VERITAS™

VERITAS Storage Foundation™ 4.1 *for Oracle*

Administrator's Guide

Solaris

N14624F

March 2005

Disclaimer

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

VERITAS Legal Notice

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

Third-Party Copyrights

Data Encryption Standard (DES)

GNUPlot 3.7

Apache Jakarta Lucene

Sun Microsystems Trademarks

Contents

Preface

VERITAS Storage Foundation *for Oracle*, formerly known as VERITAS Database Edition *for Oracle*, is an integrated set of system software enhancements and configuration guidelines that combine to help Oracle database administrators configure a database system with high performance, availability, manageability, and reliability.

What's in This Guide?

This guide is organized as follows:

Chapters in This Guide

Chapter	Description
Chapter 1. "VERITAS Storage Foundation for Oracle" on page 1	Introduces the features and characteristics of VERITAS Storage Foundation *for Oracle*.
Chapter 2. "Setting Up Databases" on page 31	Discusses how to select volume layouts and create optimal file system and database configurations.
Chapter 3. "Using VERITAS Quick I/O" on page 67	Describes how to set up and use VERITAS Quick I/O.
Chapter 4. "Using VERITAS Cached Quick I/O" on page 97	Describes how to set up and use VERITAS Cached Quick I/O.
Chapter 5. "Using VERITAS Extension for Oracle Disk Manager" on page 111	Describes how Oracle Disk Manager improves I/O performance.
Chapter 6. "Using Storage Mapping" on page 125	Discusses how to use Oracle9*i*'s topology mapping feature with VERITAS Storage Foundation *for Oracle* to map datafiles to logical volumes and physical devices.

Chapters in This Guide

Chapter	Description
Chapter 7. "Converting Existing Database Configurations to VxFS" on page 147	Discusses how to convert existing databases to VxFS file systems.
Chapter 8. "Using Storage Checkpoints and Storage Rollback" on page 155	Discusses how to use Storage Checkpoint and Storage Rollback.
Chapter 9. "Using the Space Capacity Planning Utility for Storage Checkpoints" on page 183	Discusses how to use the Storage Checkpoint Capacity Planning utility to plan space for Storage Checkpoints.
Chapter 10. "Using Database FlashSnap for Backup and Off-Host Processing" on page 197	Describes the online snapshot facilities provided with VxVM.
Chapter 11. "Using VERITAS NetBackup for Database Backup" on page 261	Describes the online backup and recovery facilities that VERITAS NetBackup provides.
Chapter 12. "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267	Describes various Oracle database administrative operations that can be easily performed using the VxDBA graphical user interface (GUI).
Chapter 13. "Using the VxDBA Utility" on page 333	Describes various Oracle database administrative operations that can be easily performed using the VxDBA utility.
Chapter 14. "Tuning for Performance" on page 407	Provides tuning tips and describes commands you can use to monitor and tune database performance.
Appendix A. "VERITAS Storage Foundation for Oracle Command Line Interface" on page 425	Describes the command line interface to key operations also provided in the VxDBA utility.
Appendix B. "Using Third-Party Software to Back Up Files" on page 489	Describes special consideration and tips for using popular third-party software to back up VERITAS Quick I/O files.
Appendix B. "VERITAS Database FlashSnap Status Information" on page 495	Describes the various snapshot and snapshot database status messages you will see in the VERITAS Database FlashSnap CLI and GUI.

Chapters in This Guide

Chapter	Description
Appendix B. "Accessibility and VERITAS Storage Foundation for Oracle" on page 509	Provides information relating to Storage Foundation *for Oracle*'s conformance with Section 508 of the Rehabilitation Act.
"Glossary" on page 521	Provides terminology and general background information for Oracle database administration.

How to Use This Guide

This guide is intended for database and system administrators responsible for configuring and maintaining Oracle databases with VERITAS Storage Foundation *for Oracle*, which includes:

- VERITAS Volume Manager (VxVM)
- VERITAS File System (VxFS) with Quick I/O
- VERITAS Extension for Oracle Disk Manager
- VERITAS Storage Foundation Graphical User Interface
- VERITAS VxDBA Menu Utility
- VERITAS Enterprise Administrator (VEA)
- VERITAS Database FlashSnap (included with the Enterprise Edition)
- VERITAS Storage Mapping (included with the Enterprise Edition)

This guide assumes that the administrator has a:

- Basic understanding of system and database administration
- Working knowledge of the operating system
- General understanding of file systems

Getting Help

VERITAS offers you a variety of support options.

Accessing the VERITAS Support Web Site

The VERITAS Support Web site allows you to:

- contact the VERITAS Support staff and post questions to them
- get the latest patches, upgrades, and utilities
- view the VERITAS Storage Foundation *for Oracle* Frequently Asked Questions (FAQ) pages
- search the knowledge base for answers to technical support questions
- receive automatic notice of product updates
- find out about VERITAS Storage Foundation *for Oracle* training
- read current white papers related to VERITAS Storage Foundation *for Oracle*

The address for the VERITAS Support Web site is:

- http://support.veritas.com

Subscribing to VERITAS Email Notification Service

Subscribe to the VERITAS Email notification service to be informed of software alerts, newly published documentation, Beta programs, and other services.

Go to http://support.veritas.com. Select a product and click “E-mail Notifications” on the right side of the page. Your customer profile ensures you receive the latest VERITAS technical information pertaining to your specific interests.

Accessing VERITAS Telephone and Fax Support

Telephone support for VERITAS Storage Foundation *for Oracle* is only available with a valid support contract. To contact VERITAS for technical support, dial the appropriate phone number listed on the Support Guide included in the product box and have your product license information ready for quick navigation to the proper support group.

The address for the VERITAS telephone support directory is:

- http://support.veritas.com

Contacting VERITAS Licensing

For license information call 1-800-634-4747 option 3, fax 1-650-527-0952, or e-mail amercustomercare@veritas.com.

VERITAS Storage Foundation for Oracle Guides

The following guides, along with the online help, comprise the VERITAS Storage Foundation *for Oracle* documentation set:

Guides in VERITAS Storage Foundation for Oracle Documentation Set

Guide Title	Filename
VERITAS Storage Foundation Release Notes	`sf_notes.pdf`
VERITAS Storage Foundation Installation Guide	`sf_install.pdf`
VERITAS Storage Foundation for Oracle Administrator's Guide	`sf_ora_admin.pdf`
VERITAS Storage Foundation Intelligent Storage Provisioning Administrator's Guide	`sf_isp_admin.pdf`
VERITAS Storage Foundation Cross-Platform Data Sharing Administrator's Guide	`sf_cds_admin.pdf`
VERITAS Array Integration Layer Configuration Guide	`vail_config.pdf`
VERITAS File System Administrator's Guide	`vxfs_admin.pdf`
VERITAS File System Programmer's Reference Guide	`vxfs_ref.pdf`
VERITAS Volume Manager Administrator's Guide	`vxvm_admin.pdf`
VERITAS Enterprise Administrator (VEA 500 Series) Getting Started	`vea5x_getting_started.pdf`
VERITAS FlashSnap Point-In-Time Copy Solutions Administrator's Guide	`flashsnap_admin.pdf`
VERITAS Volume Manager Hardware Notes	`vxvm_hwnotes.pdf`
VERITAS Volume Manager Troubleshooting Guide	`vxvm_tshoot.pdf`

Related Resources

The following documents describe how to use the FlashSnap agent with EMC TimeFinder on EMC Symmetrix storage systems:

- *VERITAS FlashSnap Agent for Symmetrix Release Notes* contains last-minute information pertaining to the FlashSnap agent *for Symmetrix* product.
- *VERITAS FlashSnap Agent for Symmetrix Installation guide* explains how to install the FlashSnap agent *for Symmetrix* product.
- *VERITAS FlashSnap Agent for Symmetrix Administrator's Guide* provides information about the set of commands that allows you to use EMC TimeFinder in conjunction with VERITAS Volume Manager disk groups and volumes that have been created on Symmetrix standard devices.
- *VERITAS Cluster Server Agents for VERITAS FlashSnap Agent for Symmetrix Installation and Configuration Guide* explains how to install add-ons to VERITAS Cluster Server to enable the automatic recovery of the FlashSnap agent *for Symmetrix* operations.

While not shipped with VERITAS Storage Foundation, the following documents provide related information if you plan to use VERITAS NetBackup™ to back up your databases:

- The *VERITAS NetBackup Release Notes* provide important, up-to-date, and release-specific information for VERITAS NetBackup. Reading all of the *Release Notes* before installing or using any VERITAS products is recommended.
- The *VERITAS NetBackup BusinessServer Getting Started Guide* explains how to install and configure VERITAS NetBackup.
- The *VERITAS NetBackup User's Guide* explains how to use VERITAS NetBackup to back up, archive, and restore files and directories.
- The *VERITAS NetBackup System Administrator's Guide for UNIX, Volume I* and *VERITAS NetBackup System Administrator's Guide for UNIX, Volume II* describe how to configure and manage the operation of VERITAS NetBackup.
- The *VERITAS NetBackup BusinesServer Media Manager System Administrator's Guide* and *VERITAS NetBackup DataCenter Media Manager System Administrator's Guide* describe how to use the extensive media management capabilities of VERITAS NetBackup.
- The *VERITAS NetBackup for Oracle Advanced BLI Agent System Administrator's Guide* and *VERITAS NetBackup for Advanced BLI Agent System Administrator's Guide for Backups without RMAN* provides information on how to configure and use the Advanced BLI Agent to perform Oracle database backups.

Conventions

Typographical and Symbolic

The following tables explain the typographical and symbolic conventions used throughout the guides:

Typeface Conventions

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

Symbolic Conventions

Symbol	Usage	Examples
`%`	C shell prompt	
`$`	Bourne/Korn shell prompt	
`#`	Superuser prompt (all shells)	

`SVRMGR>`	Oracle SQL prompt for Oracle8*i* and earlier.	`SVRMGR>` **`alter tablespace ts1`** `\`
		`begin backup;`

SQL>	Oracle SQL prompt for Oracle8*i* and later.	`SQL>` **`alter tablespace ts1`** `\`
		`begin backup;`

`\`	Continued input on the following line; you do not type this character	`#` **`mkfs -F vxfs -o largefiles`** `\`
		`/dev/vx/rdsk/PRODdg/db01`

Symbolic Conventions

Symbol	Usage	Examples
[ ]	In a command synopsis, brackets indicates an optional argument	`ls [ -a ]`
|	In a command synopsis, a vertical bar separates mutually exclusive arguments	`mount [ suid | nosuid ]`
blue text	In PDF and HTML files, click on these active hyperlinks to move to the specified location	See "Using Snapshots for Database Backup" on page 97 for more information.

Notes and Cautions

Note A Note provides information that makes it easier to use the product or helps you avoid problems.

Caution A Caution warns you about situations that can cause data loss.

Comment on the Documentation

Let us know what you like and dislike about the documentation. Were you able to find the information you needed quickly? Was the information clearly presented? Report errors and omissions, or tell us what you would find useful in future versions of our manuals and online help.

Please include the following information with your comment:

- The title and product version of the manual you are commenting on
- The topic (if relevant) you are commenting on
- Your comment
- Your name

Email your comment to the following address:

- sfuadocs@veritas.com

Please only use this address to comment on product documentation. See "Getting Help" on page xxi for how to contact Technical Support about our software.

We appreciate your feedback.

VERITAS Storage Foundation for Oracle

1

This chapter provides an overview of the features, components, and database-related functionality of VERITAS Storage Foundation *for Oracle*.

Topics include:

VERITAS Storage Foundation for Oracle

Standard Edition/Enterprise Edition

There are two versions of this product:

- VERITAS Storage Foundation *for Oracle* Standard Edition
- VERITAS Storage Foundation *for Oracle* Enterprise Edition

  The Enterprise Edition contains everything in the Standard Edition plus Storage Mapping, Database FlashSnap, Storage Checkpoints, and Storage Rollback.

Note VERITAS Storage Foundation/High Availability (HA) *for Oracle* is available only with the Enterprise Edition.

Unless otherwise noted, features pertain to both the Standard and Enterprise Edition products.

Features

VERITAS Storage Foundation *for Oracle* combines the strengths of the core VERITAS technology products with database-specific enhancements to offer performance, availability, and manageability for Oracle database servers.

VERITAS Storage Foundation *for Oracle* includes the following products:

- VERITAS Volume Manager (VxVM)

  A disk management subsystem that supports disk striping, disk mirroring, and simplified disk management for improved data availability and performance.

- VERITAS Database FlashSnap

  Database FlashSnap, a feature of the Enterprise Edition, lets you 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 FlashSnap tasks may be performed through the VERITAS Storage Foundation *for Oracle* GUI or the command line interface.

- VERITAS Storage Mapping

  Storage Mapping, a feature of the Enterprise Edition, lets you take full advantage of Oracle storage mapping to map datafiles to physical devices and display storage object I/O statistics. Oracle's file mapping (I/O topology) feature was introduced beginning with Oracle9*i* Release 2.

  Both storage object I/O statistics and the storage structure can be displayed for a specific file using either the `vxstorage_stats` command or the VERITAS Storage Foundation *for Oracle* GUI. In addition, mapping information showing which tablespaces reside on which physical disks can be obtained for a specified database using the `dbed_analyzer` command.

- VERITAS File System (VxFS)

  A high-performance, fast-recovery file system that is optimized for business-critical database applications and data-intensive workloads. VxFS offers online administration, letting you perform most frequently scheduled maintenance tasks (including online backup, resizing, and file system changes) without interrupting data or system availability. VxFS also provides support for large file systems (up to 8 exabytes.)

  VERITAS File System offers performance-enhancing features that are of particular interest in a database environment:

  - VERITAS Quick I/O is a VxFS feature that improves the throughput for Oracle databases built on VERITAS 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.
  - A feature of the Enterprise Edition, VxFS Storage Checkpoint technology lets you 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 *for Oracle* administrative utilities.

- VERITAS Extension for Oracle Disk Manager

  VERITAS Extension for Oracle Disk Manager is a custom storage interface designed specifically for Oracle9*i* and 10*g*. Oracle Disk Manager allows Oracle9*i* and 10*g* to improve performance and manageability system bandwidth through an improved Application Programming Interface (API) that contains advanced kernel support for file I/O.

- VERITAS VxDBA Menu Utility

  The VxDBA menu-driven utility allows you to perform various administrative tasks, including database monitoring, Storage Checkpoint and Storage Rollback administration, and file system space planning capabilities for Storage Checkpoints.

  You can also use the graphical user interface or command line interface to perform these tasks.

- VERITAS Enterprise Administrator

  VERITAS Enterprise Administrator (VEA) is the infrastructure that allows you to access VERITAS Storage Foundation *for Oracle*, VERITAS Volume Manager, and VERITAS File System information and features through the GUI.

- VERITAS NetBackup *for Oracle* Advanced BLI Agent (optional)

  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.)

An optional High Availability (HA) version of VERITAS Storage Foundation *for Oracle* Enterprise Edition, which includes VERITAS Cluster Server, is available for customers who have high system-availability requirements.

The following is an example of the VERITAS Storage Foundation *for Oracle* storage stack:

<table>
<tr><td colspan="3">Oracle</td></tr>
<tr><td>VxFS</td><td>Oracle Disk Manager</td><td>Quick I/O</td></tr>
<tr><td colspan="3">VxVM</td></tr>
<tr><td colspan="2">Disks</td><td>RAID</td></tr>
</table>

VERITAS Volume Manager

Databases require their storage media to be robust and resilient to failure. It is vital to protect against hardware and disk failures and to maximize performance using all the available hardware resources. Using a volume manager provides this necessary resilience and eases the task of management. A volume manager can help you manage hundreds of disk devices and makes spanning, striping, and mirroring easy.

VERITAS Volume Manager (VxVM) builds virtual devices called *volumes* on top of physical disks. Volumes are accessed by a file system, a database, or other applications in the same way physical disk partitions would be accessed. Using volumes, VxVM provides the following administrative benefits for databases:

- Spanning of multiple disks—eliminates media size limitations.
- Striping—increases throughput and bandwidth.
- Mirroring or RAID-5—increases data availability.
- Online relayout—allows online volume layout changes to improve database performance.
- Volume resynchronization—ensures that all mirrors contain exactly the same data and that the data and parity in RAID-5 volumes agree.
- Dirty Region Logging (DRL)—speeds the recovery of mirrored volumes after a system crash.
- SmartSync Recovery Accelerator—increases the availability of mirrored volumes by only resynchronizing changed data.
- Volume snapshots—allows backup of volumes based on disk mirroring. With this release, VxVM introduces full-sized and space-optimized instant snapshots, the preferred way to implement online and off-host point-in-time copy solutions.
- FastResync—separately licensed, optional feature that performs quick and efficient resynchronization of stale mirrors. FastResync is included with the Enterprise Edition and is also included as part of the VERITAS FlashSnap option with the Standard Edition.
- Disk group split and join—separately licensed, optional feature that supports general disk group reorganization and allows you to move volume snapshots to another host for off-host backup. Disk group split and join is included with the Enterprise Edition and is also included as part of the VERITAS FlashSnap option with the Standard Edition.
- Hot-relocation—automatically restores data redundancy in mirrored and RAID-5 volumes when a disk fails.
- Dynamic multipathing (DMP)—allows for transparent failover, load sharing, and hot plugging of SCSI devices.

- Volume sets—allows several volumes to be represented by a single logical mount device.
- Dynamic LUN Expansion—allows you to resize a disk after it has been initialized while preserving the existing data on the disk.
- Storage Expert—helps diagnose configuration problems with VxVM.
- Cluster Volume Manager (CVM)—separately licensed, optional feature that allows you to use VxVM in a cluster environment.
- VERITAS FlashSnap Agent for *Symmetrix*—separately licensed, optional feature that includes a set of commands that allows you to use EMC TimeFinder in conjunction with VxVM disk groups and volumes that have been created on Symmetrix standard devices.
- VERITAS Volume Replicator (VVR)—separately licensed, optional feature that provides data replication for disaster recovery planning.
- Free space pool management—simplifies administration and provides flexible use of available hardware.
- Online administration—allows configuration changes without system or database down time.

The following sections provide brief overviews of VxVM concepts and features that are relevant to database administration. The information and examples presented in the remainder of this guide assume that you are using VERITAS Volume Manager. For a more detailed description of VxVM and its features, refer to the *VERITAS Volume Manager Administrator's Guide.*

Volumes

A *volume* is a virtual disk device that appears to applications, databases, and file systems like a physical disk partition without the physical limitations of a disk partition. A volume consists of one or more plexes, each holding a copy of the selected data in the volume. Due to its virtual nature, a volume is not restricted to a particular disk or a specific area of a disk. For example, a volume can span multiple disks and can be used to create a large file system.

Volumes consist of other virtual objects that can be manipulated to change the volume's configuration. Volumes and their virtual components are referred to as *Volume Manager objects*. You can manipulate VERITAS Volume Manager objects in a variety of ways to optimize performance, provide redundancy of data, and perform backups or other administrative tasks on one or more physical disks without interrupting applications. As a result, data availability and disk subsystem throughput are improved.

You can change the configuration of a volume without causing disruption to databases or file systems that are using the volume. For example, you can mirror a volume on separate disks or move the volume to use different disk storage.

Disk Groups

A *disk group* is a collection of disks that share a common configuration (for example, configuration objects that belong to a single database). We recommend creating one disk group for each database.

You can move a disk group and its components as a unit from one host to another host. For example, you can move volumes and file systems that belong to the same database and are created within one disk group as a unit. You must configure a given volume from disks belonging to one disk group.

In releases before VERITAS Storage Foundation 4.0 *for Oracle*, the default disk group was `rootdg`. For VxVM to function, the rootdg disk group had to exist and it had to contain at least one disk. This requirement no longer exists, and VxVM can work without any disk groups configured (although you must set up at least one disk group before you can create any volumes of other VxVM objects).

Volume Layouts

A *Redundant Array of Independent Disks* (RAID) is a disk array in which a group of disks appears to the system as a single virtual disk or a single volume. VxVM supports several RAID implementations, as well as spanning. The following volume layouts are available to satisfy different database configuration requirements:

- Spanning and concatenation
- Striping (RAID-0)
- Mirroring (RAID-1)
- Mirrored-Stripe Volumes (RAID-0+1)
- Striped-Mirror Volumes (RAID-1+0)
- RAID-5

Caution Spanning or striping a volume across multiple disks increases the chance that a disk failure will result in failure of that volume. Use mirroring or RAID-5 to substantially reduce the chance of a single volume failure caused by a single disk failure.

Spanning and Concatenation

Concatenation maps data in a linear manner onto one or more subdisks in a plex. To access all of the data in a concatenated plex sequentially, data is first accessed in the first subdisk from beginning to end. Data is then accessed in the remaining subdisks sequentially from beginning to end, until the end of the last subdisk.

The subdisks in a concatenated plex do not have to be physically contiguous and can belong to more than one VM disk. Concatenation using subdisks that reside on more than one VM disk is called *spanning*.

Spanning is useful when you need to read or write data sequentially (for example, reading from or writing to database redo logs) and there is not sufficient contiguous space.

Striping (RAID-0)

Striping is a technique of mapping data so that the data is interleaved among multiple physical disks. Data is allocated in equal-sized units (called *stripe units*) that are interleaved between the disks. Each stripe unit is a set of contiguous blocks on a disk. A *stripe* consists of the set of stripe units at the same position across all columns. A column is a set of one or more subdisks within a striped plex.

Striping is useful if you need large amounts of data written to or read from physical disks, and performance is important. Striping is also helpful in balancing the I/O load from multi-user applications across multiple disks. By using parallel data transfer to and from multiple disks, striping significantly improves data-access performance.

When striping across multiple disks, failure of any one disk will make the entire volume unusable.

Mirroring (RAID-1)

Mirroring is a technique of using multiple copies of the data, or mirrors, to duplicate the information contained in a volume. In the event of a physical disk failure, the mirror on the failed disk becomes unavailable, but the system continues to operate using the unaffected mirrors. For this reason, mirroring increases system reliability and availability. A volume requires at least two mirrors to provide redundancy of data. A volume can consist of up to 32 mirrors. Each of these mirrors must contain disk space from different disks for the redundancy to be effective.

Striping Plus Mirroring (Mirrored-Stripe or RAID-0+1)

VxVM supports the combination of mirroring with striping. When used together on the same volume, mirroring plus striping offers the benefits of spreading data across multiple disks while providing redundancy of data.

Mirrored-stripe volumes have multiple plexes as mirrors, each constructed as a striped plex. Allocate subdisks used in the same striped plex from separate disks, and use a disk in only one mirror of a volume.

Mirroring Plus Striping (Striped-Mirror Volumes, RAID-1+0 or RAID-10)

VxVM supports the combination of striping with mirroring. When used together on the same volume, striping plus mirroring offers the benefits of spreading data across multiple disks while providing redundancy of data.

Striped-mirror volumes combine striping and mirroring, but the mirroring is done at stripe column level. In case of failure, this type of volume recovers faster than RAID-0+1 volumes and the tolerance for disk failure is greater.

For databases that support online transaction processing (OLTP) workloads, we recommend either mirrored-stripe or striped-mirror volumes to improve database performance and reliability. For highest availability, we recommend striped-mirror volumes (RAID 1+0).

RAID-5 (Striping with Parity)

RAID-5 provides data redundancy through the use of *parity* (a calculated value that the system uses to reconstruct data after a failure). While data is written to a RAID-5 volume, parity is also calculated by performing an *exclusive OR* (XOR) procedure on data. The resulting parity is then written to another part of the volume. If a portion of a RAID-5 volume fails, the data that was on that portion of the failed volume can be recreated from the remaining data and the parity.

RAID-5 offers data redundancy similar to mirroring, while requiring less disk space. RAID-5 read performance is similar to that of striping but with relatively slow write performance. RAID-5 is useful if the database workload is read-intensive (as in many data warehousing applications). You can snapshot a RAID-5 volume and move a RAID-5 subdisk without losing redundancy.

Online Relayout

As databases grow and usage patterns change, online relayout lets you change volumes to a different layout, with uninterrupted data access. Relayout is accomplished online and in place. Use online relayout to change the redundancy or performance characteristics of the storage, such as data organization (RAID levels), the number of columns for RAID-5 and striped volumes, and stripe unit size.

Volume Resynchronization

When storing data redundantly, using mirrored or RAID-5 volumes, VERITAS Volume Manager ensures that all copies of the data match exactly. However, if the system crashes, small amounts of the redundant data on a volume can become inconsistent or *unsynchronized*. For mirrored volumes, unsynchronized data can cause two reads from the same region of the volume to return different results if different mirrors are used to satisfy the read request. In the case of RAID-5 volumes, unsynchronized data can lead to parity corruption and incorrect data reconstruction.

In the event of a system crash, VERITAS Volume Manager ensures that all mirrors contain exactly the same data and that the data and parity in RAID-5 volumes agree. This process is called *volume resynchronization*. Not all volumes require resynchronization after a system failure. VxVM notices when a volume is first written and marks it as *dirty*. Only volumes that are marked dirty when the system reboots require resynchronization.

The process of resynchronization can impact system and database performance. However, it does not affect the availability of the database after system reboot. You can immediately access the database after database recovery although the performance may suffer due to resynchronization. For very large volumes or for a very large number of volumes, the resynchronization process can take a long time. You can significantly reduce resynchronization time by using Dirty Region Logging (DRL) for mirrored volumes or by making sure that RAID-5 volumes have valid RAID-5 logs. However, using logs can slightly reduce the database write performance.

For most database configurations, we recommend using dirty region logs or the RAID-5 logs when mirrored or RAID-5 volumes are used. It is also advisable to evaluate the database performance requirements to determine the optimal volume configurations for the databases.

Dirty Region Logging

Dirty Region Logging (DRL), if enabled, speeds the recovery of mirrored volumes after a system crash. DRL keeps track of the regions that have changed due to I/O writes to a mirrored volume. DRL uses this information to recover only those portions of the volume that need to be recovered.

Note In VxVM 4.1, if a version 20 data change object (DCO) volume is associated with a volume, a portion of the DCO volume can be used to store the DRL log. There is no need to create a separate DRL log for a volume that has a version 20 DCO volume.

SmartSync Recovery Accelerator

SmartSync 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 reynchronization when it is available.

Volume Sets

Volume sets are an enhancement to VxVM that allow several volumes to be represented by a single logical mount device. 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-device 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.

Volume Snapshots

A volume snapshot is a point-in-time image of a volume. VERITAS Volume Manager provides three volume snapshot features based on disk mirroring:

- Full-sized instant snapshots
- Space-optimized instant snapshots
- Emulation of third-mirror snapshots

For detailed information on performing database backups using volume snapshots, see "Using Database FlashSnap for Backup and Off-Host Processing" on page 197

VERITAS FastResync (Optional)

VERITAS FastResync (previously called Fast Mirror Resynchronization or FMR) is included with the Enterprise Edition. It is also included as part of the VERITAS FlashSnap option with the Standard Edition.

VERITAS FastResync performs quick and efficient resynchronization of stale mirrors (mirrors that are not synchronized). This increases the efficiency of the VxVM snapshot mechanism, and improves the performance of operations such as backup and decision support. Typically, these operations require that the volume is quiescent, and that they are not impeded by updates to the volume by other activities on the system. To achieve these goals, the snapshot mechanism in VxVM creates an exact copy of a primary volume at an instant in time. After a snapshot is taken, it can be accessed independently of the volume from which it was taken.

VERITAS Storage Foundation *for Oracle* Enterprise Edition includes a feature called Database FlashSnap, which takes advantage of the FastResync and disk group split and join features. Database FlashSnap provides a quicker and easier way for database administrators to use volume snapshots. For more information, see “VERITAS Database FlashSnap” on page 23.

Non-Persistent FastResync

Non-persistent FastResync allocates its change maps in memory. If non-persistent FastResync is enabled, a separate FastResync map is kept for the original volume and for each snapshot volume. Unlike a dirty region log (DRL), these maps do not reside on disk nor in persistent store. The advantage is that updates to the FastResync map have little impact on I/O performance, as no disk updates need to be performed. However, if a system is rebooted, the information in the map is lost, so a full resynchronization is required when performing a `snapback` operation. This limitation can be overcome for volumes in cluster-shareable disk groups, provided that at least one of the nodes in the cluster remains running to preserve the FastResync map in its memory.

Persistent FastResync

Non-persistent FastResync has been augmented by the introduction of persistent FastResync. Unlike non-persistent FastResync, Persistent FastResync keeps the FastResync maps on disk so that they can survive system reboots and system crashes. When the disk groups are rejoined, this allows the snapshot plexes to be quickly resynchronized. This ability is not supported by non-persistent FastResync.

If persistent FastResync is enabled on a volume or on a snapshot volume, a DCO and a *DCO log volume* are associated with the volume.

In VxVM 3.2 and 3.5, the DCO object only managed information about the FastResync maps. These maps track writes to the original volume (and to each of up to 32 snapshot volumes) since the last `snapshot` operation. The DCO log volume on disk holds the 33 maps, each of which is 4 blocks in size by default.

In VxVM 4.1, the DCO object is used not only to manage FastResync maps, but also to manage DRL recovery maps and special maps called copy maps that allow instant snapshot operations to be resume following a system crash.

Persistent FastResync can also track the association between volumes and their snapshot volumes after they are moved into different disk groups. When the disk groups are rejoined, this allows the snapshot plexes to be quickly resynchronized. This ability is not supported by non-persistent FastResync.

Disk Group Split and Join (Optional)

Disk group split and join is included with the Enterprise Edition. It is also included as part of the VERITAS FlashSnap option with the Standard Edition.

VxVM provides a disk group content reorganization feature that supports general disk group reorganization and allows you to move volume snapshots to another host for off-host backup. Additional options to the `vxdg` command enable you to take advantage of the ability to remove all VxVM objects from an imported disk group and move them to a newly created target disk group (`split`), and to remove all VxVM objects from an imported disk group and move them to an imported target disk group (`join`). The move operation enables you to move a self-contained set of VxVM objects between the imported disk groups.

Hot-Relocation

In addition to providing volume layouts that help improve database performance and availability, VxVM offers features that you can use to further improve system availability in the event of a disk failure. *Hot-relocation* is a feature that allows a system to react automatically to I/O failures on mirrored or RAID-5 volumes and restore redundancy and access to those volumes.

VxVM detects I/O failures on volumes and relocates the affected portions to disks designated as *spare disks* or free space within the disk group. VxVM then reconstructs the volumes that existed before the failure and makes them redundant and accessible again.

The hot-relocation feature is enabled by default and is recommended for most database configurations. After hot-relocation occurs, we recommend verifying the volume configuration for any possible performance impact. It is also a good idea to designate additional disks as spares to augment the spare pool.

While a disk is designated as a spare, you cannot use the space on that disk for the creation of VxVM objects within its disk group. VxVM also lets you free a spare disk for general use by removing it from the pool of hot-relocation disks.

DMP-Supported Disk Arrays

VxVM provides administrative utilities and driver support for disk arrays that can take advantage of its Dynamic Multipathing (DMP) feature. Some disk arrays provide multiple ports to access their disk devices. These ports, coupled with the host bus adaptor (HBA) controller and any data bus or I/O processor local to the array, make up multiple hardware paths to access the disk devices. Such disk arrays are called multipathed disk arrays. This type of disk array can be connected to host systems in many different configurations, (such as multiple ports connected to different controllers on a single host, chaining of the ports through a single controller on a host, or ports connected to different hosts simultaneously). DMP is available for multiported disk arrays from various vendors and provides improved reliability and performance by using path failover and load balancing.

See the *VERITAS Volume Manager Administrator's Guide* for detailed information and the *VERITAS Volume Manager Hardware Notes* for information about supported disk arrays.

Dynamic LUN Expansion

Dynamic LUN expansion allows you to resize a disk after it has been initialized while preserving the existing data on the disk. See the *VERITAS Volume Manager Administrator's Guide* for more information.

Storage Expert

Storage Expert consists of a set of simple commands that collect VxVM configuration data and compare it with "best practice." Storage Expert then produces a summary report that shows which objects do not meet these criteria and makes recommendations for VxVM configuration improvements.

These user-configurable tools help you as an administrator to verify and validate systems and non-optimal configurations in both small and large VxVM installations.

Storage Expert components include a set of rule scripts and a rules engine. The rules engine runs the scripts and produces ASCII output, which is organized and archived by Storage Expert's report generator. This output contains information about areas of VxVM configuration that do not meet the set criteria. By default, output is sent to the screen, but you can redirect it to a file using standard UNIX redirection. See the *VERITAS Volume Manager Administrator's Guide* for more information.

Cluster Functionality (Optional)

VxVM includes an optional, separately licensable clustering feature, known as Cluster Volume Manager, that enables VxVM to be used in a cluster environment. With the clustering option, VxVM supports up to 16 nodes per cluster. See the *VERITAS Volume Manager Administrator's Guide* for more information.

VERITAS Volume Replicator (Optional)

VERITAS Volume Replicator (VVR) is an optional, separately licensable feature of VxVM. VVR is a data replication tool designed to maintain a consistent copy of application data at a remote site. It is built to contribute to an effective disaster recovery plan. If the data center is destroyed, the application data is immediately available at the remote site, and the application can be restarted at the remote site.

VVR 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 can use existing VxVM configurations with some restrictions. Any application, even with existing data, can be configured to use VVR transparently.

See the VERITAS Volume Replicator documentation for more information.

VERITAS File System

VERITAS File System (referred to as VxFS) is an extent-based, intent logging file system intended for use in UNIX environments that deal with large volumes of data and that require high file system performance, availability, and manageability. VxFS also provides enhancements that make file systems more viable in database environments.

The following sections provide a brief overview of VxFS concepts and features that are relevant to database administration. For a more detailed description of VxFS and its complete feature set, see the *VERITAS File System Administrator's Guide*.

VERITAS Quick I/O

Databases can run on either file systems or raw devices. Database administrators often create their databases on file systems because it makes common administrative tasks (such as moving, copying, and backing up) easier. However, running databases on most file systems significantly reduces database performance.

When performance is an issue, database administrators create their databases on raw devices. VxFS with Quick I/O presents regular, preallocated files as raw character devices to the application. Using Quick I/O, you can enjoy the management advantages of databases created on file systems and achieve the same performance as databases created on raw devices. See "Using VERITAS Quick I/O" on page 67 for more information.

Quick I/O can be used on Oracle9*i* and Oracle 10*g*. However, if you are using Oracle9*i* or Oracle 10*g*, we recommend that you use Oracle Disk Manager.

VERITAS Cached Quick I/O

Cached Quick I/O allows databases to make more efficient use of large system memory while still maintaining the performance benefits of Quick I/O. Cached Quick I/O provides an efficient, selective buffering mechanism to back asynchronous I/O. Using Cached Quick I/O, you can enjoy all the benefits of Quick I/O and achieve even better performance.

Cached Quick I/O is first enabled for the file system and then enabled on a per file basis. See "Using VERITAS Cached Quick I/O" on page 97 for more information.

Extent-Based Allocation

The UFS file system supplied with Solaris uses block-based allocation schemes that provide good random access to files and acceptable latency on small files. For larger files, such as database files, this block-based architecture limits throughput. This limitation makes the UFS file system a less than optimal choice for database environments.

The VxFS file system addresses this performance issue by allocating storage in groups of extents rather than a block at a time. An *extent* is one or more adjacent blocks of data within the file system. An extent is presented as an *address-length* pair that identifies the starting block address and the length of the extent (in file system or logical blocks). When storage is allocated to a file on a VxFS file system, it is grouped in extents, as opposed to being allocated a block at a time as with the UFS file system.

By allocating disk space to files in extents, disk I/O to and from a file can be done in units of multiple blocks. This type of I/O can occur if storage is allocated in units of consecutive blocks. For sequential I/O, multiple block operations are considerably faster than block-at-a-time operations. Almost all disk drives accept I/O operations of multiple blocks.

The VxFS file system allocates disk space to files in groups of one or more extents. VxFS also allows applications to control some aspects of the extent allocation for a given file. *Extent attributes* are the extent allocation policies associated with a file.

For information on how to create preallocated database files using extent attributes, see "Preallocating Space for Quick I/O Files Using the setext Command" on page 74.

Fast File System and Database Recovery

VERITAS File System begins recovery procedures within seconds after a system failure by using a tracking feature called *intent logging*. This feature records pending changes to the file system structure in a circular intent log. The intent log recovery feature is not readily apparent to users or a system administrator except during a system failure. During system failure recovery, the VxFS `fsck` utility performs an intent log replay, which scans the intent log and nullifies or completes file system operations that were active when the system failed. The file system can then be mounted without completing a full structural check of the entire file system. Replaying the intent log may not completely recover the damaged file system structure if there was a disk hardware failure; hardware problems may require a complete system check using the `fsck` utility provided with VERITAS File System.

Online System Administration

The VxFS file system provides online system administration utilities to help resolve certain problems that impact database performance. You can defragment and resize a VxFS file system while it remains online and accessible to users.

Defragmentation Utility

Free resources are originally aligned in the most efficient order possible and are allocated to files in a way that is considered by the system to provide optimal performance. When a file system is active for extended periods of time, new files are created, old files are removed, and existing files grow and shrink. Over time, the original ordering of free resources is lost and the file system tends to spread along the disk, leaving unused gaps or *fragments* between areas that are in use. This process, known as *fragmentation*, leads to degraded performance because the file system has fewer choices when selecting an extent (a group of contiguous data blocks) to assign to a file. You should analyze the degree of fragmentation before creating new database files.

VxFS provides the online administration utility `fsadm` to resolve fragmentation problems. The utility can be run on demand and should be scheduled regularly as a `cron` job.

Resizing Utility

Changes in database size can result in file systems that are too large or too small for the current database. Without special utilities, expanding or shrinking a file system becomes a a matter of stopping applications, offloading the contents of the file system, rebuilding the file system to a new size, and then restoring the data. Data is unavailable to users while these administrative tasks are performed.

The VxFS file system utility `fsadm` provides a mechanism to resize file systems without unmounting them or interrupting users' productivity. Because the VxFS file system can only be mounted on one device, expanding a file system means that the underlying device must also be expandable while the file system is mounted. Working with VxVM, VxFS provides online expansion capability.

Cross-Platform Data Sharing

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. See the *VERITAS Storage Foundation Cross-Platform Data Sharing Administrator's Guide* for more information. 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 for moving sets of tablespaces between databases on the same platform in Oracle9*i*.

Quality of Storage Service (Optional)

The Quality of Storage Service (QoSS) feature is included with the Enterprise Edition.

The QoSS option is built on the multi-volume support technology introduced in this release. Using QoSS, you can map more than one device to a single file system. You can then configure policies that automatically relocate files from one device to another, or relocate files by running file relocation commands. Having multiple devices lets you determine where files are located, which can improve performance for applications that access specific types of files and reduce storage-related costs.

Support for Large File Systems and Large Files (Optional)

Support for large file systems is included with the Enterprise Edition.

In conjunction with VxVM, VxFS can support file systems up to 8 exabytes in size. You have the option of creating a file system using:

- Version 4 disk layout, which supports file systems up to one terabyte. The Version 4 disk layout encompasses all file system structural information in files, rather than at fixed locations on disk, allowing for greater scalability.
- Version 5, which supports file systems up to 32 terabytes. Files can be a maximum of two terabytes. File systems larger than one terabyte must be created on a VERITAS Volume Manager volume.
- Version 6, which supports file systems up to 8 exabytes. The Version 6 disk layout enables features such as multi-device support, Cross-Platform Data Sharing, named data streams, file change log. File systems created on VxFS 4.1 will by default use the Version 6 disk layout. An online conversion utility, `vxupgrade`, is provided to upgrade existing disk layouts to Version 6 on mounted file systems.

For large database configurations, this eliminates the need to use multiple file systems because of the size limitations of the underlying physical devices.

Changes implemented with the VxFS Version 4 disk layout have greatly expanded file system scalability, including support for large files. You can create or mount file systems with or without large files by specifying either the `largefiles` or `nolargefiles` option in `mkfs` or `mount` commands. See “Creating a VxFS File System” on page 50 for more information.

Multi-Volume File System Support

The multi-volume file system (MVS) feature allows several volumes to be represented by a single logical object. All I/O to and from an underlying logical volume is directed by way of *volume sets.* A volume set is a container for multiple different volumes. This feature can be used only in conjunction with VERITAS Volume Manager.

Storage Checkpoint and Storage Rollback

The Storage Checkpoint and Storage Rollback features are included with the Enterprise Edition. With the Standard Edition, they can be purchased as part of the VERITAS FlashSnap option.

VERITAS File System provides a *Storage Checkpoint* facility that allows you to create a persistent, point-in-time image of all user files in a file system—the Storage Checkpoint remains even after the file system is unmounted or the system is rebooted. Storage Checkpoints present a view of a file system at a point in time, and subsequently identify and maintain copies of the original file system blocks. Instead of using a disk-based mirroring method, Storage Checkpoints save disk space and significantly reduce I/O overhead by using the free space pool available to a file system.

The time required to create a Storage Checkpoint is typically only a couple of seconds. After a Storage Checkpoint is created, a consistent database backup image is made and the database can then resume its normal operation.

The Storage Rollback facility can then be used for rolling back the file system image to the point in time when the Storage Checkpoints were taken. In addition, Storage Checkpoints also keep track of the block change information that enables incremental database backup at the block level.

Storage Checkpoints are writable, and can be created, mounted, and removed. Performance enhancements in maintaining *Data Storage Checkpoints* (Storage Checkpoints that are complete images of the file system) makes using the *Storage Rollback* feature easier and more efficient, therefore more viable for backing up large databases.

Multi-Volume File System (MVS) Storage Checkpoint creation allows database backups without having to shut down the database.

MVSs provide the ability to create and administer Storage Checkpoint allocation policies. Storage Checkpoint allocation policies specify a list of volumes and the order in which to allocate Storage Checkpoint data to them. These allocation policies can be used to control where a Storage Checkpoint is created, allowing for separating Storage Checkpoint metadata and data onto different volumes. They can also be used to isolate data allocated to a Storage Checkpoint from the primary file system, which can help prevent the Storage Checkpoint from fragmenting space in the primary file system.

For more information on understanding and using Storage Checkpoints, see "Using Storage Checkpoints and Storage Rollback" on page 155. For more information on using VxDBA and the GUI to manage Storage Checkpoints and Storage Rollback, see "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267 and "Using the VxDBA Utility" on page 333.

Storage Checkpoint File System Restores

Storage Checkpoints can be used by backup and restore applications to restore either individual files or an entire file system. Restoring from Storage Checkpoints can recover data from incorrectly modified files, but typically cannot be used to recover from hardware damage or other file system integrity problems. File restoration can be done using the `fsckpt_restore`(1M) command. See the *VERITAS File System Administrator's Guide* for more information.

Quotas

VxFS supports quotas, which allocate per-user and per-group quotas and limit the use of two principal resources: files and data blocks. You can assign quotas for each of these resources. Each quota consists of two limits for each resource:

- The *hard limit* represents an absolute limit on data blocks or files. A user can never exceed the hard limit under any circumstances.
- The *soft limit* is lower than the hard limit and can be exceeded for a limited amount of time. This allows users to temporarily exceed limits as long as they fall under those limits before the allotted time expires.

You can use quotas to limit the amount of file system space used by Storage Checkpoints. With VERITAS Storage Foundation *for Oracle*, you can enable, disable, set, and display quota values for a single file system, for multiple file systems, or for all file systems in a database using the `dbed_ckptquota` command.

For details on using VxFS quotas, see the *VERITAS File System Administrator's Guide*. For information about `dbed_ckptquota`, see "VERITAS Storage Foundation for Oracle Command Line Interface" on page 425.

Cluster Functionality (Optional)

File system clustering is an optional, separately licensed feature of VxFS, where one system is configured as a primary server for the file system, and the other members of a cluster are configured as secondaries. All servers access shared disks for file data operations. If the primary server fails, one of the secondary servers takes over the file system operations. See the *VERITAS File System Administrator's Guide* for more information.

VERITAS Storage Mapping

VERITAS Storage Mapping is a feature included with VERITAS Storage Foundation *for Oracle* Enterprise Edition.

VERITAS has defined and implemented a library called VERITAS Federated Mapping Service (VxMS) that provides a mapping interface to VxFS file systems, VxVM volumes, and physical disks. With VERITAS Storage Foundation *for Oracle*, you can take full advantage of this feature to map datafiles to physical devices and display storage object I/O statistics. With the `vxstorage_stats` command, you can view the complete I/O topology mapping of datafiles through intermediate layers like logical volumes down to actual physical devices. You can also use `vxstorage_stats` to view statistics for VxFS file systems, VxVM volumes, and physical devices. This information can be used to determine the exact location of a data block on a disk and to help identify hot spots.

In addition to `vxstorage_stats`, you can use the `dbed_analyzer` command to obtain tablespace-to-physical disk mapping information for all the datafiles in a specified database. The command also provides information about the amount of disk space being used by a tablespace. Because the `dbed_analyzer` command output can be long, it is written to a file for easier viewing.

This command can help you avoid I/O contention. For example, you can use the information to avoid backing up two tablespaces that share the same physical disk.

Both storage object statistics and the storage structure are displayed in the VERITAS Storage Foundation *for Oracle* GUI.

For more information, see “Using Storage Mapping” on page 125.

VERITAS Database FlashSnap

VERITAS Database FlashSnap is a feature included with VERITAS Storage Foundation *for Oracle* Enterprise Edition. It is also a separately licensed option available with VERITAS Storage Foundation *for Oracle* Standard Edition.

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.

VERITAS Storage Foundation *for Oracle* provides three commands that can be executed by the database administrator and do not require `root` privileges:

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

These commands let 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 according to the guidelines in “Preparing Hosts and Storage for Database FlashSnap” on page 204.

VERITAS Extension for Oracle Disk Manager

VERITAS Extension for Oracle Disk Manager is a custom storage interface defined by Oracle Corporation beginning with Oracle9*i*. Oracle Disk Manager allows Oracle9*i* to exploit system bandwidth through an improved Application Programming Interface (API) that contains advanced kernel support for file I/O. Oracle Disk Manager reduces administration overhead by providing enhanced support for the Oracle Managed Files (OMF) infrastructure also introduced beginning with Oracle9*i*. Combining VERITAS Extension for Oracle Disk Manager with Oracle9*i* offers better database throughput for I/O intensive workloads due to specialized I/O features that greatly improve the I/O system call profile for such key Oracle server processes as the Log Writer and Database Writers.

With VERITAS Extension for Oracle Disk Manager, Oracle9*i* is able to use the same system calls for datafiles stored in the VERITAS File System as they do with raw partitions. Oracle Disk Manager files look just like ordinary file system files and can be handled as such. Care is given within Oracle Disk Manager to ensure files are created with contiguous disk blocks automatically for improved sequential file access performance. Oracle Disk Manager files can be backed up and recovered through VERITAS NetBackup, Oracle Recovery Manager (RMAN), or other backup software. See "Using VERITAS Extension for Oracle Disk Manager" on page 111 for more information.

VERITAS VxDBA Menu Utility

The VxDBA menu utility helps you manage the storage used by databases. You can use VxDBA to:

- Display database, tablespace, datafile, and file system information and manage the database state
- Create, display, mount, unmount, and remove Storage Checkpoints
- Roll back databases, tablespaces, or datafiles to Storage Checkpoints
- Collect and display statistics on file system and Oracle space usage
- Monitor file system and Oracle tablespace and datafile space usage and automatically grow the file system as needed
- Examine volumes used by the file systems and overall system configuration
- Start and stop database instances

For more detailed information about the VxDBA menu-driven utility, see "Using the VxDBA Utility" on page 333.

VERITAS Storage Foundation for Oracle Graphical User Interface

An alternative to the command line interface, the VERITAS Storage Foundation *for Oracle* GUI allows you to manage the storage used by databases. You can use the GUI to:

- Display database, tablespace, datafile, and file system information and manage the database state.
- Create, display, mount, unmount, and remove Storage Checkpoints.
- Roll back databases, tablespaces, or datafiles to Storage Checkpoints.
- Collect and display statistics on file system and Oracle space usage.
- Collect and display storage object I/O statistics and the storage structure.
- Monitor file system and Oracle tablespace and datafile space usage and automatically grow the file system as needed.
- Examine volumes used by the file systems and overall system configuration.
- Start, stop, or duplicate database instances. You can duplicate a database using Storage Checkpoints or Database FlashSnap.
- Create or shut down a clone database.
- Create, modify, validate, or remove snapplans.
- Create, resynchronize, or reverse resynchronize a snapshot database.

Click the Oracle icon on the object tree to expand the tree view. Every database instance running on the server can be displayed on the screen. Each database instance is represented by an internal Oracle object and identified by its unique `ORACLE_SID`.

For more detailed information about the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

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 *for Oracle*, can be purchased separately.

Block-Level Incremental Backup

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 VERITAS Storage Foundation *for Oracle*. 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, or 6 layout, and you must have a valid VERITAS Storage Foundation license. Use the `fstyp -v device` command to determine the layout version of your file system. See "Upgrading From Earlier VxFS Version Layouts" on page 150 and 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 *VERITAS NetBackup for Oracle Advanced BLI Agent System Administrator's Guide*. To shorten the amount of time it takes to back up, you can rely on storage topology information. See "Using Storage Mapping" on page 125 for more information.

VERITAS Storage Foundation/High Availability for Oracle (Optional)

VERITAS Storage Foundation/High Availability (HA) (VCS) *for Oracle* lets database administrators integrate multiple servers into high availability database configurations that can significantly reduce the down time of Oracle databases caused by a system hardware or software failure.

In addition to the VERITAS products included in the base VERITAS Storage Foundation *for Oracle,* VERITAS Storage Foundation/HA *for Oracle* incorporates the following products:

- VERITAS Cluster Server™ (VCS) *for Oracle*
- VERITAS Cluster Server (VCS) Enterprise Agent *for Oracle*

Note VERITAS Storage Foundation/HA (VCS) *for Oracle* is available only for the Enterprise Edition.

Setting Up Databases 2

This chapter describes how to use VERITAS Volume Manager and VERITAS File System to set up optimal system configurations for Oracle Databases.

Topics covered in this chapter include:

Setting Up a New Database

VERITAS Storage Foundation *for Oracle* contains a combination of performance, management, and high availability features. This section provides an overview of the steps to follow if you are using VERITAS Storage Foundation *for Oracle* to set up a new database.

▼ To set up a new database using VERITAS Storage Foundation for Oracle

1. Determine the number and sizes of file systems you need for the database you want to create. For detailed information, see the *VERITAS File System Administrator's Guide*.

2. Create volumes to meet your file system needs. You can use disk mirroring as a safeguard against disk failures and striping for better performance. For more information see the "Selecting a Volume Layout" on page 41 and "Creating a Volume" on page 43. For information about performance tuning to improve database performance, see "Tuning for Performance" on page 407.

   If you plan to create volume snapshots for the database and use them on either the same host or a secondary one, ensure that your volume layout is consistent with Database FlashSnap requirements. For information about these requirements, see "Using Database FlashSnap for Backup and Off-Host Processing" on page 197.

3. Create the VxFS file systems you need on the volumes. See "File System Creation Guidelines" on page 49 and "Creating a VxFS File System" on page 50.

4. Install and configure your database. For best OLTP performance, use Quick I/O for Oracle8*i* and Oracle Disk Manager (ODM) for Oracle9*i* and later. You must create Quick I/O files before creating the tablespaces. For information on creating and using Quick I/O files, and converting existing files to Quick I/O, see "Using VERITAS Quick I/O" on page 67.

   If you would like the ability to view detailed storage stack topology information to ensure your storage stack configuration is optimized for the database, configure and use Storage Mapping. For more information, see "Using Storage Mapping" on page 125.

   If you are already running your database on VxVM and VxFS but are not using Quick I/O or ODM, you will need to follow these steps to convert you existing files to use Quick I/O and ODM:

   a. If you want to use Database FlashSnap for off-host processing after converting your database files to use Quick I/O and ODM and your volume layout is inconsistent with Database FlashSnap requirements, you will need to "relayout" your volume manager configuration after your database files have been converted. See the *VERITAS Volume Manager's Administrator's Guide* for detailed information about VxVM's online relayout feature.

b. If you are using Oracle8*i*, convert all database files to Quick I/O files. See "Converting Oracle Files to Quick I/O Files" on page 78. For Oracle9*i*, you simply need to link the ODM library. See "Setting Up VERITAS Extension for Oracle Disk Manager" on page 118.

If you are not currently running on VxVM and VxFS, see the *VERITAS Storage Foundation for Oracle Installation Guide* and "Converting Existing Database Configurations to VxFS" on page 147 for information about how to convert your existing database storage configuration to VERITAS Storage Foundation *for Oracle*.

5. For backup and recovery on the same host, you can use the Storage Checkpoint facility to create file system snapshots of the database. 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. For more information about creating and managing Storage Checkpoints, see "Using Storage Checkpoints and Storage Rollback" on page 155.

For off-host processing or backup, you can use the Database FlashSnap feature to create a volume snapshot for the database. Database FlashSnap lets you capture an online image of an actively changing database at a given instant, known as a snapshot. You can perform backups and off-host processing tasks on snapshots while providing continuous availability of your critical data.

Setting Up a Disk Group

Before creating file systems for a database, set up a disk group for each database. A disk group lets you group disks, volumes, file systems, and files that are relevant to a single database into a logical collection for easy administration. Because you can move a disk group and its components as a unit from one machine to another, you can move an entire database when all the configuration objects of the database are in one disk group. This capability is useful in a failover situation.

Disk Group Configuration Guidelines

Follow these guidelines when setting up disk groups for use with databases:

- Only disks that are online and do not already belong to a disk group can be used to create a new disk group.
- Create one disk group for each database.
- The disk group name must be unique. Name each disk group using the Oracle database instance name specified by the environment variable `$ORACLE_SID` and a `dg` suffix. The `dg` suffix helps identify the object as a disk group. Also, each disk name must be unique within the disk group.
- Never create database files using file systems or volumes that are not in the same disk group.

See “Tuning for Performance” on page 407 for more information.

Note In earlier releases of VERITAS Volume Manager, a system installed with VxVM was configured with a default disk group, `rootdg`, that had to contain at least one disk. VxVM can now function without any disk group having been configured. Only when the first disk is placed under VxVM control must a disk group be configured.

Note Most VxVM commands require superuser or equivalent privileges.

Creating a Disk Group for a Database

You can use the vxdg command or the graphical user interface (GUI) to create a new disk group. A disk group must contain at least one disk at the time it is created. You also have the option to create a shared disk group for use in a cluster environment.

Disks must be placed in disk groups before they can be used by VxVM. You can create disk groups to organize your disks into logical sets of disks.

When you place a disk under VxVM control, the disk is initialized. Initialization destroys any existing data on the disk.

Prerequisites

- Only disks that are online and do not belong to a disk group can be used to create a disk group.
- The disk group name must be unique in the host or cluster.
- Creating a disk group requires at least one disk.

Usage Notes

- For information on the vxdg command, see the vxdg(1M) manual page.
- In the GUI, if multiple disks are specified in the **Disk Device(s)** field and only one disk name is specified in the **Disk Name(s)** field, VxVM appends numbers to the disk name so that each disk name is unique within its disk group.
- New disks must be placed under VxVM control and then added to a dynamic disk group before they can be used for volumes. The **Actions > Rescan** command performs these tasks to prepare new disks for VxVM use.
- When you place a disk under VxVM control, the disk is either encapsulated or initialized. Encapsulation preserves any existing data on the disk in volumes. Initialization destroys any existing data on the disk.
- If you place the root disk under VxVM control, you must encapsulate the disk. If you want to create an alternate boot disk, you can mirror the encapsulated root disk.
- Disks are automatically assigned a default name. Once a disk is under VxVM control, you can select **Actions > Rename Disk** in the GUI to change the disk name.

Usage Notes

- The disk group name must be unique.
- The new disk group must contain at least one disk.
- Only disks that are online and do not belong to another disk group can be used to create a disk group.

▼ To create a new disk group using the command line

Use the `vxdg` command as follows:

```
# /opt/VRTS/bin/vxdg init PRODddg PRODdg01=c1t1d1s2
```

Example

To create a disk group named `PRODdg` on a raw disk partition `c1t1d0s2`, where the disk name `PRODdg01` references the disk within the disk group:

```
# /opt/VRTS/bin/vxdg init PRODdg PRODdg01=Disk_0
```

▼ To create a new disk group using the GUI

1. Select the **Disk Groups** folder.
2. Select **Actions > New Disk Group**.

   The **New Disk Group** wizard welcome appears.
3. You can suppress the **Welcome to Create New Disk Group** page by selecting the **Do not show this page next time** checkbox.
4. Click **Next** to continue. The **New Disk Group** wizard appears. Enter the following information:

Group name:	Enter a new disk group name in the **Group Name** field.
Create cluster group	If the cluster feature is available, you can select the **Create cluster group** checkbox if the new disk group is to be used with clusters.
Available disks: Selected disks:	Select which disks (from **Available disks:**) you want to include in the group. Make sure the disks you want to include are in the right pane of the window (**Selected disks:**). (Normally, you would add all the disks you want in the group at this point. However, you can always add more disks later with the **Add Disk to Disk Group** procedure.)

Disk names:	Type in the names of any disks to be added that do not appear under **Selected disks:**.

5. When you have provided all the necessary information in the dialog box, click **Next**.

6. Click **Next** to add the disks.

7. The **Organization principle** screen now appears.

8. Click None, if you do not want to organize the disk group. Otherwise, click any of the other organization categories under **Organize Disk Group By** to create an ISP disk group (refer to Organizing Disk Groups).

9. If you choose an organization principle other than None, the **Specify Pool Names** screen appears. Specify the names for the Data pool and the Clone pool and click **Next**. You must specify the pool names in order to create the first storage pool for an ISP disk group.

10. The next screen confirms the disks you have selected. Click **Finish** to continue if you are satisfied with the disk selection. If you are not satisfied, you can click the **<Back** button to go back to the previous screen in order to modify your selections.

11. After clicking **Finish**, the new disk group will then appear under the Disks Groups node.

12. If the disks are not empty, indicate whether the disk should be initialized or encapsulated. If you initialize the disk, any existing data on the disk will be destroyed. If you encapsulate the disk, any existing data will be preserved in volumes.

    If you choose to encapsulate the root disk, the disk needs to be rebooted before encapsulation can take effect. You have the option of rebooting the system now or later.

13. Click **Next** to add the disks to the new disk group.

    The **Organization Principle** window appears.

14. Click **None**, if you do not want to organize the disk group. Otherwise, click **Organize Disk Group By** to create an ISP disk group.

15. If you choose an organization principle other than **None**, the **Specify Pool Names** window appears. Specify the names for the **Data pool** and the **Clone pool** and click **Next**. You must specify the pool names in order to create the first storage pool for an ISP disk group.

16. If you chose to encapsulate the root disk, reboot the system.

Adding Disks to a Disk Group

When a disk group is first created, it contains only a single disk. You may need to add more disks to the disk group. This section describes how to add disks to a disk group using the `vxdg` command or the GUI. If you have many disks to add to the disk group, it is easier to use the `vxdg` command.

Usage Notes

- By default, the GUI assigns default disk names by appending numbers to the disk group name so that each disk name is unique within its disk group. After the disk is under VxVM control, you can rename it.
- When you place a disk under VxVM control, the disk is either encapsulated or initialized. Encapsulation preserves any existing data on the disk in volumes. Initialization destroys any existing data on the disk.
- If you place the boot disk under VxVM control, you must encapsulate it. If you want to create an alternate boot disk, you can mirror the encapsulated boot disk.
- Disk encapsulation requires a system reboot.
- Disks cannot be added to deported disk groups.
- Disks must be under VxVM control and in a disk group before they can be used to create volumes.
- Disks must be online before they can be added to a disk group.
- Disks that already belong to a disk group cannot be added to another disk group.

▼ To add disks to a disk group using the command line

Use the `vxdg` command as follows:

```
# /opt/VRTS/bin/vxdg -g disk_group adddisk [disk_name=disk_device]
```

Example

To add disks named `PRODdg02`, `PRODdg03`, and `PRODdg04` to the disk group `PRODdg`:

```
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg02=c1t2d0s2
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg03=c1t3d0s2
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg04=c1t4d0s2
```

▼ To add a disk to a disk group using the GUI

1. Select the uninitialized disk to be placed under VxVM control.
2. Click the Disk group to which you wish to add a disk.
3. Choose **Actions > Add Disk to Disk Group**. The **Add Disk to Disk Group** wizard appears. Click **Next** to continue.
4. Complete the **Add Disk to Dynamic Group** wizard as follows:

Disk group name:	From the pull-down menu, select the group you want to add the disk to. To add the disk to a new disk group, click the **New disk group** button and enter the name of the new disk group in the dialog box.
Available disks: Selected disks:	Move the disk to be added from **Available disks** to **Selected disks**.

5. When you have provided all the necessary information in the dialog box, click **Next**.
6. When the confirmation window appears, click **Yes** to confirm your selection.
7. Click **Finish** to add the disk to the selected disk group.

Selecting a Volume Layout

VERITAS Volume Manager offers a variety of layouts that allow you to configure your database to meet performance and availability requirements. The proper selection of volume layouts provides optimal performance for the database workload.

An important factor in database performance is the tablespace placement on the disks. Disk I/O is one of the most important determining factors of your database's performance. Having a balanced I/O load usually means optimal performance. Designing a disk layout for the database objects to achieve balanced I/O is a crucial step in configuring a database.

When deciding where to place tablespaces, it is often difficult to anticipate future usage patterns. VxVM provides flexibility in configuring storage for the initial database set up and for continual database performance improvement as needs change. VxVM can split volumes across multiple drives to provide a finer level of granularity in data placement. By using striped volumes, I/O can be balanced across multiple disk drives. For most databases, ensuring that different database files and tablespaces are distributed across the available disks may be sufficient.

Striping also helps sequential table scan performance. When a table is striped across multiple devices, a high transfer bandwidth can be achieved by setting the Oracle parameter `DB_FILE_MULTIBLOCK_READ_COUNT` to a multiple of *full stripe size* divided by `DB_BLOCK_SIZE`. See "Tuning for Performance" on page 407 for more information.

Choosing Appropriate Stripe Unit Sizes

When creating a striped volume, you need to decide the number of columns to form a striped volume and the stripe unit size. You also need to decide how to stripe the volume. You may stripe a volume across multiple disk drives on the same controller or across multiple disks on multiple controllers. By striping across multiple controllers, disk I/O can be balanced across multiple I/O channels. The decision is based on the disk and controller bandwidth and the database workload. In general, for most OLTP databases, use the default stripe unit size of 64 K or smaller for striped volumes and 16 K for RAID-5 volumes.

Choosing Between Mirroring and RAID-5

VxVM provides two volume configuration strategies for data redundancy: mirroring and RAID-5. Both strategies allow continuous access to data in the event of disk failure. For most database configurations, we recommend using mirrored, striped volumes. If hardware cost is a significant concern, but having higher data availability is still important, use RAID-5 volumes.

RAID-5 configurations have certain performance implications you must consider. Writes to RAID-5 volumes require parity-bit recalculation, which adds significant I/O and CPU overhead. This overhead can cause considerable performance penalties in online transaction processing (OLTP) workloads. If the database has a high read ratio, however, RAID-5 performance is similar to that of a striped volume.

Volume Configuration Guidelines

Follow these guidelines when selecting volume layouts:

- Put the database log files on a file system created on a striped and mirrored (RAID-0+1) volume separate from the index or data tablespaces. Stripe multiple devices to create larger volumes if needed. Use mirroring to improve reliability. Do not use VxVM RAID-5 for redo logs.
- When normal system availability is acceptable, put the tablespaces on file systems created on striped volumes for most OLTP workloads.
- Create striped volumes across at least four disks. Try to stripe across disk controllers. For sequential scans, do not stripe across too many disks or controllers. The single thread that processes sequential scans may not be able to keep up with the disk speed.
- For most workloads, use the default 64 K stripe-unit size for striped volumes and 16 K for RAID-5 volumes.
- When system availability is critical, use mirroring for most write-intensive OLTP workloads. Turn on Dirty Region Logging (DRL) to allow fast volume resynchronization in the event of a system crash.
- When system availability is critical, use RAID-5 for read-intensive OLTP workloads to improve database performance and availability. Use RAID-5 logs to allow fast volume resynchronization in the event of a system crash.
- For most decision support system (DSS) workloads, where sequential scans are common, experiment with different striping strategies and stripe-unit sizes. Put the most frequently accessed tables or tables that are accessed together on separate striped volumes to improve the bandwidth of data transfer.

Creating a Volume

VERITAS Volume Manager uses logical volumes to organize and manage disk space. A volume is made up of portions of one or more physical disks, so it does not have the limitations of a physical disk.

For databases where the data storage needs to be resilient and the data layout needs to be optimized for maximum performance, we recommend using VxVM. The striping and mirroring capabilities offered by a volume manager will help you achieve your manageability, availability, and performance goals.

After you decide on a volume layout, you can use the `vxassist` command or the GUI to create the volume.

If you choose to use the GUI, the *VERITAS Volume Manager Administrator's Guide* provides a detailed comparison of the layout choices and more detailed procedures on creating each volume layout type. The GUI gives you the option of placing a file system on the new volume or mirroring during volume creation.

Usage Notes

- Creating a volume requires a disk group name, volume name, volume size, and volume layout. You will also need to know subdisk names if you are creating a striped volume.
- Striped or mirrored volumes require at least two disks.
- Striped pro and concatenated pro volumes are mirrored by default, so a striped pro volume requires more disks than an unmirrored striped volume and a concatenated pro volume requires more disks than an unmirrored concatenated volume.
- You cannot use a striped pro or concatenated pro volume for a `root` or `swap` volume.
- A RAID-5 volume requires at least three disks. If RAID-5 logging is enabled, a RAID-5 volume requires at least four disks.
- RAID-5 mirroring is not supported.

▼ To create a volume using the command line

Use the `vxassist` command as follows:

```
# /opt/VRTS/bin/vxassist -g disk_group make volume_name size \
layout=layout_type
```

Example

To create a 1 GB mirrored volume called `db01` on the `PRODdg` disk group:

```
# /opt/VRTS/bin/vxassist -g PRODdg make db01 1g layout=mirror
```

▼ To create a volume using the GUI

1. Under **Disk Groups** under the selected host in the left pane, click on the disk group to be used to create the volume.

2. Choose **Actions** > **New Volume**. You are asked if you want Volume Manager to select the disks to use or if you want to select them manually.

3. Select **Manually select disks to use with this volume**, then click **Next**.

4. The **Select Disks to use for Volume** screen appears. You can now:

   - Select disks to be used by VxVM when creating the volume. (Move to the **Included:** area.)
   - Select disks not to be used by VxVM when creating the volume. (Move to **Excluded:** area.)
   - Specify that the volume is to be mirrored across (**Mirror Across:**) or striped across (**Stripe Across:**) controllers, trays, targets, or enclosures.

Note For ISP volumes, the layouts **Mirror Across:** and **Stripe Across:** cannot be specified.

   - Specify ordered allocation (**Ordered**). Ordered allocation uses the specified storage to first concatenate disks, then to form columns, and finally to form mirrors.

5. Click **Next**. The **Select the attributes for this volume** screen appears.

6. Type in the Volume Name. This is a Volume Manager-specific name that is used in some Volume Manager commands. It is different from the volume label for the file system.

7. Specify the volume size, or select **Maxsize**.

   As you have already selected a disk, a size is shown in the volume size box that represents the maximum concatenated (simple or spanned) volume size on the disk. If you then click the **Maxsize** button, a new number appears in the volume size box that represents the maximum size for a spanned volume that spans all the disks in the group. You can also click on another volume layout and then click the **Maxsize** button to get the maximum size for that layout that involves unallocated space on all disks in the group.

If you choose Striped or RAID-5, Number of Columns and Stripe Unit Size need to have an entry. Defaults are provided.

8. Select the required layout:
   - Concatenated (Concatenated and Concatenated Mirrored)
   - Striped (Striped and Striped Mirrored)
   - RAID-5
   - Mirrored

9. Select the required options

Option	Notes
Mirror Info:	◆ To mirror the volume, select **Mirrored**. In the **Total Number of Mirrors** field, enter the total number of mirrors for the volume. **Note** Concatenated mirrored volumes are mirrored by default.
Initialize zero:	◆ To clear the volume before enabling it for general use, select **Initialize Zero**.
No layered volumes:	◆ To prevent the creation of a layered volume, select **No Layered Volumes**. In cases where a layered volume layout is appropriate, VxVM can create a layered volume when a non-layered layout is specified. This option ensures that the volume has a non-layered layout. If a layered (Pro) layout is selected, this option is ignored.
Enable FastResync Enable logging	◆ If **Enable FastResync** is checked, then **Enable Logging** is an option only if the volume is mirrored. (The reason it is only enabled when **Enable FastResync** is checked is to ensure that a new style DRL is created.)

10. After you provide all the necessary information in the dialog box, click **Next**. You are now asked if you want to create a file system.

11. If you want to create a file system, select **Create a File System** and follow the instructions. If you do not want to create a file system, select **No File System**, then click **Next**.

12. A summary of your selections appears. Click **Back** to make changes; otherwise, click **Finish**.

Creating a Volume Set

Volume Sets enable the use of the Multi-Volume Support feature with VERITAS File System (VxFS). It is also possible to use the VERITAS Enterprise Administrator (VEA) to create and administer volumes sets. For more information, see the VEA online help.

Note For details regarding usage of the `vxvset` command, see the `vxvset(1M)` manual page.

Note Most VxVM commands require superuser or equivalent privileges.

Usage Notes

- A maximum of 256 volumes may be configured in a volume set.
- Only VERITAS File System is supported on a volume set.
- The first volume in a volume set must be larger than 20MB.
- Raw I/O from and to a volume set is not supported.
- Volume sets can be used instead of volumes with the following `vxsnap` operations on instant snapshots: `addmir`, `dis`, `make`, `prepare`, `reattach`, `refresh`, `restore`, `rmmir`, `split`, `syncpause`, `syncresume`, `syncstart`, `syncstop`, `syncwait`, and `unprepare`. The third-mirror break-off usage model for full-sized instant snapshots is supported for volume sets provided that sufficient plexes exist for each volume in the volume set. Refer to the *VERITAS Volume Manager 4.1 Administrator's Guide* for more information on instant snapshots.
- A full-sized snapshot of a volume set must itself be a volume set with the same number of volumes and the same volume index numbers as the parent. The corresponding volumes in the parent and snapshot volume sets are also subject to the same restrictions as apply between standalone volumes and their snapshots.

▼ To create a volume set for use by VERITAS File System (VxFS), using the CLI

Use the following command:

```
# /opt/VRTS/bin/vxvset [-g diskgroup] -t vxfs make volset volume
```

Here `volset` is the name of the volume set, and `volume` is the name of the first volume in the volume set. The `-t` option defines the content handler subdirectory for the application that is to be used with the volume. This subdirectory contains utilities that an application uses to operate on the volume set. The operation of these utilities is determined by the requirements of the application and not by VxVM.

For example, to create a volume set named myvset that contains the volume vol1, in the disk group mydg, you would use the following command:

```
# /opt/VRTS/bin/vxvset -g mydg -t vxfs make myvset vol1
```

Note For further details on Creating a Volume Set, please refer to the *VERITAS Volume Manager Administrator's Guide*.

Adding a Volume to a Volume Set

Having created a volume set containing a single volume, you can use the following command to add further volumes to the volume set:

```
# /opt/VRTS/bin/vxvset [-g diskgroup] [-f] addvol volset volume
```

For example, to add the volume `vol2`, to the volume set `myvset`, use the following command:

```
# /opt/VRTS/bin/vxvset -g mydg addvol myvset vol2
```

Caution The `-f` (force) option must be specified if the volume being added, or any volume in the volume set, is either a snapshot or the parent of a snapshot. Using this option can potentially cause inconsistencies in a snapshot hierarchy if any of the volumes involved in the operation is already in a snapshot chain.

Note For further details on volumes and volume sets, refer to the *VERITAS Volume Manager Administrator's Guide*.

File System Creation Guidelines

Follow these guidelines when creating VxFS file systems:

- To take advantage of Quick I/O, online administration, fast recovery of the VxFS file system, and superior reliability features, select `vxfs` as the file system type.
- Specify the maximum block size and log size when creating file systems for databases.

Note Choose a file system block size that matches or is a multiple of the block size of your Oracle database (`db_block_size`).

It is possible to have a file system block size that is smaller than the database block size because the database block-size limit can be bigger than the file system block size. It is fine if the file system block size is smaller than the database block size because VxFS will not perform multiple I/O operations for each database I/O operation. VxFS is capable of performing I/Os with multiple blocks. For example, if your database block size is 32K and your file system block size is 8k, VxFS can put four 8K blocks together to perform one 32K database I/O operation.

When creating the file system, set the number of file system blocks in the intent log so that the log size is 16MB. For example, if the file system block size is 8K (that is, 8192), it will take 2000 blocks to make a 16MB log (2000 x 8192 = ~16MB). If the file system block size is 4K (that is, 4096), then twice as many blocks as in the 8K case would need to be allocated (4000 in this example).

- Never disable the intent logging feature of the file system.
- For redo logs, create a single file system using a simple (and mirrored, if necessary) volume. Put the other tablespaces and database files on separate file systems created on striped, striped and mirrored, or RAID-5 volumes.
- When using the command line, use the mount points to name the underlying volumes. For example, if a file system named `/db01` is to be created on a mirrored volume, name the volume `db01` and the mirrors `db01-01` and `db01-02` to relate to the configuration objects. If you are using the `vxassist` command or the GUI, this is transparent.

Creating a VxFS File System

Always specify `vxfs` as the file system type to take advantage of Quick I/O, Storage Rollback, online administration, fast recovery of the VxFS file system, and superior reliability features.

The GUI lets you add a file system on a new volume during the volume creation process. See the *VERITAS Volume Manager User's Guide - VERITAS Enterprise Administrator* for a detailed comparison of the layout choices and more detailed procedures on creating each volume layout type.

Usage Notes

- See the `mkfs`(1M) and `mkfs_vxfs`(1M) manual pages for more information about the options and variables available for use with the `mkfs` command.
- See the `mount`(1M) and `mount_vxfs`(1M) manual pages for more information about mount settings.
- In the GUI, you must specify a file system mount point if the file system is to be mounted at startup.
- If you select the **Add to file system table** checkbox in the GUI, the file system table file will be automatically updated when the file system is mounted.
- When specifying a mount point in the GUI, you must use an absolute path name (that is, it must begin with `/`).

▼ To create a VxFS file system on an existing volume using the command line

Use the `mkfs` command to create a VxFS file system on an existing volume as follows:

```
# /usr/sbin/mkfs -F vxfs [generic_options] \
[-o specific_options] special [size]
```

where:

- `vxfs` is the file system type
- *`generic_options`* are the options common to most file systems
- *`specific_options`* are options specific to the VxFS file system
- *`special`* is the full path name of the raw character device or VxVM volume on which to create the file system (for example, `/dev/vx/rdsk/PRODdg/db01`)
- *`size`* is the size of the new file system (optional)

If you do not specify *`size`*, the file system will be as large as the underlying volume or device partition.

Example

To create a VxFS file system that supports files larger than 2GB on the newly created db01 volume:

```
# /usr/sbin/mkfs -F vxfs -o largefiles,bsize=8192,logsize=2000 \
/dev/vx/rdsk/PRODdg/db01
```

The `-o largefiles` specific option allows you to create files larger than 2 GB.

Note Because *size* is not specified in this example, the size of the file system will be calculated automatically to be the same size as the volume on which the file system is created.

The `mkfs` command displays output similar to the following:

```
version 6 layout
20480 sectors, 10240 blocks of size 1024, log size 1024 blocks
```

You can now mount the newly created file system. See "Mounting a File System" on page 54.

▼ To add a new file system to an existing volume using the GUI

1. Select the volume to contain the file system.

2. Choose **Actions** > **File System** > **New File System**.

3. Complete the **New File System** dialog box as follows:

File System Type:	Select the file system type from the pull-down menu.
Create Options:	◆ Specify the allocation size and block size if you do not want to use the default. ◆ To specify whether large files (files greater than or equal to 2 GB) will be supported, click **New File System Details**. Select either **largefiles** or **nolargefiles**.
Compress (checkbox):	◆ If your platform supports file compression, then you can select **Compress** to compress the files on your file system

Mount Options:	◆ Enter the mount point for the file system, if you want the file system mounted at system startup. ◆ Select the **Create mount point** checkbox if you want the system to create the mount point if it does not already exist. ◆ Select the **Read only** and **Honor setuid** checkboxes, as required. ◆ Select the **Add to file system table** and **Mount at boot** checkboxes to update the system table file and mount the file system at system startup. ◆ To update the system table file and *not* mount the file system at system startup, select **Add to file sytem table** checkbox and leave the **Mount at Boot** checkbox unselected. ◆ To specify `mount` options, click **Mount File System Details** and specify the appropriate options in the Mount Details dialog box.

4. After you provide all necessary information in the dialog box, click **OK**.

Support for Large File Systems and Large Files

In conjunction with VxVM, VxFS can support file systems up to 8 exabytes in size. For large database configurations, this eliminates the need to use multiple file systems because of the size limitations of the underlying physical devices.

Changes implemented with the VxFS Version 6 disk layout have greatly expanded file system scalability, including support for large files. You can create or mount file systems with or without large files by specifying either the `largefiles` or `nolargefiles` option in `mkfs` or `mount` commands. If you specify the `nolargefiles` option, a file system cannot contain files 2 GB or larger.

Usage Notes

- See the `mount_vxfs`(1M) and `mkfs_vxfs`(1M) manual pages for detailed information on mounting and creating file systems.
- See the `fsadm_vxfs`(1M) manual pages for detailed information about large files.

▼ To enable large files on a file system that was created without the largefiles option

Use the `fsadm` command as follows:

```
# /opt/VRTS/bin/fsadm -F vxfs -o largefiles /mount_point
```

Caution Make sure the applications and tools you use can handle large files before enabling the large file capability. Applications and system administration utilities can experience problems if they are not large file aware.

Multi-volume Support

The multi-volume support feature enabled by Version 6 disk layout allows several volumes to be represented by a single logical object, known as a *volume set*. The `vxvset` command can be used to create and administer volume sets in VERITAS Volume Manager.

VxFS's multi-volume support feature can be used with volume sets. There are two VxFS commands associated with multi-volume support:

- `fsapadm` - VxFS allocation policy administration utility
- `fsvoladm` - VxFS device administration utility

For more information about volume sets and multi-volume support, see the *VERITAS File System Administrator's Guide*.

Mounting a File System

After creating a VxFS file system, mount the file system using the mount command. By default, the mount command tries to enable Quick I/O. If Quick I/O is not installed or licensed, no error messages are displayed unless you explicitly specify the -o qio mount option. If necessary, you can turn the Quick I/O option off at mount time or you can remount the file system with the -o noqio option.

Prerequisites

- A file system must exist in order to be mounted.
- DBAs should log in as the Oracle DBA user.

Usage Notes

- See the mount_vxfs(1M) manual page for more information about mount settings.
- See the mount(1M) manual page for more information about generic mount options.
- If you use the GUI, the file system table file is automatically updated.
- The mount point must be an absolute path name (that is, it must begin with /).
- If you use the GUI, the path specified for the mount point will be created if it does not already exist.

▼ To mount a file system using the command line

Use the mount command as follows:

```
# /usr/sbin/mount -F vxfs [generic_options] [-r] \
[-o specific_options] special /mount_point
```

where:

- *generic_options* are the options common to most file systems
- -r mounts the file system as read only
- *specific_options* are options specific to the VxFS file system
- *special* is a block special device
- */mount_point* is the directory where the file system will be mounted

Example

To mount a file system named /db01 that supports large files on volume /dev/vx/dsk/PRODdg/db01:

```
# mkdir /db01
# chown oracle:dba /db01
# /usr/sbin/mount -F vxfs -o largefiles /dev/vx/dsk \
  /PRODdg/db01/db01
```

If you would like /db01 to be mounted automatically after rebooting, add an entry for it in /etc/vfstab as follows:

```
/dev/vx/dsk/PRODdg/db01 /dev/vx/rdsk/PRODdg/db01 /db01 \
vxfs 2 yes largefiles,qio
```

If you do not need to use Quick I/O files, set noqio instead of qio as one of the options.

▼ To mount a file system on an existing volume using the GUI

1. Select the volume that contains the file system to be mounted.

2. Choose **Actions** > **File System** > **Mount File System**.

3. Complete the **Mount File System** dialog box (see step 3 on page 56 in the procedure below).

4. After you have provided all the necessary information in the dialog box, click **OK**.

▼ To mount any file system using the GUI

1. Select the file system to be mounted.

2. Choose **Actions** > **Mount File System**.

3. Complete the **Mount File System** dialog box as follows:

FS Type:	Select the file system type.
Mount Options:	◆ If you want the system to use the mount options defined in the system table, check **Mount using options in the file system table**. ◆ Enter the mount point for the file system, if you want the file system mounted at system startup. ◆ Select the **Create Mount Point** checkbox if you want the system to create the mount point if it does not already exist. ◆ Select the **Read Only** and **Honor setuid** checkboxes, as required. ◆ To specify mount options, click **Mount File System Details** and specify the appropriate options in the **Mount Details** dialog box.

4. After you have provided all the necessary information in the dialog box, click **OK**.

Unmounting a File System

If you no longer need to access the data in a file system, you can unmount the file system using the `umount` command.

Prerequisites

- A file system must exist and be mounted in order to be unmounted.

Usage Notes

- See the `umount`(1M) manual page for more information on mounting file systems.
- You cannot unmount a file system that is in use.

▼ To unmount a file system using the command line

1. Use the `fuser` command to make sure that the file system is not being used:

```
# fuser -c /mount_point
```

where the `-c` option provides information on file system mount points and any files within mounted file systems.

Note If the file system is being used and you need to unmount it, use the `fuser -ck` command. See the `fuser`(1M) man page for more information.

2. Unmount the file system using the `umount` command:

```
# umount special
```

or

```
# umount /mount_point
```

or

```
# umount -f /mount_point
```

where:

- *`special`* is a block special device
- *`/mount_point`* is the location where the file system is mounted
- `-f` forcibly unmounts the mount point

Example

To verify that the file system `/db01` is not in use and then unmount the file system:

```
# fuser -c /db01
/db01:
# umount /db01
```

▼ To unmount a file system on a volume using the GUI

1. Select the volume containing the file system to be unmounted.

2. Choose **Actions > File System > Unmount File System**.

3. Click **Yes** in the **Unmount File System** dialog box to confirm that you want to unmount the file system.

4. If an entry exists for a file system in the file system table, a **Remove File System** dialog appears. Click **Yes** in the **Remove File System** dialog if you want the file system table entry removed.

Understanding Fragmentation

When free resources are initially allocated to files in a VERITAS file system, they are aligned in the most efficient order possible to provide optimal performance. On an active file system, the original order is lost over time as files are created, removed, or resized. As space is allocated and deallocated from files, the available free space becomes broken into fragments. This means that space must be assigned to files in smaller and smaller extents. This process is known as fragmentation. Fragmentation leads to degraded performance and availability. The degree of fragmentation depends on file system usage and activity.

Controlling Fragmentation

Allocation units in VxFS are designed to help minimize and control fragmentation. Over time, however, file systems eventually become fragmented.

VxFS provides online reporting and optimization utilities to enable you to monitor and defragment a mounted file system. These utilities are accessible through the file system administration command, `fsadm`. Using the `fsadm` command, you can track and eliminate fragmentation without interrupting user access to the file system.

Types of Fragmentation

VxFS addresses two types of fragmentation:

- Directory Fragmentation

  As files are created and removed, gaps are left in directory inodes. This is known as directory fragmentation. Directory fragmentation causes directory lookups to become slower.

- Extent Fragmentation

  As files are created and removed, the free extent map for an allocation unit changes from having one large free area to having many smaller free areas. Extent fragmentation occurs when files cannot be allocated in contiguous chunks and more extents must be referenced to access a file. In a case of extreme fragmentation, a file system may have free space that cannot be allocated.

Monitoring Fragmentation

You can monitor fragmentation in VxFS by running reports that describe fragmentation levels. Use the `fsadm` command to run reports on directory fragmentation and extent fragmentation. The `df` command, which reports on file system free space, also provides information useful in monitoring fragmentation.

Use the following commands to report fragmentation information:

- `fsadm -D`, which reports on directory fragmentation.
- `fsadm -E`, which reports on extent fragmentation.
- `/opt/VRTS/bin/df -F vxfs -o s`, which prints the number of free extents of each size.

Defragmenting a File System

You can use the online administration utility `fsadm` to defragment or reorganize file system directories and extents. The `fsadm` utility defragments a file system mounted for read/write access by:

- Removing unused space from directories.
- Making all small files contiguous.
- Consolidating free blocks for file system.

The following options are for use with the `fsadm` utility:

Options

`-d` Reorganizes directories. Directory entries are reordered to place subdirectory entries first, then all other entries in decreasing order of time of last access. The directory is also compacted to remove free space.

`-a` Use in conjunction with the `-d` option to consider files not accessed within the specified number of days as “aged” files. Aged files are moved to the end of the directory. The default is 14 days.

`-e` Reorganizes extents. Files are reorganized to have the minimum number of extents.

`-D -E` Produces reports on directory and extent fragmentation, respectively.

`-v` Specifies verbose mode and reports reorganization activity.

-l Specifies the size of a file that is considered large. The default is 64 blocks.

-t Specifies a maximum length of time to run, in seconds.

-p Specifies a maximum number of passes to run. The default is five.

-s Prints a summary of activity at the end of each pass.

-r Specifies the pathname of the raw device to read to determine file layout and fragmentation. This option is used when *fsadm* cannot determine the raw device.

Usage Notes

- If you specify -d and -e, directory reorganization is always completed first.
- If you use both -D and -E with the -d and -e options, the fragmentation reports are produced both before and after reorganization.
- The -t and -p options control the amount of work performed by fsadm, either in a specified time or by a number of passes. By default, fsadm runs five passes. If both -t and -p are specified, fsadm exits if either of the terminating conditions are reached.

Note You must have superuser (root) privileges to reorganize a file system using the fsadm command.

▼ To defragment a file system using the command line

Run the fsadm command followed by the options specifying the type and amount of defragmentation. Complete the command by specifying the mount point or raw device to identify the file system.

```
# /opt/VRTS/bin/fsadm [-d] [-D] [-e] [-E] [-s] [-v] \
  [-l largesize] [-a days] [-t time] [-p pass_number] \
  [-r rawdev_path] mount_point
```

Refer to the *File System Administrator's Guide* for instructions and information on scheduling defragmentation.

Example

To defragment a file system:

```
# /opt/VRTS/bin/fsadm -d -D /oradata_qiovm
Directory Fragmentation Report
             Dirs        Total      Immed     Immeds    Dirs to    Blocks to
             Searched    Blocks     Dirs      to Add    Reduce     Reduce
  total             5         1         4         0         00


  Directory Fragmentation Report
             Dirs        Total      Immed     Immeds    Dirs to    Blocks to
             Searched    Blocks     Dirs      to Add    Reduce     Reduce
  total             5         1         4         0         00
```

▼ To defragment a file system on a volume using the GUI

1. Select the volume containing the file system to be defragmented.

2. Choose **Actions > File System > Defrag File System**.

3. Select **Yes** in the displayed dialog box.

Resizing a File System

If you need to extend or shrink a VxFS file system, you can use the `fsadm` command.

If a VxFS file system requires more space, you can use this procedure to extend the size of the file system. If a VxFS File System is too large and you need the space elsewhere, you can use this procedure to shrink the file system.

Note If you are using the command line, remember to increase the size of the underlying device or volume before increasing the size of the file system. See the *VERITAS Volume Manager Administrator's Guide* for more information.

Prerequisites

- This task requires a mounted file system.
- You must know either the desired size or the amount of space to add to or subtract from the file system size.

Usage Notes

- See the `format`(1M) and `fsadm_vxfs`(1M) manual pages for more details.

▼ To resize a file system using the command line

Use `fsadm` command as follows:

```
# /opt/VRTS/bin/fsadm -F vxfs [-b newsize] \
  [-r rawdev] /mount_point
```

where:

- *`newsize`* is the size (in sectors) to which the file system will increase or shrink
- *`rawdev`* specifies the name of the raw device if there is no entry in `/etc/vfstab` and `fsadm` cannot determine the raw device
- */`mount_point`* is the location where the file system is mounted

Example

To extend the file system /db01 to 2 GB:

```
# /opt/VRTS/bin/fsadm -F vxfs -b 2g /db01
```

Note See the *VERITAS File System Administrator's Guide* and fsadm_vxfs(1M) manual page for information on how to perform common file system tasks using fsadm.

Resizing a File System and the Underlying Volume

The fsadm command resizes the file system only. If you attempt to use fsadm to make the file system the same size or larger than the underlying volume, the fsadm command will fail. To resize the file system *and* its underlying volume, use the vxresize command instead.

Prerequisites

- You must know the new desired size of the file system.

Usage Notes

- If you use the GUI, the underlying volume is resized when the file system is resized.
- vxresize works with VxFS file systems only.
- When resizing large volumes, vxresize may take a long time to complete.
- Resizing a volume with a usage type other than FSGEN or RAID5 can result in data loss. If such an operation is required, use the -f option to forcibly resize such a volume.
- You cannot resize a volume that contains plexes with different layout types.
- See the vxresize(1M) manual page for more details.

Example

To extend a 1-gigabyte volume, homevol, that contains a VxFS file system, to 10 gigabytes using the spare disks disk10 and disk11, enter:

```
# /etc/vx/bin/vxresize -b -F vxfs -t homevolresize homevol 10g \
disk10 disk11
```

The -b option specifies that this operation runs in the background. Its progress can be monitored by specifying the task tag homevolresize to the vxtask command.

Growing a File System Automatically Using VxDBA Monitoring Agent

You can use the VxDBA Monitoring Agent to monitor file system space, and when the space usage reaches a configured threshold value, a predefined action script automatically grows the file system. See “Managing File System Space” on page 380 for more information.

Using VERITAS Quick I/O 3

VERITAS Quick I/O is a VxFS feature included in VERITAS Storage Foundation *for Oracle* 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 performance degradation typically associated with running databases on file systems or raw devices. This chapter describes how to set up and use Quick I/O.

Note If you are using Oracle*8i*, VERITAS recommends that you use Quick I/O. If you are using Oracle9*i*, we recommend that you use VERITAS Extension for Oracle Disk Manager. For more information, see "Using VERITAS Extension for Oracle Disk Manager" on page 111.

Topics covered in this chapter include:

Understanding Quick I/O

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 for Oracle databases 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

If you are using Oracle8*i*, we recommend using Quick I/O. Although Quick I/O is compatible with Oracle9*i*, VERITAS Extension for Oracle Disk Manager was designed especially for Oracle9*i*. If you are upgrading to Oracle9*i* and would like to migrate from Quick I/O to Oracle Disk Manager, see "Converting Quick I/O Files to Oracle Disk Manager Files" on page 121 for more information.

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

Supporting Kernel Asynchronous I/O

Asynchronous I/O is a form of I/O that performs non-blocking system level reads and writes, allowing the system to handle multiple I/O requests simultaneously. Operating systems such as Solaris provide kernel support for asynchronous I/O on raw devices, but not on regular files. As a result, even if the database server is capable of using asynchronous I/O, it cannot issue asynchronous I/O requests when the database runs on file systems. Lack of asynchronous I/O significantly degrades performance. Quick I/O lets the database server take advantage of kernel-supported asynchronous I/O on file system files accessed using the Quick I/O interface.

Supporting Direct I/O

I/O on files using `read`() and `write`() system calls typically results in data being copied twice: once between user and kernel space, and later between kernel space and disk. In contrast, I/O on raw devices is direct. That is, data is copied directly between user space and disk, saving one level of copying. As with I/O on raw devices, Quick I/O avoids extra copying.

Avoiding Kernel Write Locks

When database I/O is performed using the `write`() system call, each system call acquires and releases a write lock inside the kernel. This lock prevents multiple simultaneous write operations on the same file. Because database systems usually implement their own locking to manage concurrent access to files, per file writer locks unnecessarily serialize I/O operations. Quick I/O bypasses file system per file locking and lets the database server control data access.

Avoiding Double Buffering

Most database servers maintain their own buffer cache and do not need the file system buffer cache. Database data cached in the file system buffer is therefore redundant and results in wasted memory and extra system CPU utilization to manage the buffer. By supporting direct I/O, Quick I/O eliminates double buffering. Data is copied directly between the relational database management system (RDBMS) cache and disk, which lowers CPU utilization and frees up memory that can then be used by the database server buffer cache to further improve transaction processing throughput.

Quick I/O Requirements

To use Quick I/O, you must:

- Preallocate files on a VxFS file system
- Use a special file naming convention to access the files

Preallocation

Preallocating database files for Quick I/O allocates contiguous space for the files. The file system space reservation algorithms attempt to allocate space for an entire file as a single contiguous extent. When this is not possible due to lack of contiguous space on the file system, the file is created as a series of direct extents. Accessing a file using direct extents is inherently faster than accessing the same data using indirect extents. Internal tests have shown performance degradation in OLTP throughput when using indirect extent access. In addition, this type of preallocation causes no fragmentation of the file system.

You must preallocate Quick I/O files because they cannot be extended through writes using their Quick I/O interfaces. They are initially limited to the maximum size you specify at the time of creation. To extend Quick I/O files, see "Extending a Quick I/O File" on page 89.

Naming Convention

VxFS uses a special naming convention to recognize and access Quick I/O files as raw character devices. VxFS recognizes the file when you add the following extension to a file name:

```
::cdev:vxfs:
```

Whenever an application opens an existing VxFS file with the extension `::cdev:vxfs:` (`cdev` being an acronym for *character device*), the file is treated as if it were a raw device. For example, if the file `temp01` is a regular VxFS file, then an application can access `temp01` as a raw character device by opening it with the name:

```
.temp01::cdev:vxfs:
```

Note We recommend reserving the `::cdev:vxfs:` extension *only* for Quick I/O files. If you are not using Quick I/O, you could technically create a regular file with this extension; however, doing so can cause problems if you later enable Quick I/O.

How to Set Up Quick I/O

Quick I/O is part of the VxFS binaries shipped with VERITAS Storage Foundation *for Oracle*. By default, Quick I/O is enabled when you mount a VxFS file system.

If Quick I/O is not available in the kernel, or the VERITAS Storage Foundation *for Oracle* license is not installed, a file system mounts by default without Quick I/O, the Quick I/O file name is treated as a regular file, and no error message is displayed. If, however, you specify the `-o qio` option, the `mount` command prints the following error message and terminates without mounting the file system.

```
VxFDD: You don't have a license to run this program
vxfs mount: Quick I/O not available
```

Depending on whether you are creating a new database or are converting an existing database to use Quick I/O, you have the following options:

- If you are creating a new database:
  - You can use the `qiomkfile` command to preallocate space for database files and make them accessible to the Quick I/O interface. See "Creating Database Files as Quick I/O Files Using qiomkfile" on page 72 for more information.
  - You can use the `setext` command to preallocate space for database files and create the Quick I/O files. See "Preallocating Space for Quick I/O Files Using the setext Command" on page 74.
- If you are converting an existing database:
  - You can create symbolic links for existing VxFS files, and use these symbolic links to access the files as Quick I/O files. See "Accessing Regular VxFS Files as Quick I/O Files" on page 76 for more information.
  - You can convert your existing Oracle database files to use the Quick I/O interface using the `qio_getdbfiles` and `qio_convertdbfiles` commands. See "Converting Oracle Files to Quick I/O Files" on page 78 for more information.

Creating Database Files as Quick I/O Files Using qiomkfile

The best way to preallocate space for database files and to make them accessible using the Quick I/O interface is to use the `qiomkfile` command. You can use the `qiomkfile` command to create Quick I/O files for either temporary 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.

qiomkfile Options

`-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 89 for more information.

`-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 89 for more information.

`-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.

Caution 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.

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.

▼ 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. 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' \
size 100M reuse;
SQL> exit;
```

Example

- 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 this 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 (soft) 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.

Prerequisites

- The `setext` command requires superuser (`root`) privileges.

Usage Notes

- You need to use the `chown` and `chgrp` commands 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 .filename
# chgrp dba .filename
# chmod 660 .filename
```

Example

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 .dbfile
# chgrp 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
```

Example

To access the VxFS file `dbfile` as a Quick I/O file:

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

To show the symbolic link created:

```
$ ls -lo .dbfile dbfile
-rw-r--r--   1 oracle         104890368     Oct 2 13:42  .dbfile
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. For information on creating Storage Checkpoints, see "Using Storage Checkpoints and Storage Rollback" on page 155.

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*.
- The repository must exist before you can convert to Quick I/O files. Run the dbed_update command to update or create the repository.
- Files you want to convert must be regular files on VxFS file systems or links that point to regular VxFS files.

Options

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. Specify this option only in environments where the type of database is ambiguous (for example, when multiple types of database environment variables, such as *$ORACLE_SID*, *SYBASE*, *DSQUERY*, and *$DB2INSTANCE*, are present on a server).

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).

`-f` Reports on the current fragmentation levels for database files listed in the mkqio.dat file. Fragmentation is reported as not fragmented, slightly fragmented, fragmented, highly fragmented.

`-h` Displays a help message.

`-i` 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`. Specify this option only in environments where the type of database is ambiguous (for example, when multiple types of database environment variables, such as *$ORACLE_SID*, *SYBASE*, *DSQUERY*, and *$DB2INSTANCE* are present on a server).

`-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.

Usage Notes

- Converting existing database files to be Quick I/O files may not be the best choice if the files are fragmented. Use the `-f` option to determine the fragmentation levels and either:

  - Exclude files that are highly fragmented and do not have sufficient contiguous extents for Quick I/O use.

  or

  - Create new files with the `qiomkfile` command, rather than converting the files using the `qio_convertdbfiles` command. The new files 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 84 for more information.

  For information on creating Quick I/O files for temporary tablespaces, see "Creating Database Files as Quick I/O Files Using qiomkfile" on page 72.

- 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 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.

▼ 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.

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:

```
/db01/file1         210356
/db01/file2         157996
/db01/file3         38096
/db01/file4         394932
/db01/file5         911784
```

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

1. Shut down the database.

Caution Running the qio_convertdbfiles command while the database is up and running can cause severe problems with your database, including loss of data, and corruption.

2. Run the `qio_convertdbfiles` command from the writable directory where the `mkqio.dat` list resides:

   ```
   $ 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.

   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 writable directory where the `mkqio.dat` list resides:

   ```
   $ 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.

Understanding 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 using regular files to Quick I/O with the `qio_getdbfiles` command on Oracle9*i*.

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

   To use the `create tablespace` command:

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

   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;
   ```

Example

To drop tablespace `tempts`, create a Quick I/O file `temp01.dbf`, and then create a new temporary tablespace `tempts` using the `create 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
$ sqlplus /nolog
SQL> connect / as dba;
SQL> create tablespace tempts \
datafile '/db01/temp01.dbf' \
size 100M reuse \
temporary;
Tablespace created.
```

Example

To drop tablespace `tempts`, create a Quick I/O file `temp01.dbf`, 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
$ 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.

Options

-al	Lists all files on a file system, including Quick I/O files and their links.
-lL	Shows if Quick I/O was successfully installed and enabled.
-alL	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
```

Example

To show the absolute path name created using `qiomkfile` with the `-a` option:

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

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

Use the `ls` command as follows:

```
  $ ls -lL filename
```

Example

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
```

where the first character, `c`, indicates it is a raw character device file, and the major and minor device numbers are displayed in the size field. If you see a `No such file or directory` message, Quick I/O did not install properly or does not have a valid license key.

▼ 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
```

Example

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

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.

Prerequisites

- You must have sufficient space on the file system to extend the Quick I/O file.

Options

`-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.

Usage Notes

- You can also grow VxFS file systems online (provided the underlying disk or volume can be extended) using the `fsadm` command.

Note You must have superuser (`root`) privileges to resize VxFS file systems using the `fsadm` command.

- You can also use the VxDBA Monitoring Agent to monitor file system space and automatically grow file systems. See "Managing Space Usage and the VxDBA Monitoring Agent" on page 379 for information on how to start and use the VxDBA Monitoring Agent, set space alarm thresholds, and automatically grow file systems.
- See the `fsadm_vxfs`(1M) and `qiomkfile`(1M) manual pages for more information.

▼ 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:

   ```
   # /opt/VRTS/bin/fsadm -b newsize /mount_point
   ```

   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
```

Example

- 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
```

- 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.

Options

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.

Usage Notes

◆ 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
```

Example

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
```

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 Recovering a Database

If you need to recover 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.

Prerequisites

- Recover your database before attempting to recreate the Quick I/O files.
- Log in as the Database Administrator (typically, the user ID oracle) 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. If you do not have a mkqio.dat file, see "Converting Oracle Files to Quick I/O Files" on page 78.
- The *ORACLE_SID* environment variable must be set.

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

Use the qio_recreate command as follows:

```
# /opt/VRTSdbed/bin/qio_recreate
```

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.

If...	**Then...**
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 you will see a warning message.
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 you will see a warning message.

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.

Prerequisites

- The file system you are planning to remount must be located in the `/etc/filesystems` 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 writable directory where the `mkqio.dat` list resides:

   ```
   $ /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
   ```

Using VERITAS Cached Quick I/O 4

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. Cached Quick I/O is best-suited for use with Oracle8*i*.

This chapter describes how to enable and use Cached Quick I/O for enhanced performance.

Topics covered in this chapter include:

Understanding Cached Quick I/O

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.

Overview of How to Set Up Cached Quick I/O

To set up and use Cached Quick I/O:

1. Enable Cached Quick I/O on the underlying file systems used for your database.

2. Exercise the system in your production environment to generate file I/O statistics.

3. Collect the file I/O statistics while the files are in use.

4. Analyze the file I/O statistics to determine which files benefit from Cached Quick I/O.

5. Disable Cached Quick I/O on files that do not benefit from caching.

The rest of this chapter discusses how to set up Cached Quick I/O in more detail.

Enabling Cached Quick I/O on the 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
```

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
```

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 is the name of a block device
- *dgname* is the name of the disk group
- *volname* is the name of the volume

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.

For 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
```

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.

▼ To obtain information on all vxtunefs system parameters

Use the vxtunefs command without grep:

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

Example

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

The vxtunefs command displays output similar to the following:

```
Filesystem i/o parameters for /db01
read_pref_io = 65536
read_nstream = 1
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 1
write_unit_io = 65536
pref_strength = 10
buf_breakup_size = 1048576
discovered_direct_iosz = 262144
max_direct_iosz = 1048576
default_indir_size = 8192
qio_cache_enable = 1
write_throttle = 0
max_diskq = 1048576
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 = 130150400
fcl_keeptime = 0
fcl_winterval = 3600
```

For a complete description of vxtunefs parameters and the tuning instructions, refer to 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.

Prerequisites

- You must enable Cached Quick I/O for the file systems. See "Enabling Cached Quick I/O on the File System" on page 101.

Usage Notes

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

Collecting I/O Statistics

▼ 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

Example

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

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

Analyzing 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
```

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. For more information on enabling and disabling Cached Quick I/O at the file level, see "Enabling and Disabling Cached Quick I/O for Individual Files" on page 108.

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.

Using 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 for more information.

Setting Cache Advisories for Individual 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
```

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
```

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
```

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
```

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
```

Using VERITAS Extension for Oracle Disk Manager 5

VERITAS Extension *for Oracle* Disk Manager is specifically designed for Oracle9i 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 Oracle9i or later users to improve database throughput for I/O intensive workloads with special I/O optimization.

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. With Oracle9i or later, you can access both Oracle Disk Manager and Quick I/O files so you have the option to convert or not to convert your old Quick I/O files.

Note If you are using Oracle8 or Oracle8*i*, we recommend using Quick I/O.

Note VERITAS Storage Foundation 4.1 *for Oracle* is the last major release that supports Oracle Disk Manager for raw devices.

This chapter describes how to set up and use the Oracle Disk Manager feature.

Topics covered in this chapter include:

Understanding 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 Oracle9i 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.

Note If you are upgrading to Oracle9i or later and would like to convert from Quick I/O to Oracle Disk Manager, see "Converting Quick I/O Files to Oracle Disk Manager Files" on page 121.

The benefits of using Oracle Disk Manager are:

- 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

Supporting Kernel Asynchronous I/O

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.

Supporting Direct I/O 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.

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.

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.

Note With Oracle9i Release 2 (9.2) or later, you can use the `FILESYSTEMIO_OPTIONS` initialization parameter to enable or disable asynchronous I/O, direct I/O, or Concurrent I/O on file system files. This parameter is applicable to JFS and JFS2 files only. This parameter is not applicable to VxFS files, ODM files, or QIO files. For further information, please refer to Oracle Disk Manager documentation provided by Oracle.

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.

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.

Oracle Disk Manager and Oracle Managed Files

Oracle9i 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 Oracle9i 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

This section contains examples illustrating the relationship between Oracle Disk Manager and OMF.

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.

Example

The following example shows the init.ora contents and the command for starting the database instance. To simplify Oracle UNDO management, the new Oracle9i or later init.ora parameter UNDO_MANAGEMENT is set to AUTO. This is known as System-Managed Undo.

```
$ 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
```

Example

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> HOST mkdir /PROD/EMP_INDEX;
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 oracle9i dba 96 May  3 15:43 EMP_INDEX/
drwxr-xr-x 2 oracle9i dba 96 May  3 15:43 EMP_TABLE/
-rw-r--r-- 1 oracle9i dba 104858112 May 3 17:28 ora_1_BEhYgc0m.log
-rw-r--r-- 1 oracle9i dba 104858112 May 3 17:27 ora_2_BEhYu4NA.log
-rw-r--r-- 1 oracle9i dba 806912 May 3 15:43 ora_BEahlfUX.ctl
-rw-r--r-- 1 oracle9i dba 10489856 May 3 15:43
ora_sys_undo_BEajPSVq.dbf
-rw-r--r-- 1 oracle9i dba 104861696 May 3 15:4
ora_system_BEaiFE8v.dbf
-rw-r--r-- 1 oracle9i dba 186 May 3 15:03 PROD.ora

./EMP_INDEX:
total 204808
-rw-r--r-- 1 oracle9i dba 104861696 May 3 15:43
ora_emp_inde_BEakGfun.dbf

./EMP_TABLE:
total 204808
-rw-r--r-- 1 oracle9i 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 *for Oracle*. VERITAS Extension for Oracle Disk Manager is enabled once VERITAS Storage Foundation *for Oracle* and Oracle9i or later are installed, and the VERITAS Extension for Oracle Disk Manager library is linked to the library in the `{ORACLE_HOME}/lib` directory.

Note Because 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.

Prerequisites

- VERITAS Storage Foundation *for Oracle* must be installed on your system.
- Oracle9*i*, or later, must be installed on your system.
- Do not enable Oracle Disk Manager when Cached Quick I/O is enabled for datafiles.

Note Oracle uses default file access methods if Oracle9i or later or VERITAS Storage Foundation *for Oracle* is not installed, or VxFS 4.1 is not available in the kernel.

Oracle 10g

▼ To link the VERITAS Extension for Oracle Disk Manager library into Oracle home for Oracle 10g

Use the `rm` and `ln` commands as follows.

If you are running 64-bit Oracle 10*g*, use the following commands:

```
# rm ${ORACLE_HOME}/lib/libodm10.so
# ln -s /opt/VRTSodm/lib/sparcv9/libodm.so \
${ORACLE_HOME}/lib/libodm10.so
```

Oracle9i

▼ To link the VERITAS Extension for Oracle Disk Manager library into Oracle home for Oracle9i

Use the `rm` and `ln` commands as follows.

If you are running 32-bit Oracle9*i*, use the following commands:

```
# rm ${ORACLE_HOME}/lib/libodm9.so
# ln -s /opt/VRTSodm/lib/libodm.so ${ORACLE_HOME} \
 /lib/libodm9.so
```

If you are running 64-bit Oracle9*i*, use the following commands:

```
# rm ${ORACLE_HOME}/lib/libodm9.so
# ln -s /opt/VRTSodm/lib/sparcv9/libodm.so \
${ORACLE_HOME}/lib/libodm9.so
```

When Oracle Disk Manager is enabled, the message “`Oracle instance running with ODM: VERITAS 4.0ODM Library, Version 1.1`.” is sent to the alert log.

When the system and instance are configured correctly, the Oracle Disk Manager feature is used, by default, for accessing any database storage.

Preparing 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. See "Understanding Fragmentation" on page 59 for information on defragmenting a file.

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. See "Converting Quick I/O Files to Oracle Disk Manager Files" on page 121 for more information.

If you plan to convert to Oracle9i or later, convert your database and then migrate your storage to VxFS files.

Converting Quick I/O Files to Oracle Disk Manager Files

If you plan to run VERITAS Storage Foundation *for Oracle* with Oracle9i or later, and you have been using Quick I/O files, it is recommended that you convert your Quick I/O files to regular files. This should be done after you upgrade VERITAS Storage Foundation *for Oracle*.

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 Oracle9i or later only.

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

1. 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. Run `qio_convertdbfiles` in the directory containing the `mkqio.dat` file. (This script converts all Quick I/O files to ODM files.)

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

4. Restart the database instance.

Note Because 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.

Verifying that Oracle Disk Manager is Configured

To verify the configuration of the Oracle Disk Manager, make sure the following Prerequisites are satisfied.

Prerequisites

- The `VRTSdbed` license must be valid.
- The `VRTSodm` package must be installed.
- `/opt/VRTSodm/lib/libodm.so` must exist.
- If you are using Oracle9*i*, `$ORACLE_HOME/lib/libodm9.so` is linked to `/opt/VRTSodm/lib/libodm.so`
- If you are using Oracle 10*g*, `$ORACLE_HOME/lib/libodm10.so` is linked to `/opt/VRTSodm/lib/libodm.so`

▼ To verify that Oracle Disk Manager is configured

1. Check the `VRTSdbed` license:

```
# /opt/VRTS/bin/vxlictest -n “VERITAS Storage Foundation for \
Oracle” -f “ODM”
```

```
  Using VERITAS License Manager API Version  3.00, Build 2
  ODM feature is licensed
```

Check that the VRTSodm package is installed:

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

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

If you are running 32-bit Oracle9*i*, 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 Oracle9*i*, 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
```

3. Check that `libodm.so` is linked:

For Oracle 10g

If you are running 64-bit Oracle 10*g*, use the following command:

```
# cmp $ORACLE_HOME/lib/libodm10.so \
/opt/VRTSodm/lib/sparcv9/libodm.so
# echo $?
0
```

For Oracle9i

If you are running 32-bit Oracle9*i*, use the following command:

```
# cmp $ORACLE_HOME/lib/libodm9.so /opt/VRTSodm/lib/libodm.so
# echo $?
0
```

If you are running 64-bit Oracle9*i*, use the following command:

```
# cmp $ORACLE_HOME/lib/libodm9.so \
/opt/VRTSodm/lib/sparcv9/libodm.so
# echo $?
0
```

Note You can also use the `dbed_checkconfig` command, which is installed with VERITAS Storage Foundation *for Oracle* to check these conditions. See "Checking Oracle Configuration Environment Using dbed_checkconfig" on page 432 or the `dbed_checkconfig`(1M) manual page for more information.

▼ 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. 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 4.1 ODM Library,
Version 1.1
```

Disabling the Oracle Disk Manager Feature

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

Note To convert to VxFS with Quick I/O, disable the Oracle Disk Manager using the steps below. Then, see "Converting Oracle Files to Quick I/O Files" on page 78 for more information.

▼ To disable the Oracle Disk Manager feature in an Oracle instance

1. Shut down the database instance.

2. Use the `rm` and the `ln` commands to remove the link to the Oracle Disk Manager Library as follows:

   For Oracle 10g

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

   For Oracle9i

```
  # rm ${ORACLE_HOME}/lib/libodm9.so
  # ln -s ${ORACLE_HOME}/lib/libodmd9.so
${ORACLE_HOME}/lib/libodm9.so
```

3. Restart the database instance.

Note The steps for disabling the Oracle Disk Manager feature are the same for both 32- and 64-bit Oracle9*i*.

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

Using Storage Mapping 6

Storage mapping is included with VERITAS Storage Foundation *for Oracle* Enterprise Edition. Storage mapping enables you to map datafiles to physical devices. You may obtain and view detailed storage topology information using the `vxstorage_stats` and `dbed_analyzer` commands or the VERITAS Storage Foundation *for Oracle* GUI. You may also use the Oracle Enterprise Manager to access storage mapping information.

Topics include:

Understanding Storage Mapping

Access to mapping information is important since it allows for a detailed understanding of the storage hierarchy in which files reside, information that is critical for effectively evaluating I/O performance.

Mapping files to their underlying device is straightforward when datafiles are created directly on a raw device. With the introduction of host-based volume managers and sophisticated storage subsystems that provide RAID features, however, mapping files to physical devices has become more difficult.

With the VERITAS Storage Foundation *for Oracle* Storage Mapping option, you can map datafiles to physical devices. VERITAS Storage Mapping relies on VERITAS Federated Mapping Service (VxMS), a library that assists in the development of distributed SAN applications that must share information about the physical location of files and volumes on a disk.

The VERITAS Storage Mapping option supports Oracle's set of storage APIs called Oracle Mapping ("ORAMAP" for short) that lets Oracle determine the mapping information for files and devices.

Oracle provides a set of dynamic performance views (`v$` views) that shows the complete mapping of a file to intermediate layers of logical volumes and physical devices. These views enable you to locate the exact disk on which any specific block of a file resides. You can use these mappings, along with device statistics, to evaluate I/O performance.

The VERITAS Storage Foundation *for Oracle* Storage Mapping option supports a wide range of storage devices and allows for "deep mapping" into EMC, Hitachi, and IBM Enterprise Storage Server ("Shark") arrays. Deep mapping information identifies the physical disks that comprise each LUN and the hardware RAID information for the LUNs.

You can view storage mapping topology information and I/O statistics using:

- The `vxstorage_stats` command. This command displays the complete I/O topology mapping of specific datafiles through intermediate layers like logical volumes down to actual physical devices.
- The `dbed_analyzer` command. This command retrieves tablespace-to-physical disk mapping information for all the datafiles in a specified database. It also provides information about the amount of disk space being used by a tablespace.
- The VERITAS Storage Foundation *for Oracle* GUI. The VERITAS Storage Foundation *for Oracle* GUI performs file mapping and displays both storage mapping topology information and I/O statistics.

In addition, you can also use the Oracle Enterprise Manager GUI to display storage mapping information after file mapping has occurred. Oracle Enterprise Manager does not display I/O statistics information. Unlike the information displayed in the VERITAS Storage Foundation *for Oracle* GUI, the information displayed in Oracle Enterprise Manager may be “stale,” that is, it may not be the latest information.

Note If you use VERITAS FlashSnap Agent for Symmetrix, you cannot use the mapping functionality for non-Symmetrix arrays.

Verifying VERITAS Storage Mapping Setup

Before using the VERITAS Storage Mapping option, verify that the features are set up correctly.

▼ To verify that your system is using the VERITAS Storage Mapping option

1. Verify that you have a license key for the storage mapping option.

```
# /opt/VRTS/bin/vxlictest -n "VERITAS Mapping Services" -f \
 "Found_Edi_map"
 Found_Edi_map feature is licensed
```

2. Verify that the VRTSvxmsa package is installed.

```
# pkginfo -l VRTSvxmsa
```

Output similar to the following is displayed:

```
PKGINST:  VRTSvxmsa
     NAME:  VxMS - VERITAS Mapping Service, Application Libraries.
 CATEGORY:  system,utilities
     ARCH:  sparc
  VERSION: 4.2.1-REV=build218_2004.10.29
  BASEDIR:  /opt
   PSTAMP:  oigpsol0920041029112628
 INSTDATE:  Nov 02 2004 15:22
   STATUS:  completely installed
    FILES:     33 installed pathnames
               8 shared pathnames
               14 directories
               15 executables
               2931 blocks used (approx)
```

Using vxstorage_stats

The vxstorage_stats command displays detailed storage mapping information and I/O statistics about an individual VxFS file. The mapping information and I/O statistics are recorded only for VxFS files and VxVM volumes.

In vxstorage_stats command output, I/O topology information appears first followed by summary statistics for each object.

The command syntax is as follows:

```
/opt/VRTSdbed/bin/vxstorage_stats [-m] [-s] [-i interval
-c count] -f filename
```

Prerequisites

- You must log in as the database administrator (typically, the user ID oracle) or root.

Usage Notes

- The -m option displays the I/O topology for the specified file.
- The -s option displays the file statistics for the specified file.
- The -c *count* option specifies the number of times to display statistics within the interval specified by -i *interval*.
- The -i *interval* option specifies the interval frequency for displaying updated I/O statistics.
- The -f *filename* option specifies the file to display I/O mapping and statistics for.
- For more information, see the vxstorage_stats(1m) online manual page.

Displaying Storage Mapping Information

▼ To display storage mapping information

Use the vxstorage_stats command with the -m option to display storage mapping information:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -m -f file_name
```

Example

```
$ /opt/VRTSdbed/bin/vxstorage_stats -m -f /oradata/system01.dbf
```

Output similar to the following is displayed:

```
TY NAME                      NSUB  DESCRIPTION   SIZE(sectors)   OFFSET(sectors)
PROPERTIES
fi /oradata/system01.dbf     1     FILE          2621442048(B)   4718592(B)
Extents: 3      Sparse Extents:0
v  myindex                   1     MIRROR        16777216        0
pl vxvm:mydb/myindex-01      3     STRIPE        16779264        0
Stripe_size:2048
rd /dev/vx/rdmp/c3t1d3s3     1     PARTITION     5593088         0
sd /dev/rdsk/c3t1d3s3        1     PARTITION     17674560        960
sd c3t1d3                    2     MIRROR        17677440        0
da EMC000184502242:02:0c:02  0     DISK          143113019       0
da EMC000184502242:31:0c:02  0     DISK          143113019       0
rd /dev/vx/rdmp/c3t1d15s4    1     PARTITION     5593088         0
sd /dev/rdsk/c3t1d15s4       1     PARTITION     17669760        5760
sd c3t1d15                   2     MIRROR        17677440        0
da EMC000184502242:01:0c:02  0     DISK          143113019       0
da EMC000184502242:32:0c:02  0     DISK          143113019       0
rd /dev/vx/rdmp/c3t1d2s4     1     PARTITION     5593088         0
sd /dev/rdsk/c3t1d2s4        1     PARTITION     17671680        3840
sd c3t1d2                    2     MIRROR        17677440        0
da EMC000184502242:16:0c:02  0     DISK          143113019       0
da EMC000184502242:17:0c:02  0     DISK          143113019       0
```

Note For file type (`fi`), the `SIZE` column is number of bytes, and for volume (`v`), plex (`pl`), sub-disk (`sd`), and physical disk (`da`), the `SIZE` column is in 512-byte blocks. Stripe sizes are given in sectors.

Displaying I/O Statistics Information

▼ To display I/O statistics information

Use the `vxstorage_stats` command with the `-s` option to display I/O statistics information:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -f file_name
```

Example

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -f \
/data/system01.dbf
```

Output similar to the following is displayed:

```
                            I/O OPERATIONS     I/O BLOCKS(512 byte)   AVG TIME(ms)

OBJECT                      READ      WRITE    B_READ      B_WRITE    AVG_RD    AVG_WR
/data/system01.dbf          2         2479     8           5068810    0.00      53.28
/dev/vx/rdsk/mydb/myindex   101       2497     1592        5069056    12.18     52.78
```

```
vxvm:mydb/myindex-01        101      2497     1592       5069056    12.18     52.76
/dev/rdsk/c3t1d3s3          131      1656     2096       1689696    14.43     39.09
c3t1d3                      131      1656     2096       1689696    14.43     39.09
EMC000184502242:02:0c:02    8480     231019   275952     23296162   -         -
EMC000184502242:31:0c:02    3244     232131   54808      23451325   -         -
/dev/rdsk/c3t1d15s4         0        1652     0          1689606    0.00      46.47
c3t1d15                     0        1652     0          1689606    0.00      46.47
EMC000184502242:01:0c:02    23824    1188997  1038336    32407727   -         -
EMC000184502242:32:0c:02    5085     852384   135672     29956179   -         -
/dev/rdsk/c3t1d2s4          14       1668     200        1689834    18.57     34.19
c3t1d2                      14       1668     200        1689834    18.57     34.19
EMC000184502242:16:0c:02    4406     271155   121368     23463948   -         -
EMC000184502242:17:0c:02    3290     269281   55432      23304619   -         -
```

▼ To display storage mapping and I/O statistics information at repeated intervals

Use the `vxstorage_stats` command with the `-i` *`interval`* and `-c` *`count`* options to display storage mapping and I/O statistics information at repeated intervals. The `-i` *`interval`* option specifies the interval frequency for displaying updated I/O statistics and the `-c` *`count`* option specifies the number of times to display statistics.

```
$ /opt/VRTSdbed/bin/vxstorage_stats [-m] [-s] \
[-i interval -c count ] -f file_name
```

Example

To display statistics twice with a time interval of two seconds:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -i2 -c2 \
-f /data/system01.dbf
```

Output similar to the following is displayed:

```
                          OPERATIONS     FILE BLOCKS(512 byte)      AVG TIME(ms)

OBJECT                      READ     WRITE     B_READ  B_WRITE     AVG_RD    AVG_WR
/data/system01.dbf          615      19        20752   152         3.53      24.74
/dev/vx/rdsk/mapdg/data_vol 19386    33227     895692  1376438     9.27      16.18
vxvm:mapdg/data_vol-01      19386    33227     895692  1376438     9.26      14.03
/dev/rdsk/c1t10d0s2         19386    33227     895692  1376438     9.26      14.03
c1t10d0                     19386    33227     895692  1376438     9.26      14.03
vxvm:mapdg/data_vol-03      0        33227     0       1376438     0.00      14.21
/dev/rdsk/c1t13d0s2         0        33227     0       1376438     0.00      14.21
c1t13d0                     0        33227     0       1376438     0.00      14.21
                          OPERATIONS     FILE BLOCKS(512 byte)      AVG TIME(ms)

OBJECT                      READ     WRITE     B_READ  B_WRITE     AVG_RD    AVG_WR
/data/system01.dbf          0        0         0       0           0.00      0.00
/dev/vx/rdsk/mapdg/data_vol 0        1         0       2           0.00      0.00
vxvm:mapdg/data_vol-01      0        1         0       2           0.00      0.00
/dev/rdsk/c1t10d0s2         0        1         0       2           0.00      0.00
c1t10d0                     0        1         0       2           0.00      0.00
vxvm:mapdg/data_vol-03      0        1         0       2           0.00      0.00
```

```
/dev/rdsk/c1t13d0s2           0       1         0       2          0.00      0.00
c1t13d0                       0       1         0       2          0.00      0.00
```

Using dbed_analyzer

Effectively performing a parallel backup requires an understanding of which tablespaces reside on which disks. If two tablespaces reside on the same disk, for example, backing them up in parallel will not reduce their downtime.

The `dbed_analyzer` command provides tablespace-to-physical disk mapping information for *all the datafiles in a specified tablespace, list of tablespaces, or an entire database*. (In contrast, the `vxstorage_stats` command provides this information on a per-file basis only.) In addition, `dbed_analyzer` provides information about the amount of disk space they are using.

Prerequisites

- You must log in as the database administrator (typically, the user ID `oracle`).

Usage Notes

- The `-o sort=tbs` option provides the layout of the specified tablespaces on the physical disk as well as the amount of disk space they are using.
- The `-o sort=disk` option provides the name of the disks containing the specified tablespaces as well as the amount of disk space the tablespaces are using.
- The `-f` *`filename`* option specifies the name of a file containing a list of the tablespaces for which to obtain mapping information.
- The `-t` *`tablespace`* option specifies the name of a tablespace for which to obtain mapping information.
- If `-f` *`filename`* or `-t` *`tablespace`* is not specified then all the tablespaces in the database will be analyzed.
- For more information, see the `dbed_analyzer`(1M) online manual page.

Obtaining Storage Mapping Information for a List of Tablespaces

▼ To obtain storage mapping information sorted by tablespace

Use the `dbed_analyzer` command with the `-f` *`filename`* and `-o sort=tbs` options:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S $ORACLE_SID -H $ORACLE_HOME \
-o sort=tbs -f filename
```

Example

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S PROD -H /usr1/oracle \
-o sort=tbs -f /tmp/tbsfile
```

Output similar to the following is displayed in the file `tbsfile`:

```
TBSNAME    DATAFILE                          DEVICE
SIZE(sectors)
SYSTEM     /usr1/oracle/rw/DATA/PROD.dbf        c3t21000020379DBD5Fd0  819216
TEMP       /usr1/oracle/rw/DATA/temp_20000      c3t21000020379DBD5Fd0  1021968
TEMP       /usr1/oracle/rw/DATA/temp_20001      c3t21000020379DBD5Fd0  2048016
SYSAUX     /usr1/oracle/rw/DATA/sysaux.dbf      c3t21000020379DBD5Fd0  819216
ITEM       /usr1/oracle/rw/DATA/item_1000       c3t21000020379DBD5Fd0  1021968
ITM_IDX    /usr1/oracle/rw/DATA/itm_idx_2000    c3t21000020379DBD5Fd0  1021968
PRODID_IDX /usr1/oracle/rw/DATA/prodid_idx_3000 c3t21000020379DBD5Fd0  1021968
QTY_IDX    /usr1/oracle/rw/DATA/qty_idx_7000    c3t21000020379DBD5Fd0  1021968
ROLL_1     /usr1/oracle/rw/DATA/roll_1_5000     c3t21000020379DBD5Fd0  1021968
ROLL_2     /usr1/oracle/rw/DATA/roll_2_6000     c3t21000020379DBD5Fd0  1021968
ORDERS     /usr1/oracle/rw/DATA/orders_4000     c3t21000020379DBD5Fd0  1021968
ORD_IDX    /usr1/oracle/rw/DATA/ord_idx_10000   c3t21000020379DBD5Fd0  1021968
QTY_IDX    /usr1/oracle/rw/DATA/qty_idx_7001    c3t21000020379DBD5Fd0  1024016
ITM_IDX    /usr1/oracle/rw/DATA/itm_idx_2001    c3t21000020379DBD5Fd0  1024016
ROLL_1     /usr1/oracle/rw/DATA/roll_1_5001     c3t21000020379DBD5Fd0  1024016
QTY_IDX    /usr1/oracle/rw/DATA/qty_idx_7002    c3t21000020379DBD5Fd0  1024016
ROLL_2     /usr1/oracle/rw/DATA/roll_2_6001     c3t21000020379DBD5Fd0  1024016
ITEM       /usr1/oracle/rw/DATA/item_1001       c3t21000020379DBD5Fd0  4096016
```

▼ To obtain storage mapping information sorted by disk

Use the `dbed_analyzer` command with the `-f` *`filename`* and `-o sort=disk` options:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S $ORACLE_SID -H $ORACLE_HOME \
-o sort=disk -f filename
```

Example

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S PROD -H /usr1/oracle \
-o sort=disk -f /tmp/tbsfile
```

Output similar to the following is displayed in the file `tbsfile`:

```
DEVICE                 TBSNAME  DATAFILE                         SIZE(sectors)
c3t21000020379DBD5Fd0  SYSTEM   /usr1/oracle/rw/DATA/PROD.dbf          819216
c3t21000020379DBD5Fd0  TEMP     /usr1/oracle/rw/DATA/temp_20000        1021968
c3t21000020379DBD5Fd0  TEMP     /usr1/oracle/rw/DATA/temp_20001        2048016
c3t21000020379DBD5Fd0  SYSAUX   /usr1/oracle/rw/DATA/sysaux.dbf        819216
```

```
c3t21000020379DBD5Fd0  ITEM      /usr1/oracle/rw/DATA/item_1000         1021968
c3t21000020379DBD5Fd0  ITM_IDX   /usr1/oracle/rw/DATA/itm_idx_2000      1021968
c3t21000020379DBD5Fd0  PRODID_IDX /usr1/oracle/rw/DATA/prodid_idx_3000  1021968
c3t21000020379DBD5Fd0  QTY_IDX   /usr1/oracle/rw/DATA/qty_idx_7000      1021968
c3t21000020379DBD5Fd0  ROLL_1    /usr1/oracle/rw/DATA/roll_1_5000       1021968
c3t21000020379DBD5Fd0  ROLL_2    /usr1/oracle/rw/DATA/roll_2_6000       1021968
c3t21000020379DBD5Fd0  ORDERS    /usr1/oracle/rw/DATA/orders_4000       1021968
c3t21000020379DBD5Fd0  ORD_IDX   /usr1/oracle/rw/DATA/ord_idx_10000     1021968
c3t21000020379DBD5Fd0  QTY_IDX   /usr1/oracle/rw/DATA/qty_idx_7001      1024016
c3t21000020379DBD5Fd0  ITM_IDX   /usr1/oracle/rw/DATA/itm_idx_2001      1024016
c3t21000020379DBD5Fd0  ROLL_1    /usr1/oracle/rw/DATA/roll_1_5001       1024016
c3t21000020379DBD5Fd0  QTY_IDX   /usr1/oracle/rw/DATA/qty_idx_7002      1024016
c3t21000020379DBD5Fd0  ROLL_2    /usr1/oracle/rw/DATA/roll_2_6001       1024016
c3t21000020379DBD5Fd0  ITEM      /usr1/oracle/rw/DATA/item_1001         4096016
```

Viewing Storage Mapping Topology and I/O Statistics Using the GUI

▼ **To view Oracle datafile storage mapping topology and I/O statistics using the GUI**

1. Start the GUI and connect to the desired host.

2. Expand the Oracle Databases icon in the object tree and then expand the desired database.

3. Under the desired database, select a datafile in the object tree. (You will need to further expand the tree view to find the icon.)

4. Use one of the following methods to generate datafile statistics.

   ◆ Click on **Datafiles**> **Topology/Statistic**.

   or

   ◆ Right click on the datafile to bring up a pop-up menu. Then click on **Topology/Statistic**.

   or

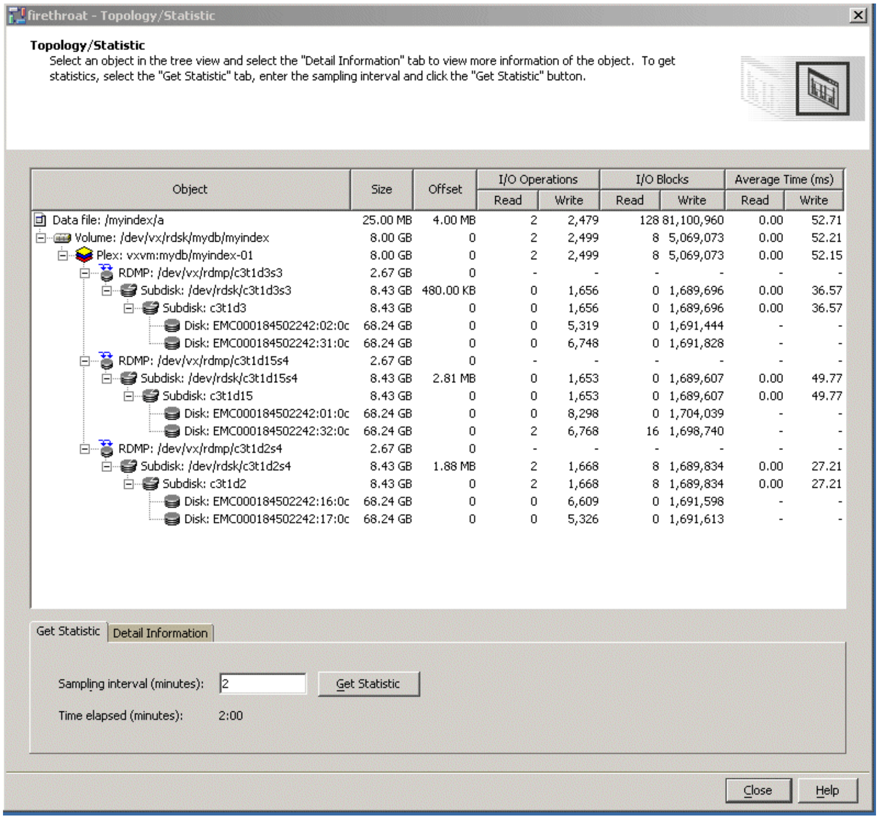
   ◆ Click on the **Topology/Statistic** icon in the main toolbar.

   The **Topology/Statistic** window is displayed.

5. To view more information about an object, select an object in the object tree and then click on the **Detailed Information** tab at the bottom of the **Topology/Statistic** window.

6. Optionally, if you want to view I/O statistics, enter a sampling time (in minutes) in the **Sampling interval (minutes)** field and then press **Get Statistics**.

   Information is displayed in the **I/O Operations**, **I/O Blocks**, and **Average Time (ms)** columns of the window.



Topology/Statistic

Select an object in the tree view and select the "Detail Information" tab to view more information of the object. To get statistics, select the "Get Statistic" tab, enter the sampling interval and click the "Get Statistic" button.

Object	Size	Offset	I/O Operations		I/O Blocks		Average Time (ms)	
			Read	Write	Read	Write	Read	Write
Data file: /myindex/a	25.00 MB	4.00 MB	2	2,479	128	81,100,960	0.00	52.71
Volume: /dev/vx/rdsk/mydb/myindex	8.00 GB	0	2	2,499	8	5,069,073	0.00	52.21
Plex: vxvm:mydb/myindex-01	8.00 GB	0	2	2,499	8	5,069,073	0.00	52.15
RDMP: /dev/vx/rdmp/c3t1d3s3	2.67 GB	0	-	-	-	-	-	-
Subdisk: /dev/rdsk/c3t1d3s3	8.43 GB	480.00 KB	0	1,656	0	1,689,696	0.00	36.57
Subdisk: c3t1d3	8.43 GB	0	0	1,656	0	1,689,696	0.00	36.57
Disk: EMC000184502242:02:0c	68.24 GB	0	0	5,319	0	1,691,444	-	-
Disk: EMC000184502242:31:0c	68.24 GB	0	0	6,748	0	1,691,828	-	-
RDMP: /dev/vx/rdmp/c3t1d15s4	2.67 GB	0	-	-	-	-	-	-
Subdisk: /dev/rdsk/c3t1d15s4	8.43 GB	2.81 MB	0	1,653	0	1,689,607	0.00	49.77
Subdisk: c3t1d15	8.43 GB	0	0	1,653	0	1,689,607	0.00	49.77
Disk: EMC000184502242:01:0c	68.24 GB	0	0	8,298	0	1,704,039	-	-
Disk: EMC000184502242:32:0c	68.24 GB	0	2	6,768	16	1,698,740	-	-
RDMP: /dev/vx/rdmp/c3t1d2s4	2.67 GB	0	-	-	-	-	-	-
Subdisk: /dev/rdsk/c3t1d2s4	8.43 GB	1.88 MB	2	1,668	8	1,689,834	0.00	27.21
Subdisk: c3t1d2	8.43 GB	0	2	1,668	8	1,689,834	0.00	27.21
Disk: EMC000184502242:16:0c	68.24 GB	0	0	6,609	0	1,691,598	-	-
Disk: EMC000184502242:17:0c	68.24 GB	0	0	5,326	0	1,691,613	-	-

Sampling interval (minutes): 2

Time elapsed (minutes): 2:00

7. When you are finished, click **Close**.

Oracle File Mapping (ORAMAP)

VERITAS has defined and implemented a single mapping library, `libvxoramap_64.so`, that provides a mapping interface to Oracle9*i* release 2 or later. `libvxoramap_64.so` serves as a bridge between Oracle's set of storage APIs (known as "ORAMAP") and VERITAS Federated Mapping Service (VxMS), a library that assists in the development of distributed SAN applications that must share information about the physical location of files and volumes on a disk.

With VERITAS Storage Foundation *for Oracle* Storage Mapping option, you can view the complete I/O topology mapping of datafiles through intermediate layers like logical volumes down to actual physical devices. This information can be used to determine the exact location of an Oracle data block on a physical device and to help identify hot spots.

Note To use the mapping functionality, you must be using Oracle 9.2.0.3 or later.

Mapping Components

The mapping components in the System Global Area (SGA) and Oracle's representation of these components are described in this section. You will need an understanding of these components to interpret the mapping information in Oracle's dynamic performance views.

The mapping information in Oracle's dynamic performance views consists of:

- File components

  A mapping file component is a mapping structure describing a file. It provides a set of attributes for a file, including the file's size, number of extents, and type. File components are exported to the user through `v$map_file`.

- File extent components

  A mapping file extent component describes a contiguous group of blocks residing on one element. The description specifies the device offset, the extent size, the file offset, the extent type (`Data` or `Parity`), and the name of the element where the extent resides.

- Element components

  A mapping element component is a mapping structure that describes a storage component within the I/O stack. Elements can be mirrors, stripes, partitions, RAID5, concatenated elements, and disks.

  This component contains information about the element's mapping structure, such as the element's size, type, number of subelements, and a brief description. Element components are exported to the user through `v$map_element`.

- Subelement components

  A mapping subelement component describes the link between an element and the next element in the I/O stack. The subelement component contains the subelement number, size, the element name for the subelement, and the element offset. Subelement components are exported to the user through `v$map_subelement`.

These four types of mapping components completely describe the mapping information for an Oracle instance.

Storage Mapping Views

The mapping information that is captured is presented in Oracle's dynamic performance views. Brief descriptions of these views are provided below. For more detailed information, refer to your Oracle documentation.

View	Description
`V$MAP_LIBRARY`	Contains a list of all the mapping libraries that have been dynamically loaded by the external process.
`V$MAP_FILE`	Contains a list of all the file mapping structures in the shared memory of the instance.
`V$MAP_FILE_EXTENT`	Contains a list of all the file extent mapping structures in the shared memory of the instance.
`V$MAP_ELEMENT`	Contains a list of all the element mapping structures in the SGA of the instance.
`V$MAP_EXT_ELEMENT`	Contains supplementary information for all element mapping structures.
`V$MAP_SUBELEMENT`	Contains a list of all subelement mapping structures in the shared memory of the instance.
`V$MAP_COMP_LIST`	Describes the component list associated with the element name.
`V$MAP_FILE_IO_STACK`	Contains the hierarchical arrangement of storage containers for the file. This information is displayed as a series of rows. Each row represents a level in the hierarchy.

Verifying Oracle File Mapping Setup

▼ To verify that $ORACLE_HOME is set up for Oracle file mapping (ORAMAP)

1. Enter:

```
# cd $ORACLE_HOME/rdbms/filemap/bin
# ls -l
-r-xr-x--- 1 root system 900616 Apr 08 19:16 fmputl
-r-sr-xr-x 1 root system 14614 Apr 08 19:16 fmputlhp
```

2. Verify that:

   - fmputlhp is owned by root and that the setud bit is set.
   - The permissions for fmputlhp are set to -r-sr-xr-x.
   - The permissions for fmputl are set to -r-xr-x---.

3. If any of these items is not set as specified, make the appropriate corrections.

Enabling Oracle File Mapping

▼ To enable Oracle file mapping with the VERITAS Storage Mapping option

1. Ensure that the file filemap.ora exists and contains a valid entry for the VERITAS mapping library for Oracle storage mapping.

```
# cd $ORACLE_HOME/rdbms/filemap/etc
# cat filemap.ora
```

   For 32-bit Oracle, the filemap.ora file should contain the following setting:

```
lib=VERITAS:/opt/VRTSdbed/lib/libvxoramap_32.so
```

   For 64-bit Oracle, the filemap.ora file should contain the following setting:

```
lib=VERITAS:/opt/VRTSdbed/lib/libvxoramap_64.so
```

2. After verifying that the system is using the VERITAS library for Oracle storage mapping, set the file_mapping initialization parameter to true.

```
SQL> alter system set file_mapping=true;
```

   The file_mapping initialization parameter is set to false by default. You do not need to shut down the instance to set this parameter. Setting file_mapping=true starts the FMON background process.

Note If you want storage mapping to be enabled whenever you start up an instance, set the `file_mapping` initialization parameter to `true` in the `init.ora` file.

Accessing Dynamic Performance Views

▼ To access dynamic performance views

1. Confirm that the VERITAS mapping library for Oracle file mapping has been enabled.

```
SQL> select lib_idx idx, lib_name name, vendor_name vname, \
     path_name path from v$map_library;
```

```
IDX      NAME                                     VNAME      PATH
-------- ---------------------------------------- ------------------------------
1        Veritas ORAMAP API                       Veritas    /opt/VRTSdbed/lib/libvxoramap.so
```

2. After storage mapping has been enabled, Oracle datafiles can be mapped using the `DBMS_STORAGE_MAP` package.

   The following example shows how to map a datafile using SQL:

```
SQL> execute dbms_storage_map.map_file('/ora92/dbs/qio10m.dbf',
'DATAFILE', TRUE);
```

   For more information about various features and capabilities of the `DBMS_STORAGE_MAP` package, see your Oracle documentation.

3. Use SQL commands to display the mapping information that is captured in Oracle's dynamic performance views.

   To display the contents of `v$map_file` for a Quick I/O file:

```
SQL> select file_name name, file_map_idx idx, \
     file_status status, file_type type, file_structure str, \
     file_size fsize, file_nexts nexts from v$map_file;
```

```
NAME                     DX      STATUS    TYPE        STR       FSIZE       NEXTS
----------------------   ----    -------   ---------   -----     -------     -----
/ora92/dbs/qio10m.dbf    0       VALID     DATAFILE    FILE      20488       1
```

   To display the contents of `v$map_file_extent`.

```
SQL> select * from v$map_file_extent;
```

```
FILE_MAP_IDX EXT_NUM     EXT_ELEM_OFF EXT_SIZE    EXT_FILE_OFF EXT_TY ELEM_IDX
------------ ----------------------- ----------------------- ------ ------------
0            0           7733248      20488       0            DATA   0
```

To display the contents of v$map_element:

```
SQL> select elem_idx idx, elem_name, elem_type type, elem_size, \
     elem_nsubelem nsub, elem_descr, stripe_size from \
     v$map_element;
```

```
IDX  ELEM_NAME                              TYPE         ELEM_SIZE  NSUB      ELEM_DESCR      STRIPE_SIZE
---- -------------------------------------- ---------    ---------  --------  ------------    -----------
0    /dev/vx/rdsk/PROD_dg/ora92             MIRROR       12582912   1         MIRROR          0
1    vxvm:PROD_dg/ora92-01                  CONCATENATED 12586455   1         VERITAS VOLUME  0
2    /dev/vx/rdmp/c2t5d0s4                  PARTITION    17674902   1         HOST DEVICE     0
3    /dev/rdsk/c2t5d0s4                     PARTITION    17674902   1         DEVICE          0
4    c2t5d0                                 DISK         17682084   0         DISK            0
```

To display the contents of v$map_subelement:

```
SQL> select * from v$map_subelement;
```

```
CHILD_IDX PARENT_IDX  SUB_NUM SUB_SIZE ELEM_OFFSET SUB_FLAGS
--------- ---------- -------- -------- ----------- ---------
        1          0        0 12586455           0         0
        2          1        0 12586455           0         0
        3          2        0 17674902           0         0
        4          3        0 17682084        7182         0
```

To display all the elements within the I/O stack for a specific file.

```
SQL> with fv as
  2       (select file_map_idx, file_name from v$map_file
  3       where file_name = '/ora92/dbs/qio10m.dbf')
  4  select
  5       fv.file_name, lpad(' ', 4 * (level - 1)) || \
          el.elem_name elem_name, el.elem_size, el.elem_type, \
          el.elem_descr
  6  from
  7       v$map_subelement sb, v$map_element el, fv,
  8       (select unique elem_idx from v$map_file_io_stack io, fv
  9       where io.file_map_idx = fv.file_map_idx) fs
  10      where el.elem_idx = sb.child_idx
  11      and fs.elem_idx = el.elem_idx
  12      start with sb.parent_idx in
  13      (select distinct elem_idx
  14      from v$map_file_extent fe, fv
  15      where fv.file_map_idx = fe.file_map_idx)
  16      connect by prior sb.child_idx = sb.parent_idx;
```

```
FILE_NAME                     ELEM_NAME                     ELEM_SIZE ELEM_TYPE    ELEM_DESCR
```

```
---------------------------- ---------------------------- --------- ------------ -------------
/ora92/dbs/qio10m.dbf      vxvm:PROD_dg/ora92-01       12586455 CONCATENATED VERITAS VOLUME
/ora92/dbs/qio10m.dbf            /dev/vx/rdmp/c2t5d0s4      17674902 PARTITION    HOST DEVICE
/ora92/dbs/qio10m.dbf                /dev/rdsk/c2t5d0s4     17674902 PARTITION    DEVICE
/ora92/dbs/qio10m.dbf                    c2t5d0             17682084 DISK         DISK
```

Using Oracle Enterprise Manager

Oracle Enterprise Manager is a web-based GUI for managing Oracle databases. You can use this GUI to perform a variety of administrative tasks such as creating tablespaces, tables, and indexes; managing user security; and backing up and recovering your database. You can also use Oracle Enterprise Manager to view performance and status information about your database instance.

From Oracle Enterprise Manager, you can view storage mapping information and a graphical display of the storage layout. Storage mapping information cannot be viewed with the Oracle 10g version of the Oracle Enterprise Manager client. However, the Oracle9i version of Oracle Enterprise Manager can be used with Oracle 10g to view storage mapping information.

For more information about Oracle Enterprise Manager, refer to your Oracle documentation.

▼ To view storage information

1. To view storage information, start Oracle Enterprise Manager and select a database from the left navigational pane (the object tree) of the Oracle Enterprise Manager Console.

2. Expand the Databases icon and select the desired database.

   The Database Connect Information window appears.

3. Enter a user name and password to log in to the database and click **OK**.

4. In the object tree, expand the **Storage** icon.

5. Under the **Storage** icon, expand the **Datafiles** icon.

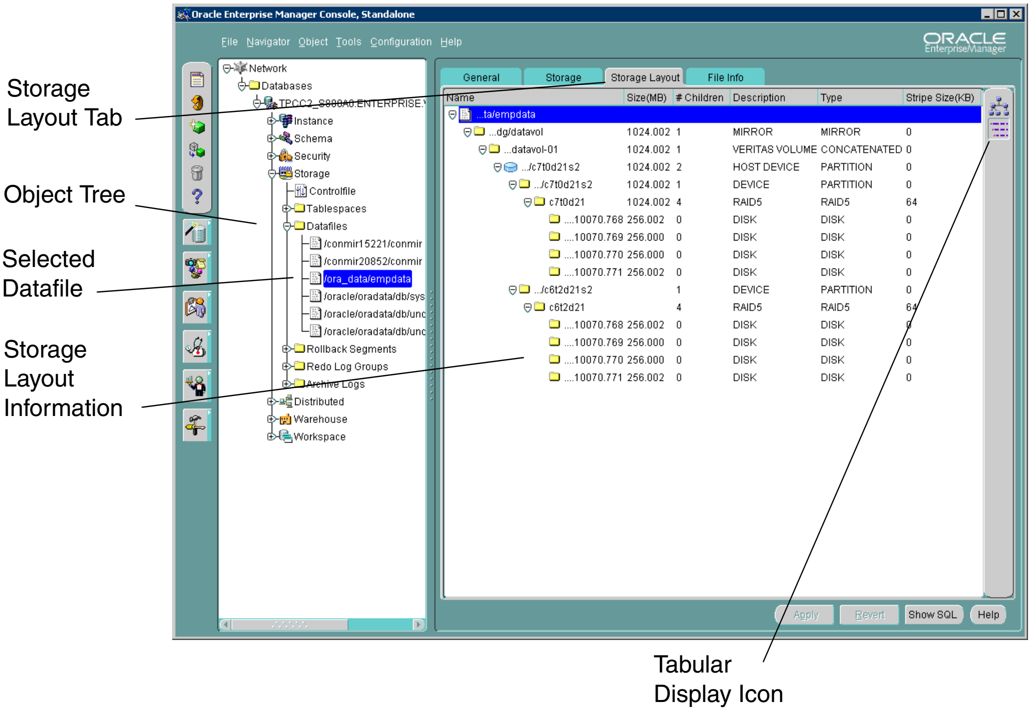
6. Select the datafile for which you want to view storage layout information.

7. In the right pane, click the **Storage Layout** tab.

8. Expand the objects to display their storage layout.

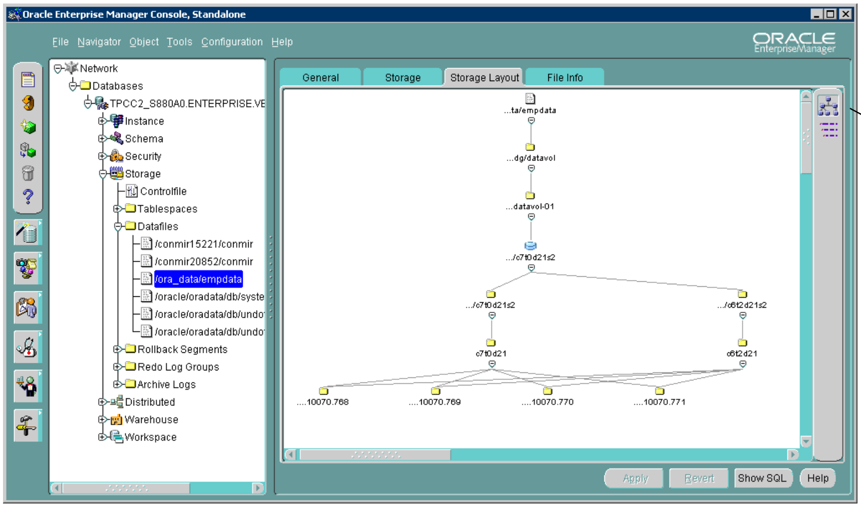
Note Within the Oracle Enterprise Manager Console, you can point to an object on the screen and a description of the object is displayed in a pop-up field. If an object name or path appears truncated, point to it and the pop-up field will display the full object name and path.

Note You can also right click on an object and select **View Details** to see detailed information about the object.



Name	Size(MB)	# Children	Description	Type	Stripe Size(KB)
...ta/empdata					
...dg/datavol	1024.002	1	MIRROR	MIRROR	0
...datavol-01	1024.002	1	VERITAS VOLUME	CONCATENATED	0
.../c7t0d21s2	1024.002	2	HOST DEVICE	PARTITION	0
.../c7t0d21s2	1024.002	1	DEVICE	PARTITION	0
c7t0d21	1024.002	4	RAID5	RAID5	64
....10070.768	256.002	0	DISK	DISK	0
....10070.769	256.000	0	DISK	DISK	0
....10070.770	256.000	0	DISK	DISK	0
....10070.771	256.002	0	DISK	DISK	0
.../c6t2d21s2		1	DEVICE	PARTITION	0
c6t2d21		4	RAID5	RAID5	64
....10070.768	256.002	0	DISK	DISK	0
....10070.769	256.000	0	DISK	DISK	0
....10070.770	256.000	0	DISK	DISK	0
....10070.771	256.002	0	DISK	DISK	0

9. By default, storage layout information is displayed in a tabular format. That is, the **Tabular Display** icon is selected. To view a graphical display of the storage layout, click the **Graphical Display** icon.



Graphical Display Icon

10. Expand the objects to display their storage layout information graphically.

11. To exit, choose **Exit** from the **File** menu.

Configuring Arrays for Storage Mapping and Statistics

VERITAS Storage Foundation *for Oracle* provides "deep" mapping information and performance statistics for supported storage arrays. Deep mapping information consists of identifying the physical disks that comprise each LUN and the hardware RAID information for the LUNs.

Note To use deep mapping, you must have Oracle 9.2.0.3. or later installed.

VERITAS Array Integration Layer (VAIL) software interfaces third-party hardware storage arrays with VERITAS storage software. VAIL providers are software modules that enable VERITAS applications to discover, query, and manage third-party storage arrays.

On Solaris, the following VAIL providers support these third-party storage arrays:

- The `vx_hicommand` provider manages Hitachi arrays.
- The `vx_emc_symmetrix` provider manages EMC Symmetrix arrays.
- The `vx_ibmshark` provider manages IBM ESS (Shark) arrays.

For the most up-to-date array support information, see the appropriate hardware compatibility list (HCL) on the VERITAS Technical Support Web page at:

http://support.veritas.com

If you want to use storage array information accessible through the VAIL providers, install VAIL and perform any required configuration for the storage arrays and VAIL providers. To use deep mapping services and performance statistics for supported storage arrays, you must install both VAIL and VERITAS Mapping Services (VxMS).

You will need to install required third-party array CLIs and APIs on the host where you are going to install VAIL before you install VAIL. If you install any required CLI or API after you install VAIL, rescan the arrays so that VERITAS Storage Foundation *for Oracle* can discover them.

For detailed information about supported array models, see the *VERITAS Array Integration Layer Array Configuration Guide*.

Converting Existing Database Configurations to VxFS 7

You can convert existing database configurations to VERITAS File System. This chapter describes how to migrate UFS file systems, earlier version layouts, and raw devices to current VxFS file systems.

Topics include:

Converting From UFS to VxFS With Quick I/O

If you are currently using UFS file systems, you can use the following procedure to upgrade each file system used by the database to a VxFS file system with Quick I/O.

▼ To convert a UFS file system to VxFS with Quick I/O

1. Shut down the database.

2. Create a backup of the UFS file system.

3. Unmount the UFS file system.

4. Remove the entry for the UFS file system from the `/etc/filesystems` file.

5. Create a VxFS file system of the same size as the original UFS file system, using the mount point where the UFS file system was originally mounted. Use the procedure described in "Creating a VxFS File System" on page 50 to create a VxFS file system.

6. Preallocate Quick I/O files using `qiomkfile`. Use the procedure described in "Creating Database Files as Quick I/O Files Using qiomkfile" on page 72.

7. Restore the backup created in step 2 on page 148 to the Quick I/O files in the new VxFS file system.

8. Restart the database.

Converting From UFS to VxFS For Oracle Disk Manager (Oracle9*i* Only)

If you are currently using UFS file systems, you can use the following procedure to upgrade each file system used by the database to a VxFS file system. You can then use the Oracle Disk Manager feature.

Caution Do not upgrade your root file system to VxFS.

Prerequisites

- Configure Oracle Disk Manager according to "Setting Up VERITAS Extension for Oracle Disk Manager" on page 118.

▼ To convert a UFS file system to VxFS for Oracle Disk Manager

1. Shut down the database.
2. Create a backup of the UFS file system.
3. Unmount the UFS file system.
4. Remove the UFS entry in the `/etc/filesystems` file.
5. Create a VxFS file system of the same size as the original UFS file system, using the mount point where the UFS file system was originally mounted. Use the procedure described in "Creating a VxFS File System" on page 50 to create a VxFS file system.
6. Preallocate ODM files using `odmmkfile`. Make the files the same size or larger than what they were on the original UFS file system.
7. Restore the backup created in step 2 to the new VxFS file system.
8. Restart the database.

Upgrading From Earlier VxFS Version Layouts

Prerequisites

- Perform a full backup of the file system before upgrading to a new disk layout.

Usage Notes

- The `vxupgrade` command lets you to upgrade the VxFS file system disk layout while the file system is mounted. See the `vxupgrade`(1M) manual page for more details.
- VxFS supports three file system disk layouts: Versions 4, 5, and 6. New file systems are created with the Version 6 (for large file systems) disk layout by default when the current version of VERITAS Storage Foundation *for Oracle* is installed on a system. You must minimally upgrade to Version 4 disk layout if you want to use the Storage Rollback or VERITAS NetBackup BLI Backup features.

▼ To upgrade an existing VxFS file system to a new file system disk layout version

Use the `vxupgrade` command to upgrade to Version 4, 5, or 6 disk layout:

```
# /opt/VRTS/bin/vxupgrade -n new_version /mount_point
```

where:

- *`new_version`* is the version of the file system disk layout you want to upgrade to
- *`/mount_point`* is the location where the file system is mounted

Example

This is an example of upgrading to disk layout Version 6.

```
# /opt/VRTS/bin/vxupgrade -n 6 /db01
```

▼ To use Quick I/O after upgrading the file system disk layout to Version 4, 5, or 6

1. Shut down the database.
2. Make each database file accessible as a Quick I/O file.

   See "Accessing Regular VxFS Files as Quick I/O Files" on page 76 for more information.
3. Restart the database.

▼ **To use Oracle Disk Manager after upgrading the file system disk layout to Version 4, 5, or 6**

Simply use the files. Because ODM uses regular files, no setup is required.

Converting From Raw Devices

If the database is currently using raw disks or volumes, use one of the following procedures to use VxFS with the Quick I/O feature or Oracle Disk Manager.

Note The procedure provided assumes the database runs on a single file system after the upgrade.

▼ To convert from raw devices to VxFS with Quick I/O

1. Create a VxFS file system using a size that is 10 percent larger than the original database or total raw device size.

   Use the procedure described in "Creating a VxFS File System" on page 50 to create a new VxFS file system. You can create more file systems based on your performance and availability requirements.

2. Shut down the database.

3. Preallocate Quick I/O files using `qiomkfile`. Use the procedure described in "Creating Database Files as Quick I/O Files Using qiomkfile" on page 72.

4. Copy each raw device file to the new VxFS file system.

   For example, use the `dd` command to copy the file `/dev/rdsk/c0t1d0` to `/db01/dbfile`:

   ```
   $ dd if=/dev/rdsk/c0t1d0 of=/db01/dbfile bs=128k
   ```

5. If the database uses symbolic links to access the database files, change the symbolic links to point to the Quick I/O files.

   For example, if the database has a datafile specification `/data/file1` that was linked to `/dev/rdsk/c0t1d0`, change it to point to the new Quick I/O file:

   ```
   $ rm /data/file1
   $ ln -s /db01/dbfile /data/file1
   ```

6. If the database was using absolute paths to access the database files, rename each file within Oracle before bringing the database online. For example:

   ```
   $ sqlplus "/ as sysdba"
   SQL> startup mount;
   SQL> alter database rename file <filename> to <newfilename>;
   ```

7. Restart the database. For example:

```
SQL> alter database open;
SQL> exit
```

▼ To convert from raw devices to regular files for ODM (Oracle9*i* only)

1. Create a VxFS file system using a size that is 10 percent larger than the original database or total raw device size.

   Use the procedure described in "Creating a VxFS File System" on page 50 to create a new VxFS file system. You can create more file systems based on your performance and availability requirements.

2. Shut down the database.

3. Preallocate the files for ODM using `odmmkfile`.

   ```
   $ /opt/VRTS/bin/odmmkfile -h -s file_size file_name
   ```

   where `-h` creates a file with additional space allocated for the Oracle header and `-s` preallocates a certain amount of space for the file.

4. Copy each raw device file to the new VxFS file system.

   For example, use the `dd` command to copy the file `/dev/rdsk/c0t1d0` to `/db01/dbfile`:

   ```
   $ dd if=/dev/rdsk/c0t1d0 of=/db01/dbfile bs=128k
   ```

5. If the database uses symbolic links to access the database files, change the symbolic links to point to the new files.

   For example, if the database has a datafile specification `/data/file1` that was linked to `/dev/rdsk/c0t1d0`, change it to point to the new file:

   ```
   $ rm /data/file1
   $ ln -s /db01/dbfile /data/file1
   ```

6. If the database was using absolute paths to access the database files, rename each file within Oracle before bringing the database online. For example:

   ```
   $ sqlplus "/ as sysdba"
   SQL> startup mount;
   SQL> alter database rename file file_name to new_file_name;
   ```

7. Restart the database. For example:

```
SQL> alter database open;
SQL> exit
```

Using Storage Checkpoints and Storage Rollback 8

The VERITAS Storage Checkpoint feature is available with the Enterprise Edition 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. This chapter describes what Storage Checkpoints and storage rollback are and how to make use of these technologies through VERITAS Storage Foundation *for Oracle*.

Topics covered in this chapter include:

- "Using Storage Checkpoints and Storage Rollback for Backup and Restore" on page 156
- "Determining Space Requirements for Storage Checkpoints" on page 158
- "Performance of Storage Checkpoints" on page 160
- "Storage Checkpoint Allocation Policies" on page 161
- "Backing Up and Recovering the Database Using Storage Checkpoints" on page 169
- "Cloning the Oracle Instance Using dbed_clonedb" on page 175
- "Guidelines for Oracle Recovery" on page 179
- "Using the VxDBA Utility or GUI to Perform Storage Checkpoint-Related Operations" on page 181

Using Storage Checkpoints and Storage Rollback for Backup and Restore

VERITAS Storage Foundation *for Oracle* provides a Storage Checkpoint facility that is similar to the snapshot file system mechanism; however, a Storage Checkpoint persists after a system reboot. 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 VxDBA utility, the GUI, or the VERITAS Storage Foundation *for Oracle* command line interface (CLI). VERITAS NetBackup also makes use of Storage Checkpoints to provide a very efficient Oracle backup mechanism.

Note For more information on creating Storage Checkpoints with the CLI, see "Managing Storage Checkpoints" on page 352 and "VERITAS Storage Foundation for Oracle Command Line Interface" on page 425.

A direct application of the Storage Checkpoint facility is Storage Rollback. 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 blocks contained in a Storage Checkpoint into the primary file system for restoring the database faster. For more information on Storage Checkpoints and Storage Rollback, see the *VERITAS File System Administrator's Guide*.

Understanding Storage Checkpoints and Storage Rollback

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. 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 or the GUI.

Initially, a Storage Checkpoint contains no data—it contains only 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 needed 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 *n* 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.

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 or the GUI.

Note You must run the `dbed_update` command after upgrading to VERITAS Storage Foundation 4.0 *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.

Moreover, 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`.

Mountable Storage Checkpoints can be used for a wide range of application solutions, including backup, investigations into data integrity, staging upgrades or database modifications, and data replication solutions.

If you mount a Storage Checkpoint as read-write, the VxDBA utility and GUI 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.

Note For more information on mountable Storage Checkpoints, see “Mounting Storage Checkpoints” on page 357 and “VERITAS Storage Foundation for Oracle Command Line Interface” on page 425.

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).

When you use VERITAS NetBackup to back up your database, VERITAS NetBackup creates one set of Storage Checkpoints to provide a consistent view of the file systems for the database backups. The space required to hold this additional set of Storage Checkpoints depends on how busy the database load is when the backup is running. If the database is offline during the entire backup window, there is no additional space required.

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, use the VxDBA utility to set up a Monitoring Agent to monitor file system space usage. When file system space usage exceeds a preset threshold value (say, 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. See "Managing File System Space" on page 380 for more information.

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. See the `vxassist`(1) and `fsadm_vxfs`(1) manual pages for more information.

Performance of Storage Checkpoints

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 like 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 Quick 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.

Storage Checkpoint Allocation Policies

VERITAS File System provides Multi-Volume File Systems (MVS) when used in conjunction with the Volumes Set feature in VERITAS Volume Manager. A volume set is a container for multiple different volumes. MVS enables creation of a single file system over multiple volumes, each volume with properties of its own. This helps administrators specify which data goes on which volume types. For more details about MVS, see *VERITAS Volume Manager Administrator's Guide*. Setting up a storage configuration for MVS operations is a system administrator's responsibility and requires superuser (`root`) privileges.

Multi-Volume File Systems provide, a database administrator, through the SFUA checkpoint administration interface, the ability to create Storage Checkpoint Allocation Policies. A Storage Checkpoint Allocation policy specifies a list of volumes and the order in which to attempt allocations. Once defined, a database administrator can use these policies to:

- Control where the storage checkpoint should be created enabling separation of metadata and data of a storage checkpoint to different volumes.
- Separate storage checkpoints so that data allocated to a storage checkpoint is isolated from the primary file system. This helps control the space used by the checkpoint and prevents the checkpoint from fragmenting the space in the primary fileset.

When policies are assigned to a storage checkpoint, the database administrator must specify the mapping to both metadata and file data. If no policies are specified for the storage checkpoint, the data is placed randomly within the primary file system. Data and metadata of storage checkpoints can have different policies assigned to them or use the same policy to be applied to data and metadata. Multiple checkpoints can be assigned the same checkpoint allocation policy. A *partial* policy is also allowed; a *partial* policy means that the policy does not exist on all file systems used by the database.

Once the policy is assigned to checkpoints, the allocation mechanism attempts to satisfy the request from each device in the policy in the order the devices are defined. If the request cannot be satisfied from any of the devices in the policy, the request will fail, even if other devices exist in the file system which have space. Only those devices can provide allocation that are listed in the policy. This implementation is the mechanism for preventing allocation requests from using space in other devices which are not specified in the policy. It is recommended that you allocate sufficient space for the volumes defined in the Storage Checkpoint policy or update the policy to include additional volumes. This also helps in retaining the old Storage Checkpoints.

Once the assigned policy is deleted, the allocation for metadata and file data for subsequent requests of storage checkpoint will return to the *no policy* assigned state.

Usage Notes

- Since the checkpoint policies feature is associated with MVS file system, it is available only on file systems using disk layout Version 6.
- Storage checkpoint allocation policy requires VxVM Volume Set and VxFS Multi-Volume File Systems features to be enabled. These features are included in the Enterprise Edition of Storage Foundation. Refer to the Multi-Volume File System chapter in the *VERITAS File System Administrator's Guide*, for creating Volume Sets and MVS file systems for the primary file systems used by the database datafiles.
- Data allocation is done by the volumes in the order that was assigned in the policy.
- The maximum length of an checkpoint policy name is 64 characters.

Using Storage Checkpoint Allocation Policies

You can use the `dbed_ckptpolicy` command to administrate the storage checkpoint allocation policies. For detailed information on administrating Storage Checkpoint Policy via the CLI, see "VERITAS Storage Foundation for Oracle Command Line Interface" on page 425.

Note You cannot administer Storage Checkpoint Allocation policies through the VxDBA utility menu or the GUI.

Usage Notes

See the `dbed_ckptpolicy(1M)` and `dbed_ckptcreate(1M)` manual pages for more information. `dbed_ckptpolicy` command need to be executed by the Oracle database administrator.

In the following example, the file systems for database datafiles are set up as follows:

- Two MVS file systems `/mvsfs/v1` and `/mvsfs/v2` used for database datafiles.
- File system `/mvsfs/v1` is created on volume set `mvsvset1`.
- File system `/mvsfs/v2` is created on volume set `mvsvset2`.
- Volume set `mvsvset1` contains volumes `mvsv1`, `mvsv2`, and `mvsv3` .
- Volume set `mvsvset2` contains volumes `mvsv4` and `mvsv5` .

Usage

Use the `dbed_ckptpolicy` command with the following options.

```
$ dbed_ckptpolicy -S ORACLE_SID [ -H ORACLE_HOME ] [-n] [-h] options
```

Where *options* could be any of the following parameters:

```
-o create|update|remove -p ckpt_sample
-o display [-c ckpt_name | -p ckpt_sample]
-o assign -c ckpt_name -p ckpt_data_policy[,ckpt_metadata_policy]
```

Creating a Storage Checkpoint Allocation policy

▼ To create a Storage Checkpoint allocation policy

Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
-o create -p ckpt_policy
```

Note A *partial* policy indicates that the Storage Checkpoint allocation policy does not include all the file systems used by the database.

Example

In the following example, the database name is PROD and the Storage Checkpoint allocation policy that is created is named ckpt_sample.

```
$ dbed_ckptpolicy -S PROD -o create -p ckpt_sample
```

Output similar to the following is displayed.

```
File System: /mvsfs/v2 (MVS volumes: mvsv4,mvsv5)
Assigned Data Policy: NONE (MVS Volumes: N/A)
Assigned Meta Data Policy: NONE (MVS Volumes: N/A)
Please enter the volume name(s), separated by space, for the policy
ckpt_sample [skip,quit]: mvsv4

File System: /mvsfs/v1 (MVS volumes: mvsv1,mvsv2,mvsv3)
Assigned Data Policy: NONE (MVS Volumes: N/A)
Assigned Meta Data Policy: NONE (MVS Volumes: N/A)
Please enter the volume name(s), separated by space, for the policy
ckpt_sample [skip,quit]: mvsv2

The following information will be used to create policy ckpt_sample
ckpt_sample           /mvsfs/v2           mvsv4
ckpt_sample           /mvsfs/v1           mvsv2
```

Assigning a Storage Checkpoint Allocation policy

You can use either of the following methods to assign an allocation policy to a Storage Checkpoint:

- Use the `dbed_ckptpolicy` command to assign allocation policies to an existing storage checkpoint.
- Use `[ -p ckpt_data_policy[,ckpt_metadata_policy]]` option in `dbed_ckptcreate` command to supply policies when executed.

Note `dbed_ckptcreate` command automaticlly assigns the policies when the storage checkpoint is created.

To assign an allocation policy to an existing Storage Checkpoint

The following procedure uses `dbed_ckptpolicy` to assign an allocation policy to an existing Storage Checkpoint. This example uses PROD as the database name and Checkpoint_1096060202 as a sample Storage Checkpoint.

1. Create an online Storage Checkpoint for database PROD.

```
$ dbed_ckptcreate -S PROD -H ORACLE_HOME -o online
```

   As a result, Checkpoint_1096060202 is created.

2. Assign a Storage Checkpoint policy to the Checkpoint_1096060202.

```
$ dbed_ckptpolicy -S PROD -n -o assign -c Checkpoint_1096060202 \
 -p ckpt_data,ckpt_metadata
```

3. Display the details of the Storage Checkpoint allocation policy assigned to Checkpoint_1096060202.

```
$ dbed_ckptpolicy -S PROD -n -o display -c Checkpoint_1096060202
```

   Output similar to the following is displayed:

```
Storage Checkpoint   File System   Data Policy   Meta Data Policy
------------------   ------------  ------------ ------------------
Checkpoint_1096060202 /mvsfs/v2    ckpt_data     ckpt_metadata
Checkpoint_1096060202 /mvsfs/v1    ckpt_data     ckpt_metadata
```

To assign an allocation policy to a new Storage Checkpoint

1. Use `dbed_ckpcreate` to assign an allocation policy to a new Storage Checkpoint.

   ```
   $ dbed_ckptcreate -S PROD -H -o online -p ckpt_data,ckpt_metadata
   ```

   As a result, a Storage Checkpoint allocation policy is assigned to Checkpoint_1096060122, which is a new Storage Checkpoint.

2. Display the details of the Storage Checkpoint allocation policy assigned to checkpoint 1096060122.

   ```
   $ dbed_ckptpolicy -S PROD -n -o display -c Checkpoint_1096060122
   ```

Output similar to the following is displayed:

```
Storage Checkpoint   File System   Data Policy   Meta Data Policy
------------------   ------------  -----------   ----------------
Checkpoint_1096060122 /mvsfs/v2    ckpt_data     ckpt_metadata
Checkpoint_1096060122 /mvsfs/v1    ckpt_data     ckpt_metadata
```

Displaying a Storage Checkpoint Allocation policy

To display a Storage Checkpoint allocation policy

Use the `-o display` option to list all the Storage Checkpoint allocation policies contained in the file systems used by the database.

Usage

```
$ dbed_ckptpolicy -S ORACLE_SID -n -o display
```

Output similar to the following is displayed:

```
Policy Name          File System Coverage
------------------   --------------------
ckpt                 Complete
ckpt_data            Complete
ckpt_metadata        Complete
new_ckpt             Partial
ckpt_sample          Complete
```

Note `Partial` in the File System Coverage column indicates that the Storage Checkpoint allocation policy does not include one or more of the file systems used by the database.

▼ To display Storage Checkpoint Allocation Policy information

Use the `-o display -p` *`checkpointpolicy_name`* option to display information related to a specific Storage Checkpoint allocation policy.

Usage

```
$ dbed_ckptpolicy -S ORACLE_SID -n -o display -p checkpointpolicy_name
```

Output similar to the following is displayed:

```
Policy Name     File System   MVS volumes
------------    -----------   -----------
ckpt_sample     /mvsfs/v2     mvsv4
ckpt_sample     /mvsfs/v1     mvsv2
```

▼ To display the allocation policies assigned to a Storage Checkpoint

Use the `-o display -c` *`checkpoint_xxxxxxxxx`* option to display the allocation policies assigned to the Storage Checkpoint.

Usage

```
$ dbed_ckptpolicy -S ORACLE_SID -n -o display -c Checkpoint_xxxxxxxxxx
```

Output similar to the following is displayed:

```
Storage Checkpoint      File System   Data Policy   Meta Data Policy
---------------------   -----------   -----------   ---------------
Checkpoint_1095125037   /mvsfs/v2     ckpt_data     ckpt_metadata
Checkpoint_1095125037   /mvsfs/v1     ckpt_data     ckpt_metadata
```

Updating a Storage Checkpoint Allocation Policy

▼ To update a Storage Checkpoint Allocation Policy

Use the `-o update -p` *`checkpoint_policy_name`* option to update an allocation policy.

Example

```
$ dbed_ckptpolicy -S ORACLE_SID -n -o update -p checkpointpolicy_name
$
```

Output similar to the following is displayed:

```
File System: /mvsfs/v2 (MVS volumes: mvsv4,mvsv5)
Policy: ckpt_sample (MVS volumes: mvsv4)
Please enter the volume name(s), separated by space, for the policy
ckpt_sample [skip,quit]: mvsv5

File System: /mvsfs/v1 (MVS volumes: mvsv1,mvsv2,mvsv3)
Policy: ckpt_sample (MVS volumes: mvsv2)
Please enter the volume name(s), separated by space, for the policy
ckpt_sample [skip,quit]: mvsv2,mvsv3

The following information will be used to create policy ckpt_sample
ckpt_sample     /mvsfs/v2     mvsv5
ckpt_sample     /mvsfs/v1     mvsv2,mvsv3
where mvsv4 is the volume currently using the allocation policy.
```

Note The output displays the volumes that are currently assigned in the Storage Checkpoint allocation policy.

Note You are prompted to enter any new volumes to which you would want to assign the Storage Checkpoint allocation policy.

Removing a Storage Checkpoint Allocation Policy

▼ To remove a Storage Checkpoint Allocation Policy

Use the following command to remove a Storage Checkpoint allocation policy:

```
$ dbed_ckptpolicy -S ORACLE_SID -n -o remove -p checkpointpolicy_name
```

Converting from Regular VxFS file system to MVS

The following procedure describes the conversion of a regular VxFS file system to MVS file system and optionally add new volume to it. Converting a regular VxFS to MVS requires superuser (`root`) privileges. For further details on creating and administrating the VxVM Volume Sets and VxFS Multi-Volume Files System, refer to the *VERITAS Volume Manager Administrator's Guide* and *VERITAS File System Administrator's Guide*.

▼ To convert from a regular VxFS file system to MVS:

1. Select a non-MVS file system to convert to MVS and unmount it.

   ```
   # umount /mnt1
   ```

2. Create the volume set.

```
# vxvset -g dgname make myvset old_vol
```

3. Mount the volume set.

```
# mount -F vxfs /dev/vx/dsk/dgname/myvset /mnt1
```

4. Upgrade the volume set's file system to Version 6 disk layout. See *VERITAS File System Installation Guide* and the `vxfsconvert(1M)` and `vxupgrade(1M)` manual pages for information on upgrading VxFS disk layouts.

```
# vxupgrade -n 6 /mnt1
```

5. Add the new volume to the volume set.

```
# vxvset -g dgname addvol myvset new_vol
```

6. Add the new volume to the file system. You must specify the size of the volume.

```
# fsvoladm add /mnt1 new_vol 2g
```

   where `new_vol` is the name of the newly added volume and `2g` is the size of the volume.

7. Verify the new volume in the file system.

```
# fsvoladm list /mnt1
```

Backing Up and Recovering the Database Using Storage Checkpoints

Storage Checkpoints can be created by specifying one of the following options: online, offline, or instant. To create a Storage Checkpoint with the online option, the database should be online and you must enable ARCHIVELOG mode for the database. 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.

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 Using the Command Line

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 using the procedure below.

Usage Notes

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

▼ **To verify that a Storage Checkpoint is error-free using the command line**

1. Create and mount a Storage Checkpoint:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD -H /oracle/product -o
online
Creating online Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_903937870 created.

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

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

2. Examine the content of the Storage Checkpoint:

```
$ ls -l /tmp/ckpt_ro/dbvol_82/dbinst1
drwxr-xr-x 3  oracle  dba  1024       Nