ Storage classes must be registered using the `dbdst_admin` command before assigning classes to volumes. ■ The database can be online or offline.

To classify a volume

- Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M mount_device \
-v volume_name:class[,volume_name:class]
```

For example, to assign the class "FAST" to volume new_vol1, use the `dbdst_classify` command as follows

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M /dev/vx/dsk/oradg/oradata
-v new_vol1:FAST
```

Displaying free space on your storage class

To see the free space, class information, and volume information on your storage classes, use the `dbdst_show_fs` command.

Table 9-2 shows the `dbdst_show_fs` command options.

Table 9-2 dbdst_show_fs command options

Command options	Description
`-S $ORACLE_SID`	Specifies the `ORACLE_SID`, which is the name of the Oracle instance.
`-o volume`	Displays the free space on volumes in each class.
`-m`	Specifies the mount point.

Before displaying the free space on a storage class, review the following information:

Prerequisites	■ Make sure the file system is mounted. ■ See the `dbdst_show_sf` (1M) manual page.

To display the free space on a storage class

- Use the `dbdst_show_fs` command as follows:

```
dbdst_show_fs -S $ORACLE_SID -m mount_point
dbdst_show_fs -S $ORACLE_SID -m mount_point -o volume
```

Adding new volumes to a storage class

Use the `dbdst_addvol` command to add volumes to a volume set.

Before adding a volume, review the following information:

Usage notes	■ The database must be inactive when adding volumes to a storage class. ■ The database file system has to be mounted.

To add a volume to a volume set

- Use the `dbdst_addvol` command as follows:

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID -M mount_device \
-v volume_name:class[,volume_name:class]
```

Removing volumes from a storage class

You may need to remove a volume from a volume set. To remove a volume, use the `dbdst_rmvol` command.

Before removing a volume, review the following information:

Usage notes	■ The database must be inactive when removing volumes from a storage class. ■ Only a volume that does not contain any file system data can be removed.

To remove a volume from a volume set

Use the `dbdst_rmvol` command as follows:

```
$ /opt/VRTS/bin/dbdst_rmvol -S $ORACLE_SID -M mount_device \
-v volume_name[,volume_name]
```

Dynamic Storage Tiering policy management

You can choose to manually relocate files or tablespaces, or you can use a preset Dynamic Storage Tiering (DST) policy.

Note: You must issue commands as an Oracle database administrator in the following procedures.

Relocating files

Table 9-3 shows the `dbdst_file_move` command options.

Table 9-3 dbdst_file_move command options

Command options	Description
`-o archive[n] | flashback`	Specifies which archive logs or Flashback logs to move. Do not use this option with the `-f` option.
`-o external | datafile`	Specifies whether to move external files or datafiles. Use this option with the -f option.
`-f listfile`	Specifies a listfile that contains a list of files or directories to be moved.
`-c storage_class [:days]`	Specifies the storage class to which the files should be moved. If the days option is used, the files will be moved to the class specified if they have not been accessed in the number of days specified. Do not specify days if you are using the -o datafile option.
`-R`	Removes the policy for the specified object.

Before relocating a file, review the following information:

Usage notes	■ Multiple partitions cannot reside on the same tablespace.

To relocate a file

- Use the `dbdst_file_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_file_move -S $ORACLE_SID -o datafile \
-f listfile -c storage_class:days [-c storage_class:days]
```

Relocating tablespaces

Use the `dbdst_tbs_move` command to move tablespaces to the desired storage class. The command queries the SFDB repository for the tablespace file names, then performs a one-time move based on your immediate requirements.

To relocate a tablespace

◆ Use the `dbdst_tbs_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_tbs_move -S $ORACLE_SID -t tablespace \
-c storage_class
```

where

- *tablespace* indicates which tablespace to move.
- *storage_class* indicates to which class the tablespace should be moved.

Relocating table partitions

Use the `dbdst_partition_move` to move table partitions. The command queries the database to validate the names of the table and partition. From this information, a list of datafiles is derived and a one-time move of the files to the desired class is executed.

Before relocating table partitions, review the following information:

Prerequisites	The database must be up when you run the `dbdst_partition_move` command.

To relocate a table partition

◆ Use the `dbdst_partition_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S $ORACLE_SID -T table_name \
-p partition_name -c storage_class
```

where

- `-T` indicates the table name.
- `-p` indicates the partition name.
- `-c` indicates the class to which the table partition is to be moved.

For example, to move the SALES_Q1 partition of the SALES table to storage class SLOW, use the `dbdst_partition_move` as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S $ORACLE_SID -T SALES \
-p SALES_Q1 -c SLOW
```

Using preset policies

Use the `dbdst_preset_policy` command to set a policy based on file name patterns before the files are created.

Table 9-4 shows the preset policies command options.

Table 9-4 dbdst_present_policy command options

Command option	Description
`-d` directory	Indicates the directory on which the placement policy will be applied.
`-e`	Enforces the file system of the specified directory. Use this option if there was an error in the previous enforcement that has been corrected and needs to be enforced again.
`-R`	Removes all pattern-based placement policies related to this directory.
`-l`	Lists the existing file placement that is set to the specified directory.
`-P` pattern_spec	Specifies file patterns and class assignment. This option will automatically place files in the desired class as soon as they are created. Existing files and newly created files will be moved immediately to the class specified.
`-f` pattern file	Specifies a file that contains a particular class and pattern. New files with this pattern will be placed in the class immediately. Existing files will be moved as well.
`-E`	Specifies that existing files should be moved to the designated class in a one-time move to be scheduled at a later time, such as the sweeptime specified in the `dbdst_admin` command.

To create a preset policy

- Use the `dbdst_preset_policy` command as follows:

```
$ dbdst_preset_policy -S oracle_sid -d directory [ -e | -R | -l | -P patter
-f pattern_file ] [-E]
```

  where pattern_spec has the format of "class=pattern,pattern,...[:class=pattern,pattern,...]"

Extent balancing in a database environment

To obtain better performance in a database environment, you would normally use a volume striped over several disks. As the amount of data stored in the file system increases over time, additional space in the form of new disks must be added.

To increase space, you could perform a volume relayout using the `vxrelayout` command. However, changing a large volume from a four-way striped volume to six-way striped volume involves moving old block information into temporary space and writing those blocks from the temporary space to a new volume, which would require an extended amount of time. To solve this problem, Veritas Storage Foundation for Db provides the Extent Balanced File System or EBFS .

An Extent Balanced File System is created on a multi-volume file system where individual volumes are not striped over individual disks. For data-availability, these individual volumes can be mirrored. The file system on the EBFS has a special placement policy called a balance policy. When the balance policy is applied, all the files are divided into small "chunks" and the chunks are laid out on volumes so that adjacent chunks are on different volumes. The default chunk size is 1MB and can be modified. Since every file contains chunks on all available volumes, it is important that individual volumes that make up the EBFS and volume set be of same size and same access properties.

Setting up the file system in this way provides the same benefit as striping your volumes.

Note: You cannot convert an existing file system to an EBFS file system.

Extent balancing file system

You can define allocation policies with a balance allocation order and "chunk" size to files or a file system, known as extent balancing. The chunk size is the maximum size of any extent that files or a file system with this assigned policy can have. The chunk size can only be specified for allocation policies with a balance allocation order.

An extent balancing policy specifies the balance allocation order and a non-zero chunk size. The balance allocation order distributes allocations randomly across the volumes specified in the policy and limits each allocation to a maximum size equal to the specified chunk size.

Extent balancing extends the behavior of policy enforcement by rebalancing extent allocations such that each volume in the policy is as equally used as possible. Policy enforcement handles the following cases:

- New volumes are added to the policy, and the extents associated with a file need rebalancing across all volumes, including the new ones.
- Volumes are removed from the volume set or from the policy, and the extents for a file residing on a removed volume need to be moved to other volumes in the policy.
- An extent balancing policy is assigned to a file and its extents have to be reorganized to meet the chunk size requirements defined in the policy.

The extent balancing policy is intended for balancing data extents belonging to files across volumes defined in the policy. However, there is no restriction imposed in assigning extent balancing policy for metadata.

Note: If the fixed extent size is less than the chunk size, then the extent size will be limited to the largest multiple of the fixed extent size that is less than the chunk size. If the fixed extent size is greater than the chunk size, then the extent size will be the fixed extent size.

Creating an extent balanced file system

Any MultiVolume File System (MVFS) can become an extent balanced file system, if the storage tier has a chunk size associated with the class. The `dbdst_admin` command permits the user to define a chunk size for the class.

For example, the following `dbdst_admin` commands define chunk sizes for the gold and silver storage classes:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o definechunk gold:256K
```

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o definechunk silver:128K
```

The above commands make storage class gold as extent balanced.

Once the chunksize is defined for a storage tier, we can classify any MVFS into this storage tier.

For example, assume that /oradata is the filesystem created on volume-set /dev/vx/dsk/oradg/ora_vset, and contains database datafiles. Let us further assume that datafile names end with extension *.dbf. To define storage class in this MVFS, the following `dbdst_classify` command is used:

```
$ /opt/VRTS/bin/dbdst_classify  -S $ORACLE_SID -M /dev/vx/dsk/oradg/ora_vset -v \
vol1:GOLD,vol2:GOLD,vol3:GOLD
```

It is important to note that, an MVFS can have multiple storage tiers and that each tier may have a different chunk size. For example, for the same MVFS in the above example, we can define another storage tier using the `dbdst_classify` command:

```
$ /opt/VRTS/bin/dbdst_classify  -S $ORACLE_SID -M /dev/vx/dsk/oradg/ora_vset -v \
vol4:silver,vol5:silver
```

At this point we have two storage tiers in MVFS /oradata each having different chunksizes. To create the real extent balance, we need to assign a DST policy and to enforce it.

To define and enforce the policy, you could use the following `dbdst_preset_policy` command:

```
$/opt/VRTS/bin/dbdst_preset_policy -S $ORACLE_SID -d /oradata \
-P GOLD=*.dbf:SILVER=*.inx
```

The above example creates a DST policy, assigns the policy to /oradata and enforces the policy. All datafiles of the form *.dbf will be extent balanced in GOLD tier with chunksize 256K and all index files of the form *.inx will be extent balanced in SILVER tier with chunk size 128K.

To view the space usage in the /oradata MVFS use the `dbdst_show_fs` command. For example:

```
$ /opt/VRTS/bin/dbdst_show_fs -S $ORACLE_SID -m /oradata
```

When the GOLD or SILVER tier requires more space, we could add extra space by adding new volumes to the respective storage tier using the `dbdst_addvol` command. For example:

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID -M /dev/vx/dsk/oradg/ora_vset  -v vol7:GOLD
```

As soon as you add a new volume, the DST policy is enforced and the extents are balanced over the new volume too. This can be viewed by using the `dbdst_show_fs` command again.

To view detailed extent information about a given file, you can use the `fsmap` command. For example:

Running Database Dynamic Storage Tiering reports

You can create a report that lists all updated allocation policies or you can view an audit report, which lists recent relocation changes for a specific date range resulting from your policies.

Viewing modified allocation policies

To create a list of modified allocation policies, use the `dbdst_report` command with the `policy` option.

To list allocation policies

- Use the `dbdst_report` command as follows:

  ```
  $ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o policy
  ```

For example to view a list of modified allocation policies, use the `dbdst_report` command as follows:

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o policy
```

For example:

```
$ /opt/VRTS/bin/dbdst_report -S  VRTS11r1 -o policy
TSDB Database Policy Report
-----------------------------------------------------------

Plan Name = preset_dir:/data11r1/VRTS11r1
Plan Type = PRESET POLICY DIRECTORY
Rule Name =  DBED_precreate_ino4_PRIMARY
Directory=/dev/vx/dsk/dstdatadg09/datavol : VRTS11r1
Rule Description =  PRIMARY=MEDIUM=*.log,undo*.dbf

Plan Name = tbs_move:MYTBS
Plan Type = TBS MOVE
Rule Name =  DBED_relocate_once_ino4_MEDIUM
Directory=/dev/vx/dsk/dstdatadg09/datavol : VRTS11r1
Rule Description =  MEDIUM
```

Viewing audit reports

To view an audit report, which lists recent file relocation changes within a specific date range, use the `dbdst_report` command with the `audit` option.

To view an audit report

- Use the `dbdst_report` command as follows:

  ```
  $ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o audit \
  startdate=yyyy-mm-dd,enddate=yyyy-mm-dd
  ```

For example, to view an audit report of changes from January 1, 2007 through March 1, 2007, use the `dbdst_report` command as follows:

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o audit \
startdate=2007-01-01,enddate=2007-03-01
```

Oracle Database Dynamic Storage Tiering use cases

This section discusses Oracle use cases for Dynamic Storage Tiering.

Migrating partitioned data and tablespaces

Perhaps the simplest application of multi-tier storage to databases is relocation of individual table partitions between different placement classes as usage requirements change. If exact relocation times are unpredictable, or if relocation is infrequent, administrators may wish to relocate table partitions when necessary rather than defining strict periodic relocation schedules.

Ad hoc relocation of table partitions can be useful, for example, with databases that track sales and inventory for seasonal businesses such as sports equipment or outdoor furniture retailing. As the selling season for one type of inventory (for example, summer equipment or furniture) approaches, database table partitions that represent in-season goods can be relocated to high-performance storage, since they will be accessed frequently during the coming months. Similarly, partitions that represent out-of-season goods can be relocated to lower-cost storage, since activity against them is likely to be infrequent.

For example, sales are mostly catalog-driven for a large retailer specializing in sports equipment. Product details are saved in a large database and the product table is partitioned based on type of activity. Some of the products are seasonal and do not sell well at other times. For example, very few snow skis are sold during the summer. To achieve season-based migration, see the following example.

Assume the table product_tab has two partitions, summer and winter. Each of these partitions is mapped to a separate data file.

First, you must set up your system to use Database Dynamic Storage Tiering.

To add the fast_storage and slow_storage storage classes

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
fast_storage:"Fast Storage for Production DB"
```

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
slow_storage:"Slow Storage for Production DB"
```

To convert the database's file system and add volumes for use with Database Dynamic Storage Tiering

- Use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD \
-M /dev/vx/dsk/oradg/oradata -v new_vol1,new_vol2,new_vol3
```

To classify volumes into storage classes

- Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oradata -v new_vol1:fast_storage
```

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oradata -v new_vol2:slow_storage,\
new_vol3:slow_storage
```

Once the volumes are configured, an administrator can define file placement policy rules that specify seasonal relocation of selected tablespaces and partitions and assign them to the database's file system.

To move summer data to slower storage and winter data to faster storage at the beginning of winter

- Use the `dbdst_partition_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \
-p winter -c fast_storage
```

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \
-p summer -c slow_storage
```

These commands relocate the files that comprise the winter partition of the product_tab table to placement class fast_storage, and the files that comprise the summer partition to placement class slow_storage. Database Dynamic Storage Tiering determines which files comprise the winter and summer partitions of product_tab, and uses underlying DST services to immediately relocate those files to the fast_storage and slow_storage placement classes respectively.

To move winter data to slower storage and summer data to faster storage at the beginning of summer

◆ Use the `dbdst_partition_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \
-p summer -c fast_storage
```

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \
-p winter -c slow_storage
```

Database Dynamic Storage Tiering formulates DST policy rules that unconditionally relocate the files containing the target partitions to the destination placement classes. It merges these rules into the database file system's active policy, assigns the resulting composite policy to the file system, and enforces it immediately to relocate the subject files. Because the added policy rules precede any other rules in the active policy, the subject files remain in place until the dbdst_partition_move command is next executed, at which time the rules are removed and replaced with others.

Scheduling the relocation of archive and Flashback logs

Because they are the primary mechanism for recovering from data corruption, database logs are normally kept on premium storage, both for I/O performance and data reliability reasons. Even after they have been archived, logs are normally kept online for fast recovery, but the likelihood of referring to an archived log decreases significantly as its age increases. This suggests that archived database logs might be relocated to lower-cost volumes after a certain period of inactivity.

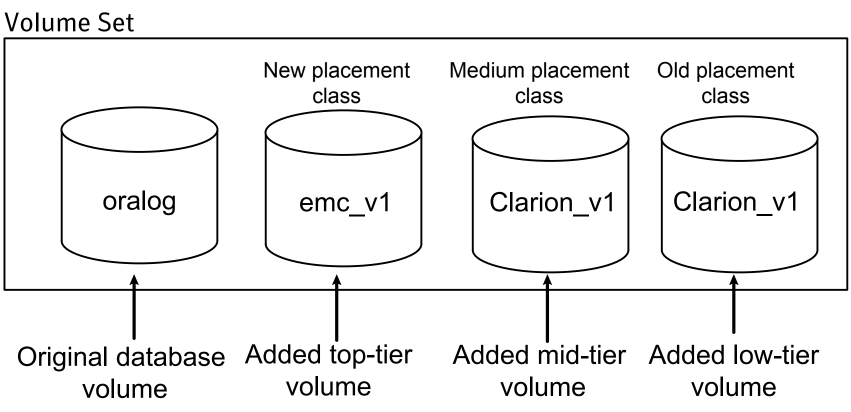
Similarly, Veritas Storage Foundation for DB Flashback technology creates logs that can be used for quick recovery from database corruption by restoring a database to its state at a previous time. Flashback logs are normally kept for a shorter period than archived database logs, if used at all, they are typically used within a few hours of creation. Two or three days are a typical Flashback log lifetime.

The rapidly decaying probability of use for archive and Flashback logs suggests that regular enforcement of a placement policy that relocates them to lower-cost storage after a period of inactivity can reduce an enterprise's average cost of online storage.

For example, a customer could be using a large OLTP Oracle database with thousands of active sessions, which needs to be up and running 24 hours a day and seven days a week with uptime of over 99%, and the database uses Flashback technology to correct any accidental errors quickly. The database generates large number of archive logs per day. If the database goes down for any reason, there is business requirement to bring the database back online and functional with in 15 minutes. To prevent Oracle log switch delays during transactions, the archive logs need to be created in a fast EMC array. Archive logs older than a week can be moved to a mid-range Clarion array. Archive logs older than 15 days can be moved to slow JBOD disks. Archive logs are purged after 30 days. Current Flashback logs are created manually by the database administrator on fast EMC storage and can be moved to Clarion storage after two days. The database administrator then deletes the Flashback logs after a week. To set up a system like this, see the following example. Assume that archive logs and Flashback logs are created on the same file system, /oralog. On the file system, /oralog/archive1 contains archive logs and /oralog/flashback contains Flashback logs.

Figure 9-1 illustrates a three-tier volume configuration that is suitable for automatic relocation and deletion of archive logs and Flashback logs.

Figure 9-1 Database storage configuration suitable for automatic relocation of archive and Flashback logs



The file system used by the production database in this example originally resides on the single volume oralog, which must be prepared by adding volumes and placement classes assigned to the volumes.

To add the NEW, MEDIUM, and OLD storage classes

◆ Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
NEW:"EMC Storage for Production DB"
```

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
MEDIUM:"Clarion Storage for Production DB"
```

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
OLD:"JBOD Storage for Production DB"
```

To convert the database's file system and add volumes for use with Database Dynamic Storage Tiering

◆ Use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD \
-M /dev/vx/dsk/oradg/oralog -v emc_v1,clarion_v1,jbod_v1
```

To classify volumes into storage classes

◆ Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oralog -v emc_v1:NEW
```

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oralog -v clarion_v1:MEDIUM
```

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oralog -v jbod_v1:OLD
```

Once the volumes are configured, an administrator can define file placement policy rules that specify access age-based relocation of selected files and assign them to the database’s file system.

To define rules that periodically relocate Flashback and archive logs

- Use the `dbdst_file_move` command as follows:

  ```
  $ /opt/VRTS/bin/dbdst_file_move -S PROD -o flashback -c MEDIUM:2
  ```

  This command relocates files in the Flashback directory that have not been accessed for two days to the MEDIUM volume.

  ```
  $ /opt/VRTS/bin/dbdst_file_move -S PROD -o archive1 -c MEDIUM:7 \
  -c OLD:15
  ```

  This command relocates files in the archive1 directory that have not been accessed for seven days to the MEDIUM volume, and files that have not been accessed for 15 days to the OLD volume.

Database Dynamic Storage Tiering translates these commands into DST access age-based policy rules, merges them with the file system's placement policy, and assigns the resulting policy to the file system. By default, Database Dynamic Storage Tiering enforces the active policy daily. During enforcement, the new rules relocate qualifying files to the destination storage tiers specified in the `dbdst_file_move` commands used to create the policies.

Chapter 10

Deploying Storage Foundation for Databases (SFDB) tools in a replicated environment

This chapter includes the following topics:

About deploying SF Databases tools with VVR

Storage Foundation for Databases (SFDB) extends the strengths of the the Storage Foundation Enterprise products with database-specific enhancements to offer performance, availability, and manageability for Oracle database servers. SFDB tools provide the means for:

- Enhanced IO methods such as Qick IO, Cached Quick IO, ODM, and Cached ODM, which enhance database performance
- Taking point-in-time copies filesystem checkpoints and Volume Snapshots
- Creating a clone database from volume snapshots

- Performing a rollback of the database from a given filesystem checkpoint
- Performing storage tiering using Database Dynamic Storage Tiering (DBDST)

Veritas Volume Replicator (VVR) is an option of Veritas Volume Manager (VxVM) that works as a fully integrated component of VxVM. VVR benefits from the robustness, ease of use, and high performance of VxVM, and at the same time, adds replication capability to VxVM.

VVR is able to:

- Replicate existing VxVM configurations
- Be transparently configured while the application is active
- Replicate application writes on the volumes at the source location to one or more remote locations across any distance
- Provide a consistent copy of application data at the remote locations

If a disaster occurs at the source location, you can use the copy of the application data at the remote location and restart the application at the remote location. This section describes how to use SFDB tools at a remote VVR site.

For more information about how to setup and configure VVR replication for Oracle database environment:

See the *Veritas Volume Replicator Administrator's Guide*

SF storage tools architecture

Storage Foundation for Databases (SFDB) tools are easy to use and can be run by database Administrators without root or system-admin privileges.

The SFDB tools consist of vxdbd daemon which starts at system startup, and a repository for storing metadata required for SFDB commands. The SFDB tools use the VCS framework to make the repository highly available. The SFDB tools store Oracle and Storage Foundation configuration information in a repository.

The SFDB repository information includes data about the Oracle Database such as:

- Datafile and tablespace names
- Volume snapshot information
- Storage configuration
- Scheduled tasks
- Storage statistics

VVR replication environment

The typical replication environment for Storage Foundation Enterprise products includes:

- The volume and volumeset (vset) configuration parameters such as size and the number of volumes should match between primary and remote DR site.
- The number of mirrors for a given volume, plex names, volume names, sub-disks, stripe-width may differ between primary and remote DR site. Hence volume snapshots taken on the primary won't be available on the secondary site.
- VVR replicates all the data in the volume. This means all the VxFS data and metadata, inclusive of DST policy and files system checkpoint data, are replicated.
- VVR does not replicate volume tags. Enforcing previously existing DST policies will work on the secondary site as it is, but modifying and defining new policies is not supported on the secondary site.
- VVR maintains write-order. Applications such as Oracle can be brought up at the remote site using proper recovery procedures.
- When active replication is going on, the volumes at remote site are not open for applications. This means the file system is not mounted at remote site.
- When we want to switch over the application from primary to remote, we need to stop the replication, mount the filesystem and start the application.

Using ODM, Cached ODM, Quick IO, and Cached IO for the Oracle database in a VVR environment

Fast IO methods such as ODM, Cached ODM, Quick IO and Cached Quick IO are used for better application performance.

When using SFDB tools in a global environment replicated by VVR:

- Using ODM and Quick IO for Oracle at the remote site does not require replicating the SFDB repository and hence there is no need to replicate the repository.
- It is not necessary to use the same IO method at the primary and the remote sites. However if you need the same application performance at the remote site you may want to use same IO method at remote site.
- There is no need to run SFDB commands qio_getdbfiles and qio_convertdbfiles because the Quick IO files are replicated.

Best practices for using ODM or Quick IO with Oracle:

- Make sure you use same version of Oracle at the primary and the remote site.
- Make sure you enable ODM by linking the proper ODM library at the remote site.
- Quick IO files will be replicated as Quick IO files at the remote site and hence no extra configuration is required. However, if you are using Cached Quick IO, and you would like to use Cached Quick IO at the remote site, then you need to change the mount properties using the `qioadmin` command and also need to copy the /etc/vx/qioadmin file from the primary to the remote.

Using Storage Foundation for Databases (SFDB) tools in a VVR environment

The following procedures describe the additional steps required to use SFDB tools in a VVR environment.

- Adding a virtual hostname
- Using existing Database Storage Checkpoints at the primary site after the DR switchover
- Creating new Database Storage Checkpoints at the remote site
- Using Database FlashSnap with VVR replication
- Using Database Dynamic Storage Tiering (DBDST)with VVR replication

For SFDB commands to work in general in a site replicated with VVR, the following steps must be performed.

To run SFDB commands in a VVR environment

1. Stop VVR replication.
2. Run `fsck` for all database volumes on the replicated site.
3. Mount the database volume file systems.
4. As Oracle user, run the `dbed_update` to either update or create the SFDB repository, depending on your use case.
5. Run the SFDB commands as needed for your tasks.
6. After finishing your SFDB tasks:
   - Shut down the database.

- Unmount the volumes.

7 Resume VVR replication.

Adding a virtual hostname

After a DR failover, you will need to add a virtual hostname to the SFDB repository.

- Copy the primary node directory: /etc/vx/vxdba/*primary_database* to to the same location on the secondary node.
- Create a symlink from the /etc/vx/vxdba/*primary_database* directory on the primary to the SFDB repository.

The SFDB tools command `dbed_update` has two command line options:

```
-o list
 -o rename,old_node=<old_name>,new_node=<new_name>
```

Th ecommand `dbed_update -o list`displays all the hostnames registered in the repository.

For example:

```
$ dbed_update -S $ORACLE_SID -H $ORACLE_HOME -o list
dblxxeon04 1
```

The command `dbed_update -o rename,old_node=<old_name>,new_node=<new_name>` is used to rename the hostname from the old_name to the new_name.

For example:

```
$ dbed_update -S ORACLE_SID -H ORACLE_HOME \
-o rename,old_node=dblxxeon04,new_node=dblxrep
```

Note: The hostname <old_name> should be exactly same as that listed by `dbed_update -o list`.

Using existing Database Storage Checkpoints at the primary site after the DR switchover

Database Storage Checkpoint data is stored as File System (FS) metadata and hence replicated to the remote site. Once you mount the File System at the remote site, File System checkpoints are available and can be seen at the remote site.

When using checkpoints created at primary site after DR switchover, the host name is stored in the repository and hence we need a procedure to rename the host name in the repository.

To perform a rollback on the remote site from a primary site checkpoint

1. By default, the repository is already included in the database volume that is part of the VVR environment. If you use an alternate path for your SFDB repository, you should also replicate the SFDB repository volume for VVR replication.
2. Stop VVR replication.
3. Run `fsck` for all database volumes on the replicated site.
4. Mount the database volume file systems.
5. The SFDB repository has host name in the SFDB repository. Change the host name using the SFDB tool.

   As Oracle user, run the `dbed_update` command to list all the hosts from the repository:

   ```
   $ dbed_update -S $ORACLE_SID -H $ORACLE_HOME \
   -o list
   ```

   As Oracle user, run the `dbed_update` command to rename the old hostname to the new hostname:

   ```
   $ dbed_update -S $ORACLE_SID -H $ORACLE_HOME \
   -o rename old_node=old_name,new_node=new_name
   ```

6. List checkpoints using `dbed_ckptdisplay` and rollback using `dbed_ckptrollback` into appropriate checkpoint.
7. After rollback, start Oracle and let it recover. This procedure assumes archive logs are also replicated and available for oracle recovery.

Creating new Database Storage Checkpoints at the remote site

To make the remote site productive and useful, some customers use the remote site for reporting or backup. This is usually done by stopping replication and running the backup or report, shutting down the database, and restarting replication.

When you are using Database Storage Checkpoint to clone the database and when you are running the cloned database, you must stop replication. To minimize the the down time for replication, Symantec recommends removing the cloned database as soon as you complete your reporting or backup tasks and unmount both the checkpoint file system and the database file system.

For this procedure the SFDB repository need not be part of the replicated volume since it does not reuse any information from the primary site's SFDB repository.

To create a checkpoint at the remote site

1. Stop VVR replication.
2. Run `fsck` for all database volumes on the replicated site.
3. Mount the database file systems.
4. Startup Oracle and recover the database.
5. Create the SFDB repository using `dbed_update`.

   Note: You do not need to use the `-o rename` option because since the SFDB repository information from the primary site is not reused.

   ---

6. Create a checkpoint using `dbed_ckptcreate`.
7. Clone the database using the newly created checkpoint.
8. Run your reporting or backup tasks.
9. Shutdown the cloned database.
10. Unmount and remove the checkpoint if it is no longer required.
11. Unmount the database file systems.
12. Restart VVR replication.

    Note: VVR wreplication will not start if you have not unmounted both the checkpoint file system and the database files system.

Using Database FlashSnap in a VVR environment

Database Flashsnap is the most popular feature of the SFDB tools and it requires the SFDB repository. The Database Flashsnap commands store volume names, plex names and putil2 tags in the SFDB repository. Since volume names, plex names and number of mirrors may differ at the remote site, you cannot use the primary site snapplans at the remote site. When you take a snapshot at the primary site, snapshot volumes are not part of VVR replication. Symantec does not support using snapplans from the primary site at remote site. However, after the application or database switchover to the remote site, you can use Database Flashsnap at a remote site for backup and offhost reporting at the remote site.

To use Flashsnap at the remote site after DR switch over

1. Stop VVR replication.
2. Run `fsck` for all database volumes on the replicated site.
3. Mount the database file systems.
4. Recover or startup the Oracle database.
5. Create the SFDB repository using `dbed_update`.

   Note: You do not need to use the `-o rename` option because since the SFDB repository information from the primary site is not reused.

6. Create FMR3 mirrors and assign putil2 tags if not already done before.
7. Create a new snapplan and validate. This snapplan can be used for Database Flashsnap for creating and resync of snapshots.
8. Create a snapshot.
9. You can start VVR replication immediately after you create the Database Flashsnap cloned database.

Using Database Dynamic Storage Tiering (DBDST) in a VVR environment

The volume tags are not replicated via VVR. This means we do not have same storage class information at the primary and remote sites. But, we do have same volume structure at both sites. DBDST policies are part of File System metadata and hence get replicated to the remote site. VVR makes sure that, if the primary site has a vset, the remote site also has vset with same number of volumes. Fortunately DBDST policies use volume index rather than volume tags internally. This means when a new file is created or old files are extended, the blocks are allocated from same volume index as that of primary site. This means the database or application after DR switch over will not fail.

Here are some good practices if you use DBDST at primary:

- Manually tag the volumes in same order at the remote site and make it same as that of the primary site.
- Whenever you add or change storage class at the primary site make sure similar storage class structure at the remote site.

- Do not apply new DBDST policy at the remote site after switchover. This means do not run any DBDST commands
- Do not run enforce at the remote site.

About deploying Storage Foundation for Databases (SFDB) tools with Oracle Data Guard

The SFDB tools functionality is extended to support standby databases in an Oracle Data Guard environment.

Oracle Data Guard terminology:

- Primary Database: Also referred to as ‘Production Database’, this database is accessed by most of the user applications.
- Standby Database: A standby database is a transactionally consistent copy of the primary database.
- Data Guard Configuration: A Data Guard configuration consists of one production database and one or more standby databases.
- Physical Standby Database: A physically identical copy of the primary database, with on disk database structures that are identical to the primary database on a block-for-block basis.

You can run the following commands on a physical standby database:

Checkpoints

- dbed_ckptcreate
- dbed_ckptdisplay
- dbed_ckptmount
- dbed_ckptumount
- dbed_ckptremove
- dbed_ckptrollback
- dbed_clonedb

Flashsnap

- dbed_vmchecksnap
- dbed_vmsnap
- dbed_vmclonedb (no reverse resync)

Cloning a standby database read-write access

Physical standby databases are read-only databases. Under normal circumstances, they offer limited utility value since you can not update their contents. Using the Data Guard support, feature, DBI commands can clone a read-only standby database into a writable database

Database FlashSnap and Active Data Guard

In environments where the Oracle Active Data Guard feature (available in Oracle 11g) is used, redo log application must be stopped explicitly on standby database before taking a database snapshot.

To use Database FlashSnap in an Active Data Guard environment

1 Stop redo log apply.

   For example:

   ```
   SQL>ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
   ```

2 Run dbed_update.

3 Create a snapplan and validate it.

4 Take a snapshot.

5 Start redo log apply.

   For example:

   ```
   SQL>ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
   ```

6 Clone the snapshot.

For more about using Database Flashsnap:

See “Planning to use Database FlashSnap” on page 149.

Chapter 11

Deploying Storage Foundation for Databases (SFDB) tools with Oracle Enterprise Manager

This chapter includes the following topics:

About Symantec Storage Plug-in for Oracle Enterprise Manager

The Symantec Storage Plugin for Oracle Enterprise Manager (OEM) enables you to view VxFS properties,VxVMvolume and LUN information for database objects such as tablespace, redo logs, controlfile, datafiles and others through the OEM interface.

Requirements

Product requirements are included in:

- *Veritas Storage Foundation™ Release Notes*
- *Veritas Storage Foundation™ for Cluster File System Release Notes*
- *Veritas Storage Foundation™a for Oracle RAC Release Notes*

For Storage Foundation for Databases (SFDB) tools requirements:

See "Requirements for Storage Foundations for Databases (SFDB) tools" on page 38.

Hardware requirements

There are no specific hardware requirements or limitations for the OEM plugin. Consult the requirements in the installation documentation for your Storage Foundation product.

Software requirements

All Storage Foundation Enterprise products support the Oracle Enterprise Manager Plugin.

The Storage Plugin supports the following Oracle database versions:

- Oracle 10gR2
- Oracle 11gR1
- Oracle 10gR2 RAC
- Oracle 11gR1 RAC

Supported configurations

Storage Plugins are supported for the following configurations:

- Storage Foundation with single-instance Oracle
- Storage Foundation HA with single-instance Oracle
- Storage Foundation for Cluster File System with single-instance Oracle
- Storage Foundation for Cluster File System HA with single-instance Oracle
- Storage Foundation for Oracle RAC,with multiple-instance Oracle

Storage Plugins do not support:

- Non-VxFS file systems

- Non-VxVM volumes
- MVFS and volume sets
- Volume or plex rename
- Clone databases created from either checkpoints or snapshots
- RAW datafiles

Before deploying the Storage Plug-in

The following prerequisites must be installed before you can deploy the plug-in:

- Oracle Enterprise Manager Grid Control 10g Release 2 or higher system and Agent
- The Symantec Plug-in can only be deployed on UNIX Agents
- Access must be granted to run plug in as oracle user for VCS plugin

Deploying the Storage Plug-in for Oracle Enterprise Manager

After you ensure that the prerequisites are met, follow these steps to deploy the plug-in

To deploy the Storage Plug-in for OEM

1. Get Symantec_storage.jar from /opt/VRTSdbed/.dba directory or download from Symantec website.
2. Log in to Enterprise Manager Grid Control as a Super Administrator.
3. Click the Setup link in the upper right corner of the Grid Control Home page, then click the Management Plug-ins link on the left side of the Setup page.
4. Click **Import**.
5. Click **Browse** and select the plug-in archive.
6. Click **List Archive**, which lists the plug-ins from the selected archive.
7. Select cluster services plugin if you have Symantec vcs or RAC.
8. Select **DB_storage** if you want to map database objects on Symantec Storage stack.
9. Select the plug-in and click OK.

10 Verify that you have set preferred credentials on all Agents where you want to deploy the plug-in.

11 In the Management Plug-ins page, click the icon in the Deploy column for the Symantec Foundation plug-in. The Deploy Management Plug-in wizard appears.

12 Click **Add Agents**, then select one or more Agents to which you want to deploy the plug-in. The wizard reappears and displays the Agent you selected.

13 Click **Next**, then click **Finish**.

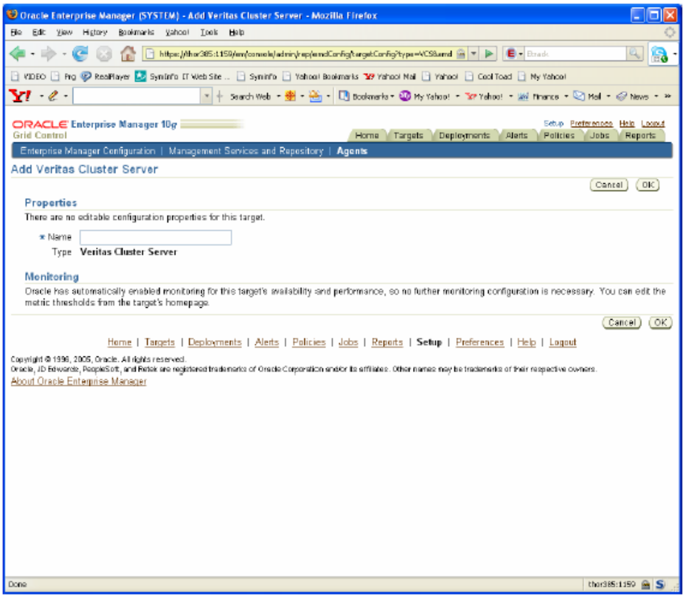
If you see an error message stating that the preferred credential is not set up, go to the Preferences page and add the preferred credentials for the Agent target type.

Adding instances for monitoring

After successfully deploying the plug-in, follow these steps to add the plug-in target to Grid Control for central monitoring and management

To add an instance for monitoring VCS

1 From the Agent home page where the Veritas Cluster Server Plug-in was deployed, select the Veritas Cluster Server target type from the Add drop-down list, then click Go. The Add Veritas Cluster Server page appears.



2 Provide the Name for the plug-in.

3 Click **Test Connection** to make sure the parameters you entered are correct.

4 If the connection test was successful, then click **OK**.

The target home and the reports pages are shown below.

Figure 11-1 Target home page

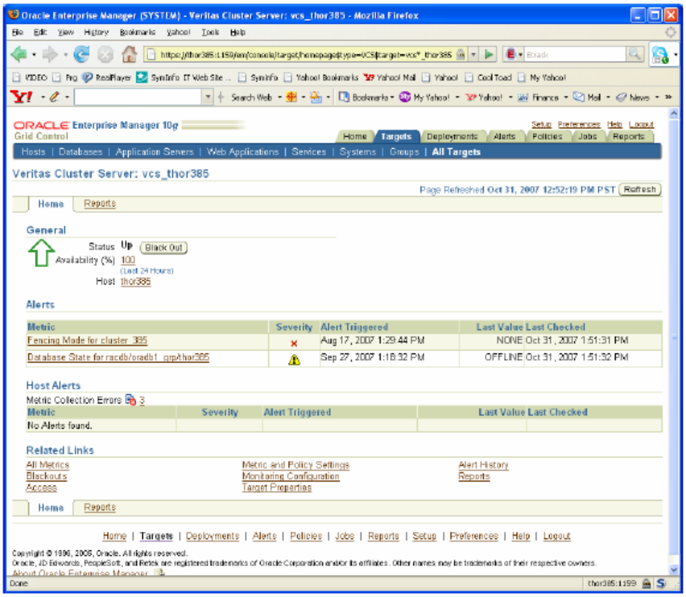
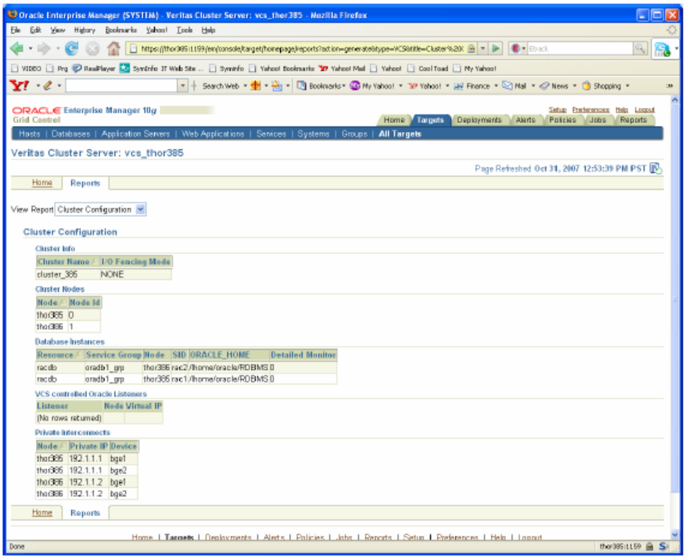
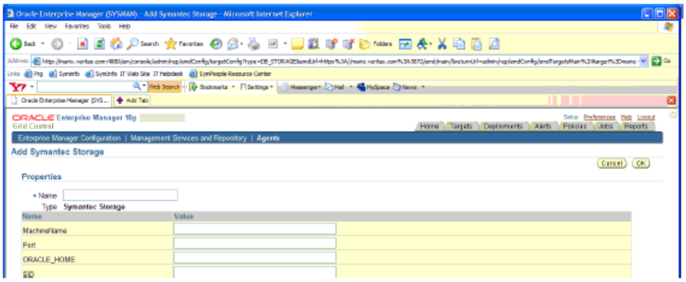


Figure 11-2 VCS reports page



To add an instance for monitoring Symantec Storage Plug-in (DB_STORAGE)

1 In the Agent home page, select the DB_STORAGE target type from the Add drop-down list, then click **GO**. The Add Symantec Storage page appears as below.

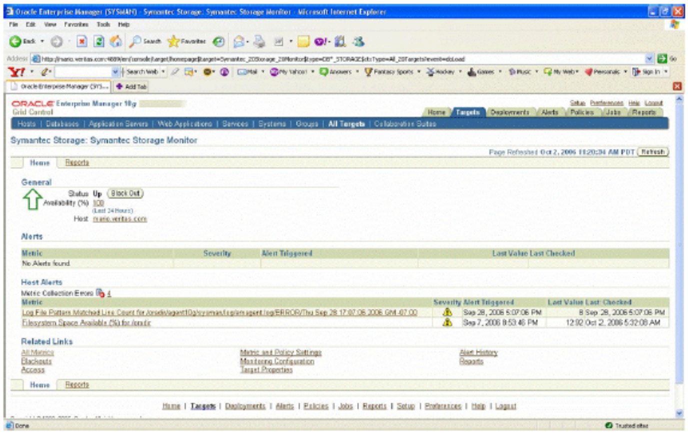


2 Add the requested information for:

- Machine name
- Port
- ORACLE_HOME
- SID
- USERNAME
- password for SID

3 Click **OK**.

The Symantec Storage plug-in home page displays, and the status may take a minute to display.

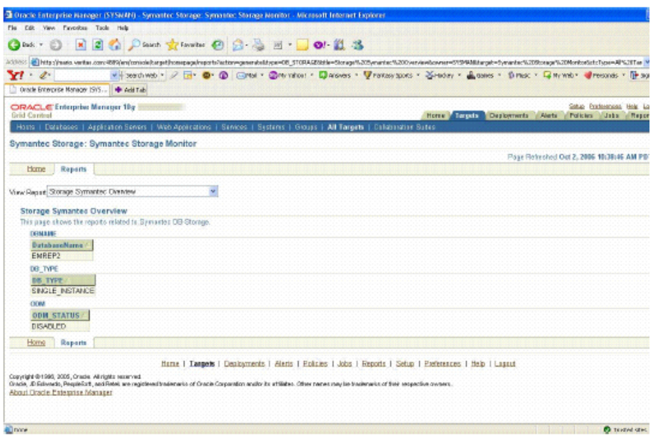


4 When the status displays, select the Reports page. Symantec Storage plug-in home page displays.

Reports

The Reports overview page describes the database information on Symantec Storage software as shown below.

Figure 11-3 Reports overview page

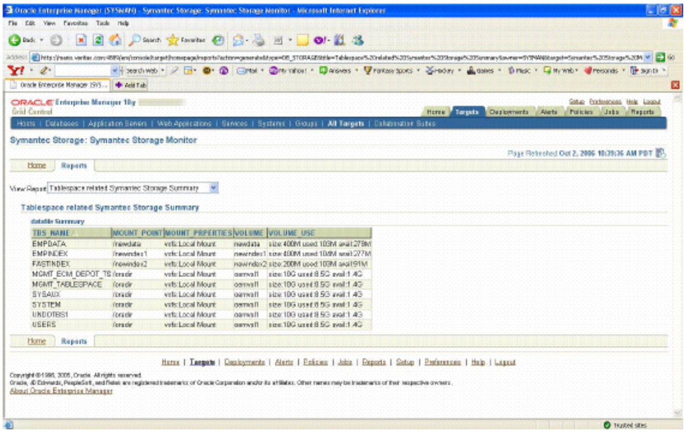


The various database objects are mapped to Symantec storage stack and categorized as tablespace ,datafile,controlfile,redolog file and temp datafile reports.

Tablespace Reports

On this page we map the tablespace name to Symantec based file system mount point, mount properties with volume usage in human readable form.

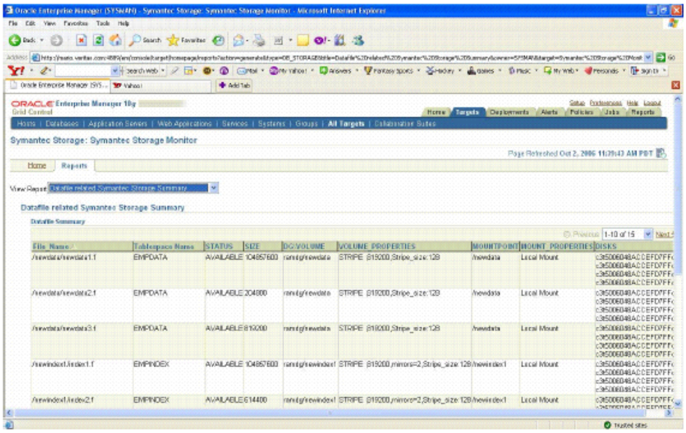
Figure 11-4 Tablespace report



Datafile Report

This report maps datafile and its tablespace to Symantec volumes & file systems with detailed property info and the LUNs being used by the volume containing the datafile.

Figure 11-5 Datafile report



The controlfile, tempfile, and redo log reports are similar to the Datafile Report.

Chapter 12

Troubleshooting Storage Foundation for Databases (SFDB) tools

This chapter includes the following topics:

About troubleshooting Storage Foundation for Databases (SFDB) tools

Storage Foundation for Databases (SFDB) tools are deployed with several Storage Foundation products, and as a result can be affected by any issue with those products. The first step in case of trouble should be to identify the source of the problem. It is rare to encounter problems in Storage Foundation for Databases (SFDB) tools; more commonly the problem can be traced to setup issues or problems in the base products.

Use the information in this chapter to diagnose the source of problems. Indications may point to base product set up or configuration issues, in which case solutions may require reference to other Storage Foundation documentation. In cases where indications point to a component product or to Oracle as the source of a problem, it may be necessary to refer to the appropriate documentation to resolve it.

For troubleshooting Storage Foundation product issues:

- *Veritas File System Administrators Guide*
- *Veritas Volume Manager Administrators Guide*
- *Veritas Storage Foundation for Cluster File System Administrators Guide*
- *Veritas Storage Foundation for Oracle RAC Administrators Guide*

Running scripts for engineering support analysis

Troubleshooting scripts gather information about the configuration and status of your product and its modules. The scripts identify package information, debugging messages, console messages, and information about disk groups and volumes. Forwarding the output of these scripts to Symantec Tech Support can assist with analyzing and solving any problems.

To obtain SFDB repository and log information

◆ Run:

```
# /opt/VRSTspt/VRTSexplorer/VRTSexplorer
```

Send the output to Support.

SFDB Log files

Checking the following log files can provide useful diagnistic information.

SFDB tools commands log files are located in the /var/vx/vxdba/logs directory.

About the vxdbd daemon

The `vxdbd` daemon handles communication to and from the Veritas Storage Foundation product software. By default, `vxdbd` communicates with product over port number 3233. If there are conflicts with this port or other port-related problems, you can change the port by changing the `VXDBD_SOCKET` setting located in the `/etc/vx/vxdbed/admin.properties` file.

Normally the `vxdbd` daemon starts automatically when the host boots up. However, if the daemon reports errors or if the daemon process dies, you may have to manually start or stop it. There is also a `status` command that reports the current state of the daemon to confirm that it is currently running.

Only the root user can stop `vxdbd`. Any user can start `vxdbd` or display its status.

Note: You must have a valid HOME directory for `vxdbd` to work correctly with several Veritas Storage Foundation for Oracle features. If you receive the following error message, but you confirm that vxdbd is running (using `ps -ef | grep vxdbd` rather than `vxdbdctrl status`), you may not have a valid HOME directory or it may not be available to `vxdbd`: VXDBA_PRODUCT exec_remote ERROR V-81-7700 Can not connect to the vxdbd.

To see the status of the vxdbd daemon

- Use the `vxdbdctrl status` command:

  ```
  /opt/VRTSdbed/common/bin/vxdbdctrl status
  ```

  If the daemon is running you see the following output:

  ```
  Status of Veritas vxdbd
  /opt/VRTSdbed/common/bin/vxdbd ping SUCCESS
  ```

To start the vxdbd daemon

- Use the `vxdbdctrl start` command:

  ```
  /opt/VRTSdbed/common/bin/vxdbdctrl start
  ```

To stop the vxdbd daemon

- As root, use the `vxdbdctrl stop` command:

  ```
  /opt/VRTSdbed/common/bin/vxdbdctrl stop
  ```

To change the communications port used by the vxdbd daemon

1. As the root user, stop the `vxdbd` daemon:

   ```
   /opt/VRTSdbed/common/bin/vxdbdctrl stop
   ```

2. In the `/etc/vx/vxdbed/admin.properties` file, change the value of the VXDBD_SOCKET variable to a new port number:

   ```
   VXDBD_SOCKET=3233
   ```

3 Restart the vxdbd daemon:

```
/opt/VRTSdbed/common/bin/vxdbdctrl start
```

4 If the system is part of a multi-host configuration, change the port on all hosts involved by repeating this procedure on each host.

The vxdbd daemon log is located in /var/vx/vxdba/logs/vxdbd.log. If you have trouble with the vxdbd daemon, you can turn on debug mode.

To enable the debug mode

1 Uncomment the following two lines in /opt/VRTSdbed/common/bin/vxdbdctrl:

```
#VXDBDDEBUG=108
  #export VXDBDDEBUG
```

2 Restart the vxdbd daemon by running /opt/VRTSdbed/common/bin/vxdbdctrl stop and /opt/VRTSdbed/common/bin/vxdbdctrl start.

3 Rerun the problem commands and debugged messages will be generated in /var/vx/vxdba/logs/vxdbd.log.

4 Then run vxexplorer to save all the log files to send to support.

Troubleshooting Database FlashSnap

If the Database Flashsnap commands fail, review the vxsnapadm_50.log.

The vxsnapadm_50.log is located at:

```
/var/vx/vxdba/logs/vxsnapadm_50.log
```

Troubleshooting Database Dynamic Storage Tiering

If the Database Dynamic Storage Tiering commands fail as in the following example, review the tsdb_debug.log.

The tsdb_debug.log is located at:

```
/var/vx/vxdba/logs/tsdb_debug.log
```

For example, when the following message appears after issuing a `dbdst_addvol` command review the tsdb_debug.log:

```
$ /opt/VRTS/bin/dbdst_addvol -S BLM21 -M /dev/vx/dsk/nobody_data1/\
data1vol -v new_vol1:fast_storage,new_vol2:slow_storage,new_vol3:slow_storage
```

```
SFORA dbdst_addvol ERROR V-81-6222 Could not add volume new_vol1 to vset
```

The tsdb_debug.log file contains information that resembles the following:

```
# view /var/vx/vxdba/logs/tsdb_debug.log
1216606 Tue May 13 10:11:05 2008
 /opt/VRTS/bin/dbdst_addvol -S BLM21 -M /dev/vx/dsk/nobody_data1/data1vol -v
new _vol1:fast_storage,new_vol2:slow_storage,
new_vol3:slow_storage
1216606 Tue May 13 10:11:09 2008
RACmaster = editor
1216606 Tue May 13 10:11:09 2008
editor:/opt/VRTSdbed/common/.dba/tsdb_setup.sh -g nobody_data1 -o addvol -d\
data1vol - v new_vol1 -m /oradata1 -t vxfs.placement_class.FAST_STORAGE
1216606 Tue May 13 10:11:09 2008
command failed, ret=1

1216606 Tue May 13 10:11:09 2008
tsdb_setup.sh arguments -g nobody_data1 -o addvol -d data1vol -v
 new_vol1 -m /or
adata1 -t vxfs.placement_class.FAST_STORAGE
05/13/08@17:08:11
size of volume  new_vol1 is 204800
VxVM vxvset ERROR V-5-1-10035 Volume set data1vol contains volume(s)
in snapshot chain.
This can cause inconsistencies in the snapshot hierarchy. Specify "-f" option
to force the operation.
^^^^ NOTE: here is the reason for the failure, barried in this log file:
^^^^ /var/vx/vxdba/logs/tsdb_debug.log
Can not add  to data1vol, ERR 1
ERROR:1
```

Sample configuration files for clustered deployments

This appendix includes the following topics:

About sample configuration files

The sample configuration files illustrate several deployment scenarios:

- Storage Foundation for High Availability
- Storage Foundation for Clustered File System HA
- Storage Foundation for Oracle RAC

You may use the sample files as a guideline for setting up your environment.

Sample configuration files with the Oracle agent resources is located in the following directory:

- For SF HA: /etc/VRTSagents/ha/conf/Oracle
- For SFCFS HA: /etc/VRTSagents/ha/conf/Oracle
- For SF Oracle RAC: /etc/VRTSvcs/conf/sample_rac

Sample configuration file for Storage Foundation for High Availability

```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster devha (
        UserNames = { admin = anoGniNkoJooMwoInl }
        Administrators = { admin }
        )

system paint (
        )

system quarter-sol10 (
        )

group Oracle_Group (
        SystemList = { paint = 0, quarter-sol10 = 1 }
        AutoStartList = { paint }
        )

        DiskGroup DG-orabindg (
                DiskGroup = qorasrc
                PanicSystemOnDGLoss = 1
                )

        DiskGroup DG-oradatadg (
                DiskGroup = oradatadg
                PanicSystemOnDGLoss = 1
                )

        IP Ora_IP (
                Device = bge0
                Address = "10.182.187.108"
                NetMask = "255.255.248.0"
                )

        Mount ora_arch1_mnt (
```

```
        MountPoint = "/oraarch"
        BlockDevice = "/dev/vx/dsk/oradatadg/oraarchvol"
        FSType = vxfs
        FsckOpt = "-n"
        )

Mount ora_bin_mnt (
        MountPoint = "/opt/oracle/orahome"
        BlockDevice = "/dev/vx/dsk/qorasrc/orasrc10g"
        FSType = vxfs
        FsckOpt = "-n"
        )

Mount ora_data1_mnt (
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
        FSType = vxfs
        FsckOpt = "-n"
        )

NIC Ora_NIC (
        Device = bge0
        )

Oracle Ora_Oracle (
        Sid = ora10g
        Owner = oracle
        Home = "/opt/oracle/orahome"
        Pfile = "/opt/oracle/orahome/dbs/initora10g.ora"
        AgentDebug = 1
        )

Ora_IP requires Ora_NIC
Ora_Oracle requires ora_arch1_mnt
Ora_Oracle requires ora_bin_mnt
Ora_Oracle requires ora_data1_mnt
ora_arch1_mnt requires DG-oradatadg
ora_bin_mnt requires DG-orabindg
ora_data1_mnt requires DG-oradatadg


// resource dependency tree
//
```

```
//      group Oracle_Group
//      {
//      IP Ora_IP
//          {
//          NIC Ora_NIC
//          }
//      Oracle Ora_Oracle
//          {
//          Mount ora_arch1_mnt
//              {
//              DiskGroup DG-oradatadg
//              }
//          Mount ora_bin_mnt
//              {
//              DiskGroup DG-orabindg
//              }
//          Mount ora_data1_mnt
//              {
//              DiskGroup DG-oradatadg
//              }
//          }
//      }
```

Sample configuration file for Storage Foundation for Clustered File System HA

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster cfgha (
        UserNames = { admin = HopHojOlpKppNxpJom }
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        )

system cdgv245c (
        )
```

```
system cdgv245d (
        )

group Oracle_Group (
        SystemList = { cdgv245c = 0, cdgv245d = 1 }
        AutoStartList = { cdgv245c }

        )

        IP Ora_IP (
                Device = bge0
                Address = "10.200.117.243"
                NetMask = "255.255.255.0"
                )

        NIC Ora_NIC (
                Device = bge0
                )

        Oracle Ora_Oracle (
                Sid = ora11g
                Owner = oracle
                Home = "/opt/oracle/orahome"
                Pfile = "/opt/oracle/orahome/dbs/initora11g.ora"
                AgentDebug = 1
                )

        Ora_IP requires Ora_NIC
        Ora_Oracle requires Ora_IP


        // resource dependency tree
        //
        //      group Oracle_Group
        //      {
        //      Oracle Ora_Oracle
        //          {
        //          IP Ora_IP
        //              {
        //              NIC Ora_NIC
        //              }
        //          }
```

```
        //      }


group cvm (
        SystemList = { cdgv245c = 0, cdgv245d = 1 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { cdgv245c, cdgv245d }
        )

        CFSMount oraarch_mnt (
                Critical = 0
                MountPoint = "/oraarch"
                BlockDevice = "/dev/vx/dsk/oradatadg/oraarchvol"
                )

        CFSMount orabin_mnt (
                Critical = 0
                MountPoint = "/ora11"
                BlockDevice = "/dev/vx/dsk/ora11bin/ora11vol"
                )

        CFSMount oradata2_mnt (
                Critical = 0
                MountPoint = "/oradata2"
                BlockDevice = "/dev/vx/dsk/oradatadg2/oradatavol2"
                )


        CFSMount oradata_mnt (
                Critical = 0
                MountPoint = "/oradata"
                BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
                )

        CFSMount oraredo_mnt (
                Critical = 0
                MountPoint = "/oraredo"
                BlockDevice = "/dev/vx/dsk/oradatadg2/oraredovol"
                )

        CFSfsckd vxfsckd (
                )
```

```
CVMCluster cvm_clus (
        CVMClustName = cfgha
        CVMNodeId = { cdgv245c = 0, cdgv245d = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )


CVMVolDg orabin_voldg (
        CVMDiskGroup = orabindg
        CVMVolume = { oracle_volume }
        CVMActivation = sw
        )

CVMVolDg oradata2_voldg (
        CVMDiskGroup = oradatadg2
        CVMVolume = { oradatavol2, oraredovol }
        CVMActivation = sw
        )

CVMVolDg oradata_voldg (
        CVMDiskGroup = oradatadg
        CVMVolume = { oradatavol, oraarchvol }
        CVMActivation = sw
        )

CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

cvm_clus requires cvm_vxconfigd
oraarch_mnt requires oradata_voldg
oraarch_mnt requires vxfsckd
orabin_mnt requires vxfsckd
orabin_voldg requires cvm_clus
oradata2_mnt requires oradata2_voldg
oradata2_mnt requires vxfsckd
oradata2_voldg requires cvm_clus
oradata_mnt requires oradata_voldg
oradata_mnt requires vxfsckd
oradata_voldg requires cvm_clus
```

```
oraredo_mnt requires oradata2_voldg
oraredo_mnt requires vxfsckd
vxfsckd requires cvm_clus


// resource dependency tree
//
//      group cvm
//      {
//      CFSMount oraarch_mnt
//          {
//          CVMVolDg oradata_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CFSMount orabin_mnt
//          {
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CVMVolDg orabin_voldg
//          {
//          CVMCluster cvm_clus
//              {
//              CVMVxconfigd cvm_vxconfigd
//              }
//          }
```

```
//      CFSMount oradata2_mnt
//          {
//          CVMVolDg oradata2_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CVMVolDg orabin_voldg
//          {
//          CVMCluster cvm_clus
//              {
//              CVMVxconfigd cvm_vxconfigd
//              }
//          }
//      CFSMount oradata2_mnt
//          {
//          CVMVolDg oradata2_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CFSMount oradata_mnt
//          {
//          CVMVolDg oradata_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
```

```
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CFSMount oraredo_mnt
//          {
//          CVMVolDg oradata2_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      }
```

Sample configuration file for Storage Foundation for Oracle RAC

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CVMTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"

cluster sfrac_sample (
        UserNames = { admin = hqrJqlQnrMrrPzrLqo,
                 "root@sxsvm02" = 0,
```

```
                 "root@sxsvm03" = 0 }
        Administrators = { admin, "root@sxsvm02",
                 "root@sxsvm03" }
        SecureClus = 1
        UseFence = SCSI3
        HacliUserLevel = COMMANDROOT
        )

system sxsvm02 (
        )

system sxsvm03 (
        )

group VxSS (
        SystemList = { sxsvm02 = 0, sxsvm03 = 1 }
        Parallel = 1
        AutoStartList = { sxsvm02, sxsvm03 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Phantom phantom_vxss (
                )

        ProcessOnOnly vxatd (
                IgnoreArgs = 1
                PathName = "/opt/VRTSat/bin/vxatd"
                )



        // resource dependency tree
        //
        //      group VxSS
        //      {
        //      Phantom phantom_vxss
        //      ProcessOnOnly vxatd
        //      }


group cvm (
        SystemList = { sxsvm02 = 0, sxsvm03 = 1 }
```

```
AutoFailOver = 0
Parallel = 1
AutoStartList = { sxsvm02, sxsvm03 }
)

Application cssd (
        Critical = 0
        StartProgram = "/opt/VRTSvcs/rac/bin/cssd-online"
        StopProgram = "/opt/VRTSvcs/rac/bin/cssd-offline"
        CleanProgram = "/opt/VRTSvcs/rac/bin/cssd-clean"
        MonitorProgram = "/opt/VRTSvcs/rac/bin/cssd-monitor"
        )

CFSMount ocrmnt (
        Critical = 0
        MountPoint = "/ocr_vote"
        BlockDevice = "/dev/vx/dsk/ocrdg/ocrvol"
        )

CFSfsckd vxfsckd (
        )

CVMCluster cvm_clus (
        CVMClustName = sfrac_rp2
        CVMNodeId = { sxsvm02 = 0, sxsvm03 = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )

CVMVolDg ocrvoldg (
        Critical = 0
        CVMDiskGroup = ocrdg
        CVMVolume = { ocrvol }
        CVMActivation = sw
        )

CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

PrivNIC ora_priv (
        Critical = 0
```

```
        Device @sxsvm02 = { bge1 = 0, bge2 = 1 }
        Device @sxsvm03 = { bge1 = 0, bge2 = 1 }
        Address @sxsvm02 = "192.168.12.1"
        Address @sxsvm03 = "192.168.12.2"
        NetMask = "255.255.240.0"
        )

cssd requires ocrmnt
cssd requires ora_priv
cvm_clus requires cvm_vxconfigd
ocrmnt requires ocrvoldg
ocrmnt requires vxfsckd
ocrvoldg requires cvm_clus
vxfsckd requires cvm_clus


// resource dependency tree
//
//      group cvm
//      {
//      Application cssd
//          {
//          CFSMount ocrmnt
//              {
//              CVMVolDg ocrvoldg
//                  {
//                  CVMCluster cvm_clus
//                      {
//                      CVMVxconfigd cvm_vxconfigd
//                      }
//                  }
//              CFSfsckd vxfsckd
//                  {
//                  CVMCluster cvm_clus
//                      {
//                      CVMVxconfigd cvm_vxconfigd
//                      }
//                  }
//              }
//          PrivNIC ora_priv
//          }
//      }
```

```
group ora_db (
        SystemList = { sxsvm02 = 0, sxsvm03 = 1 }
        AutoStart = 0
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { sxsvm02, sxsvm03 }
        )

        CFSMount archive_mnt (
                Critical = 0
                MountPoint = "/oraarchive"
                BlockDevice = "/dev/vx/dsk/oradg/archivevol"
                )

        CFSMount ora_data_mnt (
                Critical = 0
                MountPoint = "/oradata"
                BlockDevice = "/dev/vx/dsk/oradg/oradatavol"
                )

        CVMVolDg ora_data_voldg (
                Critical = 0
                CVMDiskGroup = oradg
                CVMVolume = { oradatavol }
                CVMActivation = sw
                )

        Oracle ORACLE (
                Sid @sxsvm02 = orarp1
                Sid @sxsvm03 = orarp2
                Owner = oracle
                Home = "/oracle/app/orahome"
                Pfile @sxsvm02 = "/oradata/orarp/spfileorarp.ora"
                Pfile @sxsvm03 = "/oradata/orarp/spfileorarp.ora"
                StartUpOpt = SRVCTLSTART
                ShutDownOpt = SRVCTLSTOP
                )

        requires group cvm online local firm
        ORACLE requires archive_mnt
        ORACLE requires ora_data_mnt
        archive_mnt requires ora_data_voldg
```

```
ora_data_mnt requires ora_data_voldg


// resource dependency tree
//
//      group ora_db
//      {
//      Oracle ORACLE
//          {
//          CFSMount archive_mnt
//              {
//              CVMVolDg ora_data_voldg
//              }
//          CFSMount ora_data_mnt
//              {
//              CVMVolDg ora_data_voldg
//              }
//          }
//      }
```

Appendix B

Storage Foundation for Databases (SFDB) tools command line interface

This appendix includes the following topics:

About the command line interface

You can use the SFDB command line interface to perform administrative operations. For more detailed information about the commands and their syntax and available options, see the individual manual pages.

Updating the repository using dbed_update

You can use the `dbed_update` command to update the repository.

Any time you change the structure of the database (for example, by adding or deleting datafiles, converting `PFILE` to `SPFILE`, or converting `SPFILE` to `PFILE`), you must run `dbed_update`.

Before updating the repository, review the following information:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`).
Usage notes	■ The `dbed_update` command saves or updates the information related to the Oracle database in the SFDB repository. ■ The database must be up and running, and the `ORACLE_SID` and the `ORACLE_HOME` variable arguments must be specified with the `-S` and `-H` options, respectively. ■ See the `dbed_update`(1M) manual page for more information.

Options for updating the repository:

Option	Sample value	Description
-S	ORACLE_SID	Specifies the name of the Oracle database whose information will be retrieved.
-H	ORACLE_HOME	The ORACLE_HOME setting for the ORACLE_SID database.
-G	SERVICE_GROUP	Specifies the VCS service group name for the ORACLE_SID database if it is under VCS control. This option is required in an HA environment.
-P	ORACLE_PFILE	Specifies the fully qualify path of the Oracle pfile if it is not located under ORACLE_HOME/dbs directory.
-R	REPOSITORY_PATH	Specifies a user-defined location for the repository. In a RAC or HA situation, it should be located on shared storage. The default location is on the mountpoint of the SYSTEM tablespace of the Oracle database. This way it is shared easily among cluster nodes. If REPOSITORY_PATH is slash (/), the repository location is switched back to the default path.

Option	Sample value	Description
-I	N/A	An integrity check of the repository is performed, without refreshing from the Oracle database . If successful, the string "ok" is printed to stdout. In addition, the full path of the repository file is shown. Whenever refreshing from the database, an integrity check is automatically performed.
-o	list	The list of hosts and their internal ID is shown.
-o	rename	Rename a host in the repository, by specifying old and new name. This is useful for DR after a failover, because the hostnames on the DR site differ from the original ones. For example: `-o rename,old_node=old_name,new_node=new_name`

To update the SFDB repository

◆ Use the `dbed_update` command as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product
```

Creating Storage Checkpoints using dbed_ckptcreate

You can use the `dbed_ckptcreate` command to create a Storage Checkpoint from the command line.

Storage Checkpoints can be either online, offline, or instant. By default, Storage Checkpoints are offline. If `online` is specified, the database is put into hot-backup mode when the Storage Checkpoint is created. If `offline` is specified, the database is expected to be down. If `instant` is specified, the database must be online and a Storage Checkpoint will be taken for a “crash recovery”-type Storage Rollback.

Before creating a Storage Checkpoint, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`). ■ For best recoverability, always keep `ARCHIVELOG` mode enabled when you create Storage Checkpoints.
Usage notes	■ The *dbed_ckptcreate* command stores Storage Checkpoint information in the SFDB repository. ■ See the dbed_ckptcreate(1M) manual page for more information.

To create Storage Checkpoints while the database is online

◆ Use the `dbed_ckptcreate` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o online

Storage Checkpoint Checkpoint_971672042 created.
```

To create Storage Checkpoints without updating the repository while the database is online

◆ Use the `dbed_ckptcreate` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o online -n

Storage Checkpoint Checkpoint_971672043 created.
```

To create Storage Checkpoints while the database is offline

◆ Use the `dbed_ckptcreate` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o offline

Storage Checkpoint Checkpoint_971672044 created.
```

The default option is `offline`.

To create instant Storage Checkpoints

◆ Ensure that the database is online and use the `dbed_ckptcreate` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o instant

Storage Checkpoint Checkpoint_971672045 created.
```

Scheduling Storage Checkpoints using dbed_ckptcreate and cron

You can use the `dbed_ckptcreate` command to schedule Storage Checkpoint creation in a `cron` job or other administrative script.

Before scheduling Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`).
Usage notes	■ Although `dbed_ckptcreate` is supported by Veritas Storage Foundation for Oracle RAC, the scheduling feature is not supported. ■ Create a new `crontab` file or edit an existing `crontab` file to include a Storage Checkpoint creation entry with the following space-delimited fields: minute hour day_of_month month_of_year day_of_week\ /opt/VRTS/bin/dbed_ckptcreate where: minute - numeric values from 0-59 or * hour - numeric values from 0-23 or * day_of_month - numeric values from 1-31 or * month_of_year - numeric values from 1-12 or * day_of_week- numeric values from 0-6, with 0=Sunday or * Each of these variables can either be an asterisk (meaning all legal values) or a list of elements separated by commas. An element is either a number or two numbers separated by a hyphen (meaning an inclusive range). ■ See the `dbed_ckptcreate`(1M), `cron`(1M), and `crontab`(1) manual pages for more information.

Scheduling Storage Checkpoint creation in a cron job

- To create a Storage Checkpoint twice a day, at 5:00 a.m. and 7:00 p.m., every Monday through Friday, include the following entry in your `crontab` file:

```
0 5,19 * * 1-5 /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o instant
```

- To create a Storage Checkpoint at 11:30 p.m., on the 1st and 15th day of each month, include the following entry in your `crontab` file:

```
30 23 1,15 * * /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o instant
```

- To create a Storage Checkpoint at 1:00 a.m. every Sunday while the database is offline, include the following entry in your crontab file:

```
0 1 * * 0 /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product -o offline
```

Displaying Storage Checkpoints using dbed_ckptdisplay

You can use the dbed_ckptdisplay command to display the Storage Checkpoints associated with an Oracle database from the command line.

You can also use it to display fileset quota values.

Before displaying Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You may be logged in as either the database administrator or root. If you execute the command as root, use the -n option.
Usage Notes	■ In addition to displaying the Databse Checkpoints created by the SFDB tools, dbed_ckptdisplay also displays other Storage Checkpoints (for example, Storage Checkpoints created by NetBackup). ■ The **Status** field identifies if the Storage Checkpoint is partial (P), complete (C), invalid (I), mounted (M), read-only (R), writable (W), or of type online (ON), offline (OF), instant (IN), or unknown (UN). ■ Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply. ■ See the dbed_ckptdisplay(1M) manual page for more information.

To display Database Checkpoints

◆ Use the dbed_ckptdisplay command as follows to display information for Storage Checkpoints:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product

Storage Checkpoint          Creation Time               Status
Checkpoint_1239780697       Wed Apr 15 13:01:37 2009    C+R+ON

Checkpoint_974424522_wr001  Thu May 16 17:28:42 2009    C+R+ON
```

To display other Storage Checkpoints

- Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product -o other
```

```
Storage Checkpoint                    Creator   Location
---------------------------------     -------   --------
NetBackup_incr_PROD_955133480         NBU       /db01
NetBackup_full_PROD_9551329 52        NBU       /db01
```

To display other Storage Checkpoints without updating the repository

- Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product -o other -n
```

```
NetBackup_incr_PROD_955133480            NBU   /db01
NetBackup_full_PROD_9551329          52  NBU   /db01
```

To display all Storage Checkpoints

- Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product -o all
```

```
Checkpoint_971672042       Sun May 15 13:55:53 2005        C+R+IN

Checkpoint_903937870       Fri May 13 22:51:10 2005        C+R+ON

Checkpoint_901426272       Wed May 11 16:17:52 2005        P+R+ON

NetBackup_incr_PROD_955133480            NBU   /db01

NetBackup_full_PROD_9551329          52  NBU   /db01
```

To display all Storage Checkpoints without updating the repository

◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product -o all -n

Checkpoint_971672042       Sun May 15 13:55:53 2005        C+R+IN

Checkpoint_903937870       Fri May 13 22:51:10 2005        C+R+ON

Checkpoint_901426272       Wed May 11 16:17:52 2005        P+R+ON

NetBackup_incr_PROD_955133480           NBU   /db01
NetBackup_full_PROD_9551329         52  NBU   /db01
```

To display fileset quota values

◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD -c \
Checkpoint_903937870 -Q

Checkpoint_903937870        Wed Mar 19 9:12:20 2005        C+R+ON

Filesystem            HardLim   SoftLim   CurrentUse

/oradata1/indx1_1     100000    50000     2028

/oradata1/user1_1     100000    50000     2028

/oradata1/temp        150000    80000     2142

/oradata1/system1     150000    70000     3092
```

Mounting Storage Checkpoints using dbed_ckptmount

You can use the `dbed_ckptmount` command to mount a Storage Checkpoint for the database from the command line.

Before mounting Storage Checkpoints, review the following information:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`.)

Usage notes	■ The `dbed_ckptmount` command is used to mount a Storage Checkpoint into the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands. ■ Storage Checkpoints can be mounted as read-only or read-write. By default, Storage Checkpoints are mounted as read-only. ■ If the `rw` (read-write) option is used, `_wrxxx`, where `xxx` is an integer, will be appended to the Storage Checkpoint name. ■ If the specified mount point directory does not exist, then `dbed_ckptmount` creates it before mounting the Storage Checkpoint, as long as the Oracle database owner has permission to create it. ■ Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply. ■ See the `dbed_ckptmount`(1M) manual page for more information.

To mount Storage Checkpoints with the read/write option

◆ Use the `dbed_ckptmount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_971672042 \
-m /tmp/ckpt_rw -o rw
```

```
Creating Storage Checkpoint on /tmp/ckpt_rw/share/oradata with
name Checkpoint_971672042_wr001
```

Unmounting Storage Checkpoints using dbed_ckptumount

You can use the `dbed_ckptumount`command to unmount a Storage Checkpoint from the command line.

Before unmounting Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`.)
Usage notes	■ See the `dbed_ckptumount`(1M) manual page for more information.

To unmount Storage Checkpoints

◆ Use the `dbed_ckptumount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptumount -S PROD \
-c Checkpoint_971672042_wr001
```

Administering Storage Checkpoint quotas using dbed_ckptquota

You can use the `dbed_ckptquota` command to administer file system quotas for Storage Checkpoint for a database from the command line.

Before administering Storage Checkpoint quotas, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`). ■ The repository entry for the database must exist and the DBA must be the owner of all file systems to be affected.
Usage notes	■ See the `dbed_ckptquota`(1M) manual page for more information.

To set quota limits for all file systems in the database and enable quota enforcement

◆ Use the `dbed_ckptquota` command as follows to set the hard and soft limits for all file systems in the database and enable quota enforcement:

```
$ /opt/VRTS/bin/dbed_ckptquota -S PROD -H /ora10i \
-o set=50000,40000,enable
```

To set quota limits for all file systems specified in a list file

◆ Use the `dbed_ckptquota` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptquota -S PROD -H /ora10i \
-o set=25000,20000 -f quotacfg
```

To disable quota limits for a file system

◆ Use the `dbed_ckptquota` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptquota -S PROD -H /ora10i \
-o disable /ora/testvol03
```

To display quota values for all file systems in the database

- Use the `dbed_ckptquota` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptquota -S PROD -H /ora10i \
-o display
```

```
Filesystem            Hardlimit    Softlimit    CurrentUse

/ora/prod             50000        40000        136

/ora/testvol01        25000        20000        128

/ora/testvol02        50000        40000        128

/ora/testvol03        50000        40000        0

/ora/testvol04        25000        20000        128

/ora/testvol05        50000        40000        128
```

The numbers in the “Hardlimit” and “Softlimit” columns represent the total numbers of file system blocks allowed.

`CurrentUse` displays the number of filesystem blocks currently used by all Storage Checkpoints in the filesystem. If there are no Storage Checkpoints, or if quotas have been disabled, `CurrentUse` will display 0.

Performing Storage Rollback using dbed_ckptrollback

You can use the `dbed_ckptrollback` command to rollback an Oracle database to a Storage Checkpoint.

Before performing a Storage Rollback, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID `oracle`.)

Usage notes

- The `dbed_ckptrollback` command rolls an Oracle database back to a specified Storage Checkpoint. You can perform a Storage Rollback for the entire database, a specific tablespace, or list of datafiles.

  Database rollback for the entire database requires that the database be inactive before Storage Rollback commences. The `dbed_ckptrollback` command will not commence if the Oracle database is active. However, to perform a Storage Rollback of a tablespace or datafile, only the tablespace or datafile to be rolled back must be offline (not the entire database).
- You must run the `dbed_update` command after upgrading to Veritas Storage Foundation Product version: 5.1 for Oracle from a previous release. This will allow you to roll back to a Storage Checkpoint that was created with an earlier version of this product.
- See the `dbed_ckptrollback`(1M) manual page for more information.

To roll back an Oracle database to a Storage Checkpoint

◆ Use the `dbed_ckptrollback`command as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD -H \
/oracle/product -c Checkpoint_903937870
```

To rollback a tablespace to a Storage Checkpoint

◆ Use the `dbed_ckptrollback` command with the `-T` option as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD -H \
/oracle/product -T DATA01 -c Checkpoint_903937870
```

If the Oracle database is running, you must take the tablespace offline before running this command. If the tablespace is online, the command will fail.

In the case of an instant Storage Checkpoint, rolling back a tablespace does not apply.

To rollback datafiles to a Storage Checkpoint

◆ Use the `dbed_ckptrollback` command with the `-F` option as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD -H /oracle/product \
-F /share/oradata1/data01.dbf /share/oradata2/index01.dbf \
-c Checkpoint_903937870
```

If the Oracle database is running, you must take the datafile offline before running this command. If the datafile is online, the command will fail.

In the case of an instant Storage Checkpoint, rolling back datafiles does not apply.

Removing Storage Checkpoints using dbed_ckptremove

You can use the `dbed_ckptremove`command to remove a Storage Checkpoint for an Oracle database at the command line.

Before removing Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You may be logged in as either the database administrator or `root`.
Usage notes	■ The `dbed_ckptremove`command is used to remove a Storage Checkpoint from the file system, or file systems, it is associated with. The Storage Checkpoint must have been created using the GUI or the `dbed_ckptcreate`(1M) command. ■ You must unmount the Storage Checkpoint before you can remove it. ■ See the `dbed_ckptremove`(1M) manual page for more information.

To remove Storage Checkpoints

◆ Use the `dbed_ckptremove` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptremove -S PROD \
-c Checkpoint_971672042_wr001
```

Cloning the Oracle instance using dbed_clonedb

You have the option to manually or automatically recover the database when using the `dbed_clonedb` command:

- Manual (interactive) recovery, which requires using the `-i` option, of the clone database allows the user to control the degree of recovery by specifying which archive log files are to be replayed.
- Automatic (non-interactive) recovery, which is the default usage of the command, recovers the entire database and replays all of the archive logs. You will not be prompted for any archive log names.

Before cloning the Oracle instance, review the following information:

Prerequisites	■ You must first create a Storage Checkpoint. See "Creating Storage Checkpoints using dbed_ckptcreate" on page 282. ■ You must be logged in as the database administrator. ■ Make sure you have enough space and system resources to create a clone database on your system. ■ A clone database takes up as much memory and machine resources as the primary database.
Usage notes	■ In an Oracle RAC environment, cloning an Oracle database with an instant Storage Checkpoint is not supported. ■ The `dbed_clonedb` command is used to create a copy of a database, cloning all existing database files to new locations. ■ The ORACLE_SID and ORACLE_HOME environment variables must be set to the primary database. ■ It is assumed that the user has a basic understanding of the database recovery process. ■ See the `dbed_clonedb`(1M) manual page for more information.

Table B-1 lists the options for cloning the Oracle database.

Table B-1 dbed_clonedb command options

Option	Description
`-S CLONE_SID`	Specifies the name of the new Oracle SID, which will be the name of the new database instance.
`-m MOUNT_POINT`	Indicates the new mount point of the Storage Checkpoint.
`-c CKPT_NAME`	Indicates the name of the Storage Checkpoint.
`-i`	Runs the command in interactive mode where you must respond to prompts by the system. The default mode is non-interactive. (Optional)

Table B-1 dbed_clonedb command options *(continued)*

Option	Description
`-o umount`	Shuts down the clone database and unmounts the Storage Checkpoint file system.
`-o restartdb`	Mounts the Storage Checkpoint file system and starts the clone database. The `-o restartdb` option will not attempt to recover the clone database.
`-d`	Used with the `-o umount` option. If the `-d` option is specified, the Storage Checkpoint used to create the clone database will be removed along with the clone database.
`-p pfile_modification_file`	Specifies a file containing initialization parameters to be modified or added to the clone database's initialization parameter file prior to startup. The format is the same as the Oracle initialization parameter file.

To clone an Oracle instance with manual Oracle recovery

- Use the `dbed_clonedb` command as follows:

```
$ dbed_clonedb -S test2cln -m /tmp/test2cln \
-c Checkpoint_1239013340 -i
```

```
Primary Oracle SID is FLAS11r1
New Oracle SID is test2cln
Cloning an online Storage Checkpoint Checkpoint_1239013340.
Mounting Checkpoint_1239013340 at /tmp/test2cln.
All redo-log files found.
Creating inittest2cln.ora from /tmp/vxdba.rel.tmp.27511/initFLAS11r1.ora.
Altering db_name parameter in inittest2cln.ora.
Editing remote_login_passwordfile in inittest2cln.ora.
Altering control file locations in inittest2cln.ora.
Copying inittest2cln.ora to /oracle/11gr1/dbs/inittest2cln.ora.
About to start up new database and begin reconfiguration.
Database test2cln is being reconfigured.
Did not see error regarding database name.
Altering clone database archive log directory.
Updating log_archive_dest in clone database init file.
Found archive log destination at /snap_arch11r1/FLAS11r1.
The latest archive log(s) must now be applied. To apply
the logs, open a new window and perform the following steps:
```

```
1. You must copy required archive log(s) from primary to clone:
   Primary archive logs in /snap_arch11r1/FLAS11r1
   Clone archive logs expected in /tmp/test2cln/snap_arch11r1/FLAS11r1.
2. ORACLE_SID=test2cln; export ORACLE_SID # sh and ksh, OR
   setenv ORACLE_SID test2cln #csh
3. /oracle/11gr1/bin/sqlplus /nolog
4. CONNECT / AS SYSDBA
5. RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
6. enter the archive log(s) you would like to apply
7. EXIT

Press <Return> after you have completed the above steps
```

To clone an Oracle instance with automatic Oracle recovery

◆ Use the `dbed_clonedb` command as follows:

```
$ dbed_clonedb -S test1cln -m /tmp/test1cln \
-c Checkpoint_1239012493
```

```
Primary Oracle SID is FLAS11r1
New Oracle SID is test1cln
Cloning an online Storage Checkpoint Checkpoint_1239012493.
Mounting Checkpoint_1239012493 at /tmp/test1cln.
All redo-log files found.
Creating inittest1cln.ora from /tmp/vxdba.rel.tmp.25063/initFLAS11r1.ora.
Altering db_name parameter in inittest1cln.ora.
Editing remote_login_passwordfile in inittest1cln.ora.
Altering control file locations in inittest1cln.ora.
Copying inittest1cln.ora to /oracle/11gr1/dbs/inittest1cln.ora.
About to start up new database and begin reconfiguration.
Database test1cln is being reconfigured.
Did not see error regarding database name.
Starting automatic database recovery.
Shutting down clone database.
Altering clone database archive log directory.
Updating log_archive_dest in clone database init file.
Found archive log destination at /snap_arch11r1/FLAS11r1.
Mounting clone database.
Resetting logs on new database test1cln.
The sql script for adding tempfiles to test1cln is at /tmp/add_tf.test1cln.
Database instance test1cln is up and running.
```

To shut down the clone database and unmount the Storage Checkpoint

- Use the `dbed_clonedb`command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -o umount
```

To mount a Storage Checkpoint file system and start the clone database

- Use the `dbed_clonedb`command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -o restartdb

Database instance NEW9 is up and running.
```

To delete a clone database and the Storage Checkpoint used to create it

- Use the `dbed_clonedb`command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -o umount -d
```

Creating and working with snapplans using dbed_vmchecksnap

A snapplan specifies snapshot scenarios for a database (such as `online`, `instant`, or `offline`). You can name a snapplan file whatever you choose.

You can use the `dbed_vmchecksnap -o setdefaults` option to create the snapplan and set default values for the parameters. You may then modify the snapplan file using a text editor.

You can also use the command to validate, copy, list, or remove a snapplan and check the storage to make sure it is configured appropriately for the Database FlashSnap feature.

Snapplan parameters

When using `dbed_vmchecksnap -o setdefaults` option to create the snapplan, the following parameters are set:

Table B-2 Snapplan parameters

Parameter	Value
`SNAPSHOT_VERSION`	Specifies the snapshot version for this release of the product.

Table B-2 Snapplan parameters *(continued)*

Parameter	Value
PRIMARY_HOST	Specifies the name of the host where the primary database resides.
SECONDARY_HOST	Specifies the name of the host where the clone database will reside. If the primary and secondary hosts are the same, the snapshot volumes will not be deported. The primary and secondary host should have different host names if you are performing off host Flashsnap cloning.
PRIMARY_DG	Specifies the name of the Volume Manager disk group used by the primary database.
SNAPSHOT_DG_PREFIX	The prefix to be attached to the primary disk groups to create corresponding snapshot disk groups. Snapshot disk groups will take the name using a combination of SNAPSHOT_DG_PREFIX and the corresponding primary disk group name. It's default value will be SNAP_
ORACLE_SID	The name of the Oracle database.
ARCHIVELOG_DEST	Specifies the full path of the archive logs. There are several archive log destinations that can be used for database recovery if you are multiplexing the archive logs. You must specify which archive log destination to use. It is recommended that you have the archive log destination on a separate volume if SNAPSHOT_ARCHIVE_LOG is **yes** .

Table B-2 Snapplan parameters *(continued)*

Parameter	Value
`SNAPSHOT_ARCHIVE_LOG`	**yes** or **no** Specifies whether to create a snapshot of the archive log volumes. Specify **yes** to split the archive log volume mirrors and deport them to the secondary host. When using the Oracle remote archive log destination feature to send the archive logs to the secondary host, you can specify **no** to save some space. Because the archive logs may not always be delivered to the secondary host reliably, it is recommended that you specify .
`SNAPSHOT_MODE`	**online** or **offline** or **instant** Specifies whether the database snapshot should be online, offline, or instant. Only online snapshot mode is supported by Veritas Storage Foundation for Oracle RAC. If the snapshot is created while the database is online, the `dbed_vmsnap` command will put the tablespaces into backup mode. After `dbed_vmsnap` finishes creating the snapshot, it will take the tablespaces out of backup mode, switch the log files to ensure that the extra redo logs are archived, and create a snapshot of the archived logs. If the database if offline, it is not necessary to put the tablespaces into backup mode. If the snapshot mode is `instant`, a snapshot will be taken regardless of whether the database is online or offline. If it is online `dbed_vmsnap` will skip putting the tablespace into backup mode. **Note:** If `SNAPSHOT_MODE` is set to `offline` or `instant`, a two-host configuration is required and the `-r relocate_path` option is not allowed.
`SNAPSHOT_PLAN_FOR`	The default value is **database** and cannot be changed. Specifies the database object for which you want to create a snapshot.

Table B-2 Snapplan parameters *(continued)*

Parameter	Value
`SNAPSHOT_PLEX_TAG`	Specifies the name of the tag set to the plexes that will be used by `dbed_vmsnap` to take the snapshot. The `dbed_vmchecksnap` command will use this tag name to search if all the volumes in the database have the plexes with this tag name set. By default, `SNAPSHOT_PLEX_TAG=dbed_flashsnap`.
`SNAPSHOT_VOL_PREFIX`	Specifies the snapshot volume prefix. Use this variable to specify a prefix for the snapshot volumes split from the primary disk group. A volume name cannot be more than 32 characters.
`ALLOW_REVERSE_RESYNC`	**yes** or **no** By default, reverse resynchronization is off (set equal to `no`). If it is set to `yes`, this parameter allows you to restore the original volume from a snapshot. The original database, however, must be down for this operation. In a Veritas Storage Foundation for Oracle RAC environment, this parameter must be set to **no**.
`SNAPSHOT_MIRROR`	Specifies the number of plexes to be snapshot. The default value is 1.
DG:*VOL*	Optional These entries are created by default if the `dbed_vmchecksnap` command is run with the `-o setdefaults -m` option. These entries specify the mount path for the associated snapshot volumes when we perform a clone operation using dbed_vmclonedb. The values for these fields can be left blank, if they are blank then the `-r relocate_path` needs to be specified when performing a `dbed_vmclonedb` operation.

Creating a snapplan

Before creating a snapplan, the following conditions must be met:

Prerequisites	■ You must be the Oracle database administrator. ■ The disk group must be version 110 or later. For more information on disk group versions, see the `vxdg`(1M) manual page. ■ Be sure that a DCO and DCO volume are associated with the volume(s) for which you are creating the snapshot. ■ Snapshot plexes and their associated DCO logs should be on different disks than the original plexes, and should be configured correctly for creating snapshots by the system administrator. ■ Persistent FastResync must be enabled on the existing database volumes and disks must be assigned for the snapshot volumes. ■ The database must be running in archive log mode. Archive log mode is set in the Oracle initialization parameter file (`init.ora`). ■ `ORACLE_HOME` cannot reside on disk which will be used for snapshot.
Usage notes	■ In a Veritas Storage Foundation for Oracle RAC environment, the snapplan can be created on any node within the Oracle RAC cluster; the `-o validate` option can be run on any CVM node. ■ After creating the snapplan using the `dbed_vmchecksnap` command, you can use a text editor to review and update the file, if necessary. ■ It is recommended that you create a local working directory to store your snapplans in. This applies to single-instance Oracle only. ■ See the `dbed_vmchecksnap`(1M) online manual page for more information. ■ If the `SNAPSHOT_MODE` for the database is set to `online`, the primary and secondary hosts can be the same. If the `SNAPSHOT_MODE` is set to `offline` or `instant`, the primary and secondary hosts must be different. This applies to single instance Oracle only.

Table B-3 lists the options for creating a snapplan.

Table B-3 Options for creating a snapplan

Option	Description
`-S`	Specifies the `ORACLE_SID`, which is the name of the Oracle database instance, for which a snapshot image will be created. For Oracle RAC, specifies the ORACLE_SID, which is the name of the Oracle database instance on any CVM node.

Table B-3 Options for creating a snapplan *(continued)*

Option	Description
`-H`	Specifies the Oracle home directory that corresponds to the `ORACLE_SID`.
`-f SNAPPLAN`	Specifies the local path or the full path of the snapplan that you are creating.
`-o setdefaults`	Creates a default snapplan. This option can be used with the `-o validate` option to validate that the configuration is correct.
`-o validate`	Validates each parameter in the snapplan and checks whether the snapshot volumes have been configured correctly for creating snapshots, and copies the snapplan to the repository.
`-o list`	Lists all the snapplans associated with a specific $ORACLE_SID.
`-o copy`	Copies the snapplan from the repository to your current local directory.
`-o remove`	Removes the snapplan from the repository.
`-t SECONDARY_HOST`	Specifies the name of the host to which the snapshot image will be deported. If it is the same as the primary server, the snapshot volumes will not be deported. This argument is required if `-o setdefaults` is used. It is ignored if specified for `-o validate`.
`-p plex_tag`	Specifies the tag name for the plexes used to create the snapshot. This argument is required if `-o setdefaults` is used.
-m	Enables you to specify the mapped mountpoints for individual volumes for each disk group in the snapplan.

To create a snapplan and set the default values for a single host or an Oracle RAC cluster

- Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product \
-f snap1 -o setdefaults -t host1
```

```
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/prod_ar
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

To create a snapplan and set the default values in a two-host configuration

◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product -f snap2 -o setdefaults -t host2
```

```
Snapplan snap2 for PROD.
=====================================================
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/mytest/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Validating a snapplan

You can use the `dbed_vmchecksnap` command with the `-o validate` option to validate a snapplan and check the storage to make sure it is configured appropriately for the Database FlashSnap feature.

To validate a snapplan for a snapshot image to be used for single instance Oracle

- Use the `dbed_vmchecksnap` command as follows:

  ```
  $ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product \
  -f snap1 -o validate
  ```

```
PRIMARY_HOST is host1
SECONDARY_HOST is host1
The version of PRIMARY_DG-PRODdg is 110.
SNAPSHOT_DG is SNAP_PRODdg
SNAPSHOT_MODE is online
The database is running in archivelog mode.
ARCHIVELOG_DEST is /prod_ar
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/prod_ar is mount on /dev/vx/dsk/PRODdg/prod_ar.
Examining Oracle volume and disk layout for snapshot
Volume prod_db on PRODdg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODdg01.
Snapshot plex and DCO log for prod_db is on PRODdg02.
SNAP_PRODdg for snapshot will include: PRODdg02
ALLOW_REVERSE_RESYNC is yes
The snapplan snap1 has been created.
```

```
PRIMARY_HOST is host1
SECONDARY_HOST is host1
The version of PRIMARY_DG-PRODdg is 110.
SNAPSHOT_DG is SNAP_PRODdg
SNAPSHOT_MODE is online
The database is running in archivelog mode.
ARCHIVELOG_DEST is /prod_ar
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/prod_ar is mount on /dev/vx/dsk/PRODdg/prod_ar.
Examining Oracle volume and disk layout for snapshot
Volume prod_db on PRODdg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODdg01.
Snapshot plex and DCO log for prod_db is on PRODdg02.
SNAP_PRODdg for snapshot will include: PRODdg02
ALLOW_REVERSE_RESYNC is no
The snapplan snap1 has been created.
```

To validate a snapplan for a snapshot image to be used on the secondary host

- Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H \
/oracle/product -f snap2 -o validate
```

```
PRIMARY_HOST is host1
SECONDARY_HOST is host2
The version of PRIMARY_DG-PRODdg is 110.
SNAPSHOT_DG is SNAP_PRODdg
SNAPSHOT_MODE is online
The database is running in archivelog mode.
ARCHIVELOG_DEST is /mytest/arch
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/mytest/arch is mount on /dev/vx/dsk/PRODdg/rch.
Examining Oracle volume and disk layout for snapshot.
Volume prod_db on PRODdg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODdg01.
Snapshot plex and DCO log for prod_db is on PRODdg02.
SNAP_PRODdg for snapshot will include: PRODdg02
ALLOW_REVERSE_RESYNC is yes
The snapplan snap2 has been created.
```

Listing and viewing snapplans using dbed_vmchecksnap

The `dbed_vmchecksnap` command allows you to list and view existing snapplans.

To list all available snapplans for a specific Oracle database

◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -o list

The following snapplan(s) are available for PROD:

SNAP_PLAN      SNAP_STATUS      DB_STATUS     SNAP_READY

snap1          init_full        -             yes

snap2          init_full        -             yes

snap3          init_full        -             yes
```

The command output displays all available snapplans, their snapshot status (`SNAP_STATUS`), database status (`DB_STATUS`), and whether a snapshot may be taken (`SNAP_READY`).

To view a snapplan

◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o list
```

```
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/prod_ar
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=yes
SNAPSHOT_MIRROR=1

STORAGE_INFO
PRODdg02
SNAP_PLEX=prod_db-02 prod_ar-02
```

```
STATUS_INFO
SNAP_STATUS=init_full
```

Copying or removing a snapplan using dbed_vmchecksnap

The `dbed_vmchecksnap` command allows you to copy or remove snapplans.

To copy a snapplan from the repository to your local directory

- Assuming that the snapplan is not already present in your local directory, use the `dbed_vmchecksnap` command as follows:

  ```
  $ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o copy

  Copying 'snap1' to '/export/snap_dir'
  ```

To remove a snapplan

- Use the `dbed_vmchecksnap` command as follows:

  ```
  $ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o remove

  The snapplan snap1 has been removed from the repository.
  ```

Creating, resynchronizing, or reverse resynchronizing a snapshot database using dbed_vmsnap

You can use the `dbed_vmsnap` command to create a snapshot image of a database. The snapshot can be used locally or on another host that is physically attached to the shared storage. You can also resynchronize the snapshot image back to the primary database.

Before creating, resynchronizing, or reverse resynchronizing a snapshot database, review the following information:

Prerequisites	■ You must be logged in as the Oracle database administrator. ■ You must create and validate a snapplan using `dbed_vmchecksnap` before you can create a snapshot image with `dbed_vmsnap`.

Usage notes

- In an Oracle RAC environment, the `dbed_vmsnap` command can only be used on any CVM node.
- If possible, do not share volumes between Oracle database files and other software.
- When creating a snapshot volume, create the snapshot on a separate controller and on separate disks from the primary volume.
- Make sure your archive log destination is separate from your Oracle database volumes.
- Do not place any datafiles, including control files, in the `$ORACLE_HOME/dbs` directory.
- Resynchronization speed varies based on the amount of data changed in both the primary and secondary volumes when the mirror is broken off.
- See the `dbed_vmsnap`(1M) manual page for more information.

Options for the `dbed_vmsnap` command are:

Table B-4 dbed_vmsnap command options

Option	Description
`-S ORACLE_SID`	Specifies the `ORACLE_SID`, which is the name of the Oracle database instance, for which a snapshot image will be created. For Oracle RAC, specifies the ORACLE_SID, which is the name of the Oracle database instance on any CVM node.
`-f SNAPPLAN`	Specifies the name of the snapplan you are using.
`-o snapshot [-F] | resync`	Specifies whether to create a snapshot or synchronize the snapshot image with the current database image. The -F option prepares the volumes for being snapshot and forces snapshot creation.
`-o reverse_resync_begin`	Begins reverse resynchronization. Not supported by Veritas Storage Foundation for Oracle RAC.
`-o reverse_resync_commit`	Commits the reverse resynchronization changes after you have verified that they are acceptable. Not supported by Veritas Storage Foundation for Oracle RAC.

Table B-4 dbed_vmsnap command options *(continued)*

Option	Description
`-o reverse_resync_abort`	Aborts reverse resynchronization and mounts the original volumes back with the file systems that are configured to use the volume. Not supported by Veritas Storage Foundation for Oracle RAC.

To create a snapshot image on the primary host (single instance Oracle)

- Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot

dbed_vmsnap started at 2005-04-02 14:15:27
The database is running in archivelog mode.
A snapshot of ORACLE_SID PROD is in DG SNAP_PRODdg.
Snapplan snap1 is used for the snapshot.
Oracle Database server is orasvr.
If -r <relocate_path> is used in dbed_vmclonedb,make sure
<relocate_path> is created and owned by Oracle DBA. Otherwise,
the following mount points need to be created and owned by
Oracle DBA:

   /prod_db.
   /prod_ar.

dbed_vmsnap ended at 2004-04-02 14:16:11
```

To resynchronize a snapshot to your database

- Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o resync

dbed_vmsnap started at 2005-03-15 10:07:10

The option resync has been completed.

dbed_vmsnap ended at 2005-03-15 10:07:21
```

To resynchronize your database to a snapshot

◆ Assuming the mount point for the primary database was created and owned by the Oracle DBA user before mounting the VxFS file system, use the `dbed_vmsnap`command as follows:

```
$ dbed_vmsnap -S FLAS10r1 -f snaptst \
-o reverse_resync_begin


dbed_vmsnap started at 2009-07-30 14:06:38
SFORA dbed_vmsnap WARNING V-81-5725 After reverse_resync_commit is
performed, you need to recreate the Authentication Password
File using the ORAPWD utility.

Database FLAS10r1 (SID=FLAS10r1) is running.
The option reverse_resync_begin has been completed.
dbed_vmsnap ended at 2009-07-30 14:12:22dbed_vmsnap started at
2009-03-30 03:06:22
```

Mounting a storage checkpoint carried over from the volume snapshots is allowed only in a two-host configuration without the use of relocate path.

Storage checkpoints carried over from volume snapshots can be mounted before the clone database gets created (`dbed_vmclonedb -o mount`). Once the clone database is created (`dbed_vmclonedb -o recoverdb`), the checkpoints are no longer accessible since they are removed.

To abort resynchronizing your database to a snapshot

◆ Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o \
reverse_resync_abort

dbed_vmsnap started at 2004-04-02 16:16:44

The option reverse_resync_abort has been completed.

dbed_vmsnap ended at 2004-04-02 16:16:51
```

This option is only allowed when `reverse_resync_begin` has been run. It is not allowed if `reverse_resync_commit` has been executed.

To commit reverse resynchronization changes

- Use the `dbed_vmsnap` command as follows:

  Warning: Upon completion of reverse resynchronization, the content of the original database is discarded. Storage Checkpoints taken on either the original database or the clone database before or after the snapshot was created are discarded. Storage Checkpoints taken before the snapshot was created are preserved. The `dbed_vmsnap -o reverse_resync_commit` command cannot be undone and should be used with extreme caution.

```
$ /opt/VRTS/bin/dbed_vmsnap -S FLAS10r1 -f snaptst \
-o reverse_resync_commit

dbed_vmsnap started at 2009-07-30 14:14:47
Oracle instance FLAS10r1 successfully started.
The database is not running in archivelog mode.
SFORA dbed_vmsnap WARNING V-81-5725 After reverse_resync_commit is
performed, you need to recreate the Authentication Password
File using the ORAPWD utility.

The option reverse_resync_commit has been completed.
dbed_vmsnap ended at 2009-07-30 14:20:07
```

  This option is only allowed after `reverse_resync_begin` has been run.

Creating or shutting down a clone database using dbed_vmclonedb

You can use the `dbed_vmclonedb` command to create or shutdown a clone database on either the primary or secondary host using snapshot volumes from the primary host.

Before creating or shutting down a clone database, the following conditions must be met:

Prerequisites	■ You must be logged in as the Oracle database administrator to use `dbed_vmclonedb` command. ■ Before you can use the `dbed_vmclonedb` command, you must create and validate a snapplan and create a snapshot. ■ The volume snapshot must contain the entire database. ■ The system administrator must provide the database administrator with access to the necessary volumes and mount points. ■ Before you can use the `dbed_vmclonedb` command with the `-r relocate_path` option (which specifies the initial mount point for the snapshot image), the system administrator must create the mount point and then change the owner to the Oracle database administrator. ■ If `SNAPSHOT_MODE` is set to `offline` or `instant`, a two-host configuration is required and `-r relocate_path` is not allowed.
Usage notes	■ The `dbed_vmclonedb` command can be used on the secondary host. ■ If you are not using an off-host configuration, `-r relocate_path` is required. ■ In a two-host configuration, the `server_name=`*`svr_name`* option is required. ■ Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply. ■ See the `dbed_vmclonedb`(1M) manual page for more information.

Options for `dbed_vmclonedb` are:

Table B-5 dbed_vmclonedb options

Option	Description
`-S ORACLE_SID`	Specifies the `ORACLE_SID`, which is the name of the Oracle database instance, for which a snapshot image will be created. For Oracle RAC, specifies the ORACLE_SID, which is the name of the Oracle database instance on any CVM node.
`-o mount`	Mounts the file systems so you can use them to do a backup.
`-o mountdb`	Starts the database to allow manual database recovery.

Table B-5 dbed_vmclonedb options *(continued)*

Option	Description
`-o recoverdb`	Automatically recovers the database.
`-o restartdb`	Restarts the database if the clone database is shut down. A clone database must exist to use the `-o restartdb` option.
`-o update_status`	Updates the database status information in the repository.
`-o umount`	Shuts down the clone database and unmounts all snapshot files.
`new_sid=`*`new_sid`*	Specifies the new `ORACLE_SID` for the snapshot image. This is a required argument.
`server_name=`	Specifies the host on which the primary Oracle instance runs.
`-f SNAPPLAN`	Indicates the name of the snapplan that you are using.
`-H ORACLE_HOME`	Specifies the Oracle home directory that corresponds to the `ORACLE_SID`.
`-p` *`pfile_modification_file`*	Specifies a file containing initialization parameters to be modified or added to the clone database's initialization parameter file prior to startup. The format is the same as the Oracle initialization parameter file.
`-r relocate_path`	Specifies the initial mount point for the snapshot image. If you are using an off-host configuration, -r is required. Otherwise, it is an optional argument. If `-r relocate_path` is used with the `-o mount | mountdb | reccoverdb` options, it will also be required to restart or unmount the clone database. **Note:** Do not use `-r relocate_path` if the `SNAPSHOT_MODE` parameter is set to instant or offline.

To clone the primary database automatically in a single-host configuration or a RAC cluster

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
-o recoverdb,new_sid=NEWPROD,server_name=orasvr -f snap1 -r /clone

dbed_vmclonedb started at 2009-05-26 09:29:39

Editing remote_login_passwordfile in initclone1.ora.

All redo-log files found.

Altering instance_name parameter in initclone1.ora.

Altering instance_number parameter in initclone1.ora.

Altering thread parameter in initclone1.ora.

Database CLONE1 (SID=clone1) is running.

dbed_vmclonedb ended at 2009-05-26 09:32:16
```

To clone the primary database on a secondary host automatically in a two-host configuration

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
-o recoverdb,new_sid=NEWPROD,server_name=orasvr -f snap2

dbed_vmclonedb started at 2009-05-26 09:29:39

Editing remote_login_passwordfile in initclone1.ora.
All redo-log files found.

Altering instance_name parameter in initclone1.ora.

Altering instance_number parameter in initclone1.ora.

Altering thread parameter in initclone1.ora.

Database CLONE1 (SID=clone1) is running.
dbed_vmclonedb ended at 2009-05-26 09:32:16
```

To clone the primary database manually in a single-host configuration or a RAC cluster

1. Mount the file systems.
2. Create a clone using the `dbed_vmclonedb` command.

   ```
   $ dbed_vmclonedb -S FLAS11r1 -o mountdb \
   new_sid=test5cln,server_name=slias19 -f snap1 -r /tmp/test5cln

   dbed_vmclonedb started at 2009-04-07 18:16:01
   Editing remote_login_passwordfile in inittest5cln.ora.
   All redo-log files found.
   Altering instance_name parameter in inittest5cln.ora.
   Altering instance_number parameter in inittest5cln.ora.
   Altering thread parameter in inittest5cln.ora.
   Database TEST5CLN (SID=test5cln) is in recovery mode.

   If the database test5cln is recovered manually, you must run
   dbed_vmclonedb -o update_status to change the snapshot status.
   dbed_vmclonedb ended at 2009-04-07 18:17:58
   ```

3. Follow the Oracle recovery procedure to recover the database manually.
4. Update the snapshot status (`database_recovered`) for the clone database on the primary host after manual recovery has been completed.

   ```
   $ /opt/VRTS/bin/dbed_vmclonedb -o
   update_status,new_sid=NEWPROD,server_name=orasvr -f snap1 -r /clone

   dbed_vmclonedb started at 2004-04-02 15:19:16

   The snapshot status has been updated.

   dbed_vmclonedb ended at 2004-04-02 15:19:42
   ```

To clone the primary database manually in a two-host configuration

1. Mount the file systems.
2. Create a clone using the `dbed_vmclonedb` command.

   ```
   $ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
   -o mountdb,new_sid=NEWPROD,server_name=orasvr -f snap2

   dbed_vmclonedb started at 2009-05-27 23:17:09

   Editing remote_login_passwordfile in initclone1.ora.

   All redo-log files found.

   Altering instance_name parameter in initclone1.ora.

   Altering instance_number parameter in initclone1.ora.

   Altering thread parameter in initclone1.ora.

   Database CLONE1 (SID=clone1) is in recovery mode.

   If the database clone1 is recovered manually, you must run
   dbed_vmclonedb -o update_status to change the snapshot status.
   dbed_vmclonedb ended at 2009-05-27 23:18:50
   ```

3. Follow the Oracle recovery procedure to recover the database manually.
4. Update the snapshot status (`database_recovered`) for the clone database on the secondary host after manual recovery has been completed.

   ```
   $ /opt/VRTS/bin/dbed_vmclonedb -o \
   update_status,new_sid=NEWPROD,server_name=orasvr -f snap2

   dbed_vmclonedb started at 2004-04-06 09:22:27

   The snapshot status has been updated.

   dbed_vmclonedb ended at 2004-04-06 09:22:40
   ```

To shut down the clone database and unmount all snapshot file systems if you are not using an off-host configuration

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD,\
server_name=orasvr -f snap1 -r /clone

dbed_vmclonedb started at 2004-04-02 15:11:22

Umounting /clone/prod_db.

Umounting /clone/prod_ar.

dbed_vmclonedb ended at 2004-04-02 15:11:47
```

To shut down the clone database and unmount all snapshot file systems in an off-host configuration

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD,\
server_name=orasvr -f snap2

dbed_vmclonedb started at 2004-04-09 23:09:21

Umounting /clone/arch.

Umounting /clone/prod_db.

dbed_vmclonedb ended at 2004-04-09 23:09:50
```

This shuts down the clone database, unmounts file systems, and deports the snapshot disk group for a clone on a secondary host.

To restart a clone database if you are not using an off-host configuration

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
-o restartdb,new_sid=NEWPROD,server_name=orasvr -f snap1 -r /clone

dbed_vmclonedb started at 2004-04-02 15:14:49

Oracle instance NEWPROD successfully started.

dbed_vmclonedb ended at 2004-04-02 15:15:19
```

To restart a clone database in an off-host configuration

- Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD \
-o restartdb,new_sid=NEWPROD,server_name=orasvr -f snap2

dbed_vmclonedb started at 2003-04-09 23:03:40

Oracle instance NEWPROD successfully started.

dbed_vmclonedb ended at 2003-04-09 23:04:50
```

Managing log files using edgetmsg2

You can use the `edgetmsg2` utility to manage message log files. You can use the `edgetmsg2` utility to write a message to a log file or to the console, read the log file and print to the console, and display the available log files.

Before managing log files with the `edgetmsg2` command, review the following information:

Prerequisites	■ You must be logged in as the Database Administrator or root to use this command.
Usage notes	■ The default log file for a database is located in the following directory: `/etc/vx/vxdbed/logs/sfua_`*`database`*`.log` where *database* is the `ORACLE_SID`. ■ By default, only messages with a severity equal to or greater than `ERROR` will be logged. ■ See the `edgetmsg2`(1M) manual page for more information.

Table B-6 lists options for `edgetmsg2`.

Table B-6 edgetmsg2 options

Option	Description
`-s set_num`	Specifies the message catalogue set number. The default is 1.
`-M msgid[:severity]`	Specifies the message ID and severity to be printed.

Table B-6 edgetmsg2 options *(continued)*

Option	Description
`-f msg_catalog | logfile | log_directory`	Specifies the message catalogue path, log file, or log directory.
`-v severity | severity`	Overwrites the minimum log severity or creates a severity filter. The severity values are either 0-8 or 100-108.
`-p`	Pauses the cursor at the end of a display message. By default, a line feed is added to each display message. Use the `-p` option to indicate that no line feed is to be added.
`-o list [,suppress_time]`	Displays the content of a log file. You can specify `,suppress_time` to exclude time information in the utility output.
`-o report[,no_archive]`	Displays the available log files. You can specify ,no_archive to exclude log files from the utility output.
`-t from_time[,to_time]`	Reduces the length of the utility output by specifying the time range to include. This option must be used together with the `-o list` option. Use the following format: `yyy-mm-dd HH:MM:SS`.
`-S ORACLE_SID`	Specifies the ORACLE_SID for an Oracle database.
`"default format string"`	Specifies the C language `printf()` format string.
`[args]`	Specifies arguments for the format string conversion characters.

To print a message

◆ Use the `edgetmsg2` command as follows:

```
$ /opt/VRTS/bin/edgetmsg2 [-s set_num] \

[-M msgid[:severity]] \
[-f msg_catalog] [-v severity] [-p] [-m value] \
["default format string" [args]]
```

To read a message log file

- Use the `edgetmsg2` command as follows:

```
$ /opt/VRTS/bin/edgetmsg2 -o list[,suppress_time] \
-S ORACLE_SID | [-f logfile] \
[-v severity] [-t from_time,to_time]
```

To list available log files

- Use the `edgetmsg2` command as follows:

```
$ /opt/VRTS/bin/edgetmsg2 -o report[,no_archive] \
[-f log_directory]
```

Identifying VxFS files to convert to Quick I/O using qio_getdbfiles

You can use the `qio_getdbfiles` command to identify VxFS files before converting them to Quick I/O files. Only VxFS files may be converted to Quick I/O.

The `qio_getdbfiles` command queries the database and gathers a list of datafiles to be converted to Quick I/O. The command requires direct access to the database.

Before using the `qio_getdbfiles` command, the following conditions must be met:

Prerequisites	■ To use this command for Oracle, the `ORACLE_SID` environment variable must be set. ■ You must be logged in as the database administrator.
Usage notes	■ The `-T` option forces the behavior for a specific database type. The database options that are supported are `ora`, `syb`, and `db2`. Use this option in environments with more than one type of database. ■ The `-a` option specifies that all datafiles should be included. By default, potential sparse files are excluded. ■ See the `qio_getdbfiles`(1M) manual page for more information. ■ See the `qio_getdbfiles`(1M) manual page for more information.

To identify the VxFS files to convert to Quick I/O

1. Use the `qio_getdbfiles` command as follows:

   ```
   $ /opt/VRTS/bin/qio_getdbfiles [-T ora|syb|db2]
   ```

   ```
   $ /opt/VRTSsybed/bin/qio_getdbfiles [-T syb] \
   [-d <database_name>] [-m <master_device_pathname>]
   ```

   where `-T syb` forces behavior for Sybase, <`database_name`> specifies the database device files, and <`master_device_pathname`> specifies the full path name of the master device for the Sybase ASE server.

   The `qio_getdbfiles` command stores the filenames and file sizes in bytes in a file called `mkqio.dat`.

2. View the `mkqio.dat` file:

   ```
   $ cat mkqio.dat
   ```

   The `mkqio.dat` file contains the database filenames that can be converted to Quick I/O files. The format of the file is a list of paired file paths and file sizes. For example:

   ```
   /database/dbfiles.001 1024000
   ```

   ```
   /database/dbfiles.002 2048000
   ```

Converting VxFS files to Quick I/O using qio_convertdbfiles

After running `qio_getdbfiles`, you can use the `qio_convertdbfiles` command to convert database files to use Quick I/O. This command is for use with VxFS file systems only.

The `qio_convertdbfiles` command converts regular files or symbolic links that point to regular files on VxFS file systems to Quick I/O. The `qio_convertdbfiles` command converts only those files listed in the `mkqio.dat` file to Quick I/O. The `mkqio.dat` file is created by running `qio_getdbfiles`. It can also be created manually.

Before converting files, the following conditions must be met:

Prerequisites	■ To use this command for Oracle, the `ORACLE_SID` environment variable must be set. ■ You must be logged in as the database administrator. ■ Remove any non-VxFS files from mkqio.dat before running `qio_convertdbfiles`. The `qio_convertdbfiles` command will display an error message if any of the database files in mkqio.dat are not on a VxFS file system.
Usage notes	■ The `qio_convertdbfiles` command expects all files to be owned by the database administrator. ■ Converting existing database files to Quick I/O is not recommended if the files are fragmented. In this case, it is recommended that you create new files with the `qiomkfile` command (these files are guaranteed not to be fragmented) and then convert the data from the old files (using a command such as `dd`). ■ Ensure that the database is shut down before running `qio_convertdbfiles`. ■ See the `qio_convertdbfiles`(1M) manual page for more information.

Table B-7 lists options for the `qio_convertdbfiles` command.

Table B-7 qio_convertdbfiles command options

Option	Description
`-T`	Forces the behavior for a specific database type. The database options that are supported are `ora`, `syb`, and `db2`. Use this option in environments with more than one type of database.
`-a`	Changes regular files to Quick I/O files using absolute pathnames. Use this option when symbolic links need to point to absolute pathnames. By default, relative pathnames are used.
`-f`	Reports on current fragmentation levels for files listed in `mkqio.dat`. Fragmentation is reported at four levels: not fragmented, slightly fragmented, fragmented, and highly fragmented.
`-h`	Displays a help message.
`-i`	Creates extra links for all database files and log files in the /dev directory to support the SAP `brbackup` command.
`-u`	Changes Quick I/O files back to regular files.

To convert VxFS files to Quick I/O files

1 After running the `qio_getdbfiles` command, shut down the database:

Warning: Running `qio_convertdbfiles` with any option except `-f` while the database is up and running can cause severe problems for your database, including data loss and corruption. Make sure the database is shut down before running the `qio_convertdbfiles` command.

---

2 Run the `qio_convertdbfiles` command to convert the list of files in `mkqio.dat` to Quick I/O files:

```
$ /opt/VRTS/bin/qio_convertdbfiles
```

You must remove any non-VxFS files from `mkqio.dat` before running `qio_convertdbfiles`. The `qio_convertdbfiles` command will display an error message if any of the database files in `mkqio.dat` are not on a VxFS file system.

3 Restart the database to access these database files using the Quick I/O interface.

To undo a previous run of qio_convertdbfiles

◆ Use the `qio_convertdbfiles` as follows:

```
$ /opt/VRTS/bin/qio_convertdbfiles -u
```

```
.dbfile::cdev:vxfs: --> dbfile
```

This reverts a previous run of `qio_convertdbfiles` and changes Quick I/O files back to regular VxFS files.

If the database is up and running, an error message will be displayed stating that you need to shut it down before you can run `qio_convertdbfiles`.

Recreating Quick I/O files using qio_recreate

You can use the `qio_recreate` command to automatically recreate Quick I/O files when the database is recovered.

Before converting files to Quick I/O, the following conditions must be met:

Prerequisites

- To use this command for Oracle, the `ORACLE_SID` environment variable must be set.
- You may be logged in as either the database administrator or root.

Usage notes

- The command expects to find a file named in the directory where the command is run. The mkqio.dat file contains a list of the Quick I/O files used by the database and their sizes. If the file is not in the directory, you will be prompted to create it using .
  See “Identifying VxFS files to convert to Quick I/O using qio_getdbfiles” on page 319.
- The `qio_recreate` command supports conventional Quick I/O files only (that is, Quick I/O files in the following form: `file` --> `.file::cdev:vxfs:`). In creating a Quick I/O file, the `qio_convertdbfiles` command renames the regular VxFS file, `file`, to `.file` with the Quick I/O extension (`:cdev::vxfs:`) and creates a symbolic link to it. By default, the symbolic link uses a relative path name.
- There are no options for the `qio_recreate` command and no output is returned when the command runs successfully.
- See the `qio_recreate`(1M) manual page for more information.

The qio_recreate command follows these rules in recreating Quick I/O files when a database is recovered:

- If a Quick I/O file (`.file::cdev:vxfs:`) is missing, then `qio_recreate` recreates it.
- If both a symbolic link (`file`) and its associated Quick I/O file (`.file::cdev:vxfs:`) are missing, `qio_recreate` recreates both the symbolic link and the Quick I/O file.
- If a symbolic link (`file`) from a regular VxFS file to its associated Quick I/O file (`.file::cdev:vxfs:`) is missing, then `qio_recreate` recreates the symbolic link.
- If a Quick I/O file (`.file::cdev:vxfs:`) is missing and the regular VxFS file that is symbolically linked to it is not the same one that originally created it, then `qio_recreate` issues a warning message and does not recreate the Quick I/O file.
- If a Quick I/O file (`.file::cdev: vxfs:`) is smaller than the size listed in `mkqio.dat`, `qio_recreate` issues a warning message.

To automatically recreate Quick I/O files when the database is recovered

◆ Use the `qio_recreate` command as follows:

```
$ /opt/VRTSdbed/bin/qio_recreate -T ora
```

Defining database parameters for Database Dynamic Storage Tiering using dbdst_admin

Running the `dbdst_admin` command defines parameters for the entire database. You must run this command at least once to define the database parameters for Database Dynamic Storage Tiering. Three pre-defined storage classes will be created (PRIMARY, SECONDARY, and BALANCE). Parameter values are stored in the SFDB repository.

Set at least one of the parameters in `maxclass`, `minclass`, `statinterval`, `sweeptime`, `sweepinterval`, `purgetime`, or `purgeinterval`, to enable default values. Add at least one class to enable the default classes.

Before defining your database parameteres using the `dbdst_admin` command, review the following information:

Prerequisites	■ An Oracle database must be up and running. ■ Run the `dbed_update` command before running any of the Database Dynamic Storage Tiering commands. You should also run the `dbed_update` command if any of the database files change. Because the Database Dynamic Storage Tiering commands retrieve database information from the repository, the repository must be up to date. ■ Change the owner of the mount point on which you want to implement Database Dynamic Storage Tiering to oracle.

Usage notes	■ Only the Oracle database administrator can run Database Dynamic Storage Tiering commands.
	■ If you are using any of the Database Dynamic Storage Tiering commands in a high availability (HA) environment, the time on each system in the cluster must be synchronized.
	■ Create the volumes that you want to add to the multi-volume file system in the same disk group as the file system volume. As root, use the following command to change the owner of each volume:
	`# /opt/VRTS/bin/vxedit -g disk_group \`
	`set user=oracle volume`

Define the `LD_LIBRARY_PATH` environment variable as follows:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/VRTSdbed/common/lib; \
export LD_LIBRARY_PATH
```

Table B-8 lists the options for the Oracle `dbdst_admin` command.

Table B-8 Oracle dbdst_admin command options

Option	Description
`-S $ORACLE_SID`	Specifies the `ORACLE_SID`, which is the name of the Oracle instance.
`list`	Lists all the Database Dynamic Storage Tiering parameters of the database, including class name and description. This option should be used exclusively from the other options.
`maxclass=`	Maximum number of storage classes allowed in the database. Default value is 4.
`minclass=`	Minimum number of storage classes allowed in the database. Default value is 2.
`sweepinterval=`	Interval for file sweeping for file relocation. Default value is 1, which means one per day. If this value is set to 0, all scheduled sweep tasks will become unscheduled.
`sweeptime=`	Time per day for the file sweep to take place. Times are entered in 24-hour periods and should list hour: minute. For example, 8:30 AM is represented as 08:30 and 10:00 PM is represented as 22:00. Default value is 22:00.
`statinterval=`	Interval in minutes for gathering file statistics. Default value is 30, which represents every 30 minutes. If this value is set to 0, all scheduled tasks will become unscheduled.
`purgeinterval=`	Number of days after which the file statistics in the repository will be summarized and purged. Default value is 30. It is recommended that you set your purge interval sooner because you will not be able to view any statistics until the first 30-day interval is over, if you use the default.

Table B-8 Oracle dbdst_admin command options *(continued)*

Option	Description
`purgetime=`	Time per day for the file purge to take place. Times are entered in 24-hour periods and should list hour: minute. For example, 8:30 AM is represented as 08:30 and 8:00 PM is represented as 20:00. Default value is 20:00.
`addclass=`	Parameter that allows you to add a class to a database. The information should be entered as *class:"description"*, where *class* represents the class name and *description* is a string of up to 64 characters enclosed by double quotes used to describe the class.
`rmclass=`	Parameter that allows you to remove a class from a database. Enter the class name as it appears in the database.
`definechunk=`	Defines a chunksize in bytes for a specific storage class to extent balance the files in this storage class. The information should be entered as *classname: valid chunksize value*. Valid chunksizes are 128k, 256k, 512k or 1m bytes.

To define database parameters

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S ORACLE_SID-o setup-parameters,\
[storage_class operations]
 setup-parameters
      maxclass=number,minclass=number,statinterval=minutes,\
      sweeptime=HH:MM,sweepinterval=days
      purgetime=HH:MM,purgeinterval=days
 storage_class operations
      addclass=classname:"description", rmclass=classname
      definechunk=classname:128k | 256k | 512k | 1m
```

For example, to add a class called tier1 for database PROD, and to set up a purge interval of one, meaning that the file statistics will be gathered for one day and then summarized and purged, use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=tier1:"Fast Storage",\
purgeinterval=1
```

Setting up storage classes for Database Dynamic Storage Tiering using dbdst_admin

The `dbdst_admin` command allows you to add, remove, or display storage classes required for Database Dynamic Storage Tiering.

To add a storage class

- Use the `dbdst_admin` command as follows:

  ```
  $ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=class:\
  "description"
  ```

For example, to create a storage class named "FAST" for an EMC array, use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=FAST:\
"fast EMC array"
```

To remove a storage class

- Use the `dbdst_admin` command as follows:

  ```
  $dbdst_admin -S FLAS11r1 -o rmclass=class
  ```

For example, to remove a storage class called "SLOW," use the `dbdst_admin` command as follows:

```
$dbdst_admin -S FLAS11r1 -o rmclass=SLOW
```

To display existing storage classes and properties

◆ Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o list
```

For example

```
$ /opt/VRTS/bin/dbdst_admin -S FLAS10r1 -o list

TSDB Parameters for FLAS10r1
-----------------------------------
Maxclass        =  4
Minclass        =  2
Stat-Interval   =  30
Sweep Information = 22HH:0MM, Frequency = 1
Purge Information = 20HH:0MM, Frequency = 30


TSDB Storage Classes for FLAS10r1
-----------------------------------
Name = PRIMARY : Description = PRIMARY Storage Class
Name = SECONDARY : Description = SECONDARY Storage Class
Name = BALANCE : Description = BALANCE Storage Class
```

Converting a VxFS file system to a VxFS multi-volume file system for Database Dynamic Storage Tiering using dbdst_convert

To convert your existing VxFS file system to a VxFS multi-volume file system, you must convert a single volume to a volume set.

When you convert to a volume set using the `dbdst_convert` command, the original volume will be renamed to a new volume name. The mount device name will become the new volume set name. Creating the new volume set name with the mount device name nullifies the need to rename the mount device in various locations.

Before converting to a volume set, make sure the following conditions have been met:

Prerequisites	■ The database must be shutdown. ■ Create at least one additional volume.

Usage notes	■ You must convert the single-volume file system on which you plan to implement Database Dynamic Storage Tiering. ■ The file system can be mounted or unmounted when you run the `dbdst_convert` command. ■ If the file system has n volumes, volumes 1 through n–1 will be placed in the storage class "PRIMARY" and volume n will be placed in the storage class "SECONDARY." ■ The volumes specified when running the conversion must be in the same disk group as the mount device.

To convert a mount device from a single volume device to a volume set

1. Use the `dbdst_convert` command as follows:

   ```
   $ /opt/VRTS/bin/dbdst_convert -S $ORACLE_SID -M mount_device -v \
   volume_name,volume_name
   ```

2. Bring database objects online.

For example, to convert a volume-based oradata file system to a Database Dynamic Storage Tiering-ready volume set file system on mount device /dev/vx/dsk/oradg/oradata, use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD -M /dev/vx/dsk/oradg/oradata -v \
new_vol1,new_vol2
```

After conversion, you will have a volume set named oradata containing three volumes (oradata_b4vset, new_vol1, and new_vol2). The file system will have two storage classes defined as PRIMARY and SECONDARY. The volumes will be assigned as follows:

- PRIMARY storage class will contain volumes oradata_b4vset and new_vol1.
- SECONDARY storage class will contain volume new_vol2.

Classifying volumes into a storage class for Database Dynamic Storage Tiering using dbdst_classify

Before creating a DST policy or manually moving data, assign classes to your volumes.

Before assigning classes to volumes, review the following information:

Usage notes

- You must convert your VxFS file system to a multi-volume file system first.
- Storage classes must be registered using the `dbdst_admin` command before assigning classes to volumes.
- The database can be online or offline.

To classify a volume

◆ Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M mount_device \
-v volume_name:class[,volume_name:class]
```

For example, to assign the class "FAST" to volume new_vol1, use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M /dev/vx/dsk/oradg/oradata \
-v new_vol1:FAST
```

Displaying free space on storage classes for Database Dynamic Storage Tiering using dbdst_show_fs

To see the free space, class information, and volume information on your storage classes, use the `dbdst_show_fs` command.

Table B-9 shows the Oracle `dbdst_show_fs` command options.

Table B-9 Oracle dbdst_show_fs command options

Option	Description
`-S $ORACLE_SID`	Specifies the `ORACLE_SID`, which is the name of the Oracle instance.
`-o volume`	Displays the free space on volumes in each class.
`-m`	Specifies the mount point.

Before displaying the free space on a storage class, review the following information:

Prerequisites

- Make sure the file system is mounted.

Usage notes

- See the `dbdst_show_fs`(1M) manual page.

To display the free space on a storage class

◆ Use the `dbdst_show_fs` command as follows:

```
$ /opt/VRTS/bin/dbdst_show_fs -S $ORACLE_SID -o volume \
-m mount_point
```

```
$ /opt/VRTS/bin/dbdst_show_fs -S FLAS10r2  -m /snap_data10r2
```

```
MAXCLASS=4
MINCLASS=2
TOTAL CLASS=3

VOLUME NAME          SIZE    USED     AVAILABLE
-----------          ----    ----     ---------

CLASS=FAST
datavol1             1024000 20692    1003308

CLASS=MEDIUM
datavol2             4608000 221075   4386925

CLASS=PRIMARY
datavol-b4vset       3072000 2765061  306939
```

Adding new volumes to a storage class for Database Dynamic Storage Tiering using dbdst_addvol

Use the `dbdst_addvol` command to add volumes to a volume set.

Before adding a volume, review the following information:

Usage notes	■ The database must be inactive when adding volumes to a storage class.

To add a volume to a volume set

◆ Use the `dbdst_addvol` command as follows:

For Oracle:

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID -M mount_device \
-v volume_name:class[,volume_name:class]
```

Removing volumes from a storage class for Database Dynamic Storage Tiering using dbdst_rmvol

You may need to remove a volume from a volume set. To remove a volume, use the `dbdst_rmvol` command.

Before removing a volume, review the following information:

Usage notes	■ The database must be inactive when removing volumes from a storage class. ■ Only a volume that does not contain any file system data can be removed

To remove a volume from a volume set

◆ Use the `dbdst_rmvol` command as follows:

For Oracle:

```
$ /opt/VRTS/bin/dbdst_rmvol -S $ORACLE_SID -M mount_device \
-v volume_name[,volume_name]
```

Relocating files for Database Dynamic Storage Tiering using dbdst_file_move

For Oracle:

Use the `dbdst_file_move` command to relocate flashback logs, archive logs, datafiles, and external files if the files are no longer being used frequently.

Table B-10 shows the Oracle `dbdst_file_move` command options.

Table B-10 Oracle dbdst_file_move command options

Option	Description
`-o archive[n] | flashback`	Specifies which archive logs or Flashback logs to move. Do not use this option with the `-f` option. Flashback is supported by Oracle 10g or later.
`-o external | datafile`	Specifies whether to move external files or datafiles. Use this option with the `-f` option.
`-f listfile`	Specifies a listfile that contains a list of files or directories to be moved.

Table B-10 Oracle dbdst_file_move command options *(continued)*

Option	Description
`-c` *`class`*`[:days]`	Specifies the storage class to which the files should be moved. If the *days* option is used, the files will be moved to the class specified if they have not been accessed in the number of days specified. Do not specify *days* if you are using the `-o datafile` option.
`-R`	Removes the policy for the specified object.

Before relocating a file, review the following information:

Usage notes

- Multiple partitions cannot reside on the same tablespace.

To relocate a file

◆ Use the `dbdst_file_move` command as follows:

```
/opt/VRTS/bin/dbdst_file_move -S $ORACLE_SID -o datafile \
-f listfile -c storage_class
```

For example:

```
$ /opt/VRTS/bin/dbdst_file_move -S FLAS10r2  -o datafile -f /home/extern
-c MEDIUM

Assign placement policy file successful on filesystem /snap_data10r2
Begin enforcing filesystem /snap_data10r2 ...
Enforce placement policy successful on filesystem /snap_data10r2
```

Relocating tablespaces for Database Dynamic Storage Tiering using dbdst_tbs_move

Use the `dbdst_tbs_move` command to move tablespaces to the desired storage class. The command queries the SFDB repository for the tablespace file names, then performs a one-time move based on your immediate requirements.

To relocate a tablespace

◆ Use the `dbdst_tbs_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_tbs_move -S $ORACLE_SID -t tablespace \
-c class
```

where

- *tablespace* indicates which tablespace to move.
- *class* indicates to which class the tablespace should be moved.

For example:

```
$ /opt/VRTS/bin/dbdst_tbs_move -S FLAS10r2  -t MYTBS  \
-c MEDIUM
```

```
Assign placement policy file successful on filesystem /snap_data10r2
Begin enforcing filesystem /snap_data10r2 ...
Enforce placement policy successful on filesystem /snap_data10r2
```

Appendix C

Database FlashSnap status information

This appendix includes the following topics:

- About Database FlashSnap status information
- Database FlashSnap Snapshot status information from the CLI

About Database FlashSnap status information

Veritas Database FlashSnap functionality provides the following information for the various snapplan stages and snapshot procedures:

- Snapshot status information
- Snapshot database status information

You can view this information using either the command line interface (CLI) or the GUI.

For additional information about Database FlashSnap GUI functionality, see the *Veritas Storage Foundation for Database Graphical User Interface Guide*

Database FlashSnap Snapshot status information from the CLI

To view snapshot status information from the command line interface (CLI), use the `dbed_vmchecksnap` command with the `-o list` option to list all available snapplans for a specified database. Snapshot status information is displayed in the command output under the column heading `SNAP_STATUS`.

Note: The snapshot status and snapshot database status information may also appear in error messages.

Snapshot status information from the CLI

Table C-1 shows detailed information about each snapshot status (SNAP_STATUS) value.

Note: SF Oracle RAC does not support Database FlashSnap reverse resynchronization.

Table C-1 Snapshot status information from the CLI

SNAP_STATUS	Completed operations	Allowed operations
`init_full`	■ `dbed_vmchecksnap -o validate` (successful) ■ `dbed_vmsnap -o resync` (successful)	`dbed_vmsnap -o snapshot`
`init_db`	`dbed_vmchecksnap -o validate -f` *`snapplan`* (failed)	Ensure that your storage configuration has been set up correctly.
`snapshot_start`	`dbed_vmsnap -o snapshot` (failed)	Contact your system administrator for help. Use Veritas Volume Manager commands to resynchronize the snapshot volumes, and use `dbed_vmsnap -o snapshot -F` to force snapshot creation.
`snapshot_end`	■ `dbed_vmsnap -o snapshot` (successful)	■ `dbed_vmsnap -o resync` ■ `dbed_vmclonedb -o mount|mountdb|recoverdb`

`snapshot_vol_start`
`snapshot_vol_end`
`resync_dg_start`
`resync_dg_end`
	`dbed_vmsnap -o snapshot` (failed)	Re-run `dbed_vmsnap -o snapshot`

Table C-1 Snapshot status information from the CLI *(continued)*

SNAP_STATUS	Completed operations	Allowed operations

`resync_vol_start`
`resync_vol_end`
`snapshot_dg_start`
`snapshot_dg_end`
	`dbed_vmsnap -o resync` (failed)	Re-run `dbed_vmsnap -o resync`

`resync_start`	`dbed_vmsnap -o resync` (failed)	Contact your system administrator for help. Use Veritas Volume Manager commands to resynchronize the snapshot volumes, and use `dbed_vmsnap -o snapshot -F` to force snapshot creation.
`mount start`	`dbed_vmclonedb -o mount` (failed)	`dbed_vmclonedb -o -umount`
`mount_end`	`dbed_vmclonedb -o mount` (successful)	`dbed_vmclonedb -o umount`
`restartdb_start`	`dbed_vmclonedb -o restartdb` (failed)	■ `dbed_vmclonedb -o umount` ■ Start the snapshot database manually.
`restartdb_end`	`dbed_vmclonedb -o restartdb` (successful)	`dbed_vmclonedb -o umount`
`mountdb_start`	`dbed_vmclonedb -o mountdb` (failed)	Recover the snapshot database manually, then run `dbed_vmclonedb -o update_status`
`mountdb_end`	`dbed_vmclonedb -o mountdb` (successful)	■ `dbed_vmclonedb -o update_status` ■ `dbed_vmclonedb -o umount`

Table C-1 Snapshot status information from the CLI *(continued)*

SNAP_STATUS	Completed operations	Allowed operations
recoverdb_start	dbed_vmclonedb -o recoverdb (failed)	■ Recover the snapshot database manually, then run dbed_vmclonedb -o update_status ■ dbed_vmclonedb -o umount
recoverdb_end	dbed_vmclonedb -o recoverdb (successful)	dbed_vmclonedb -o umount
umount_start	dbed_vmclonedb -o umount (failed)	Verify that your file system(s) are not busy and retry the command.
umount_end	dbed_vmclonedb -o umount (successful)	■ dbed_vmclonedb -o mount ■ dbed_vmclonedb -o restartdb ■ dbed_vmsnap -o resync

Snapshot database status information from the CLI

To view snapshot database status information from the command line, use the dbed_vmchecksnap command with the -o list option to list all available snapplans for a specified database. Snapshot database status information is displayed in the command output under the column heading DB_STATUS.

Table C-2 shows detailed information about each database status (DB_STATUS) value.

Note: SF Oracle RAC does not support Database FlashSnap reverse resynchronization.

Table C-2 Snapshot database status information from the CLI

DB_STATUS	Completed operations
`init`	■ `dbed_vmchecksnap -o validate` (successful) ■ `dbed_vmsnap -o snapshot` (successful)
`database_recovered`	`dbed_vmclonedb -o recoverdb` (successful)

Appendix D

Using third party software to back up files

This appendix includes the following topics:

- About using third party software to back up files
- Using third party software to back up files

About using third party software to back up files

SF Enterprise products supports the use of third party software for backing up files. However, Quick I/O is not supported for SF Enterprise products.

Using third party software to back up files

If you are using third party backup software other than Veritas NetBackup, ensure that it can back up and restore VxFS extent attributes. This is important because restored Oracle Disk Manager (ODM) files rely on proper extent layouts for best performance.

Backing up and restoring Oracle Disk Manager files using Oracle RMAN

Oracle allocates Oracle Disk Manager files with contiguous extent layouts for good database performance. When you restore database files they are allocated using these extent attributes. If you are using Oracle RMAN's conventional backup method with any backup software, datafiles are also restored with the proper extent layouts.

If you are using RMAN's “proxy copy” backup method with a backup software other than NetBackup, the extent attributes may not be backed up. To ensure the restored datafiles have proper extent layouts, preallocate the lost datafiles using the `odmmkfile` command. This command preallocates contiguous space for files prior to restoring them.

For example, to preallocate an Oracle datafile with size 100M, assuming the Oracle database block size is 8K, use the `odmmkfile` command and enter:

```
# /opt/VRTS/bin/odmmkfile -h 8k -s 100m filename
```

For additional information about the `odmmkfile` command, see the `odmmkfile(1)` manual page.

Index

S

T

U

V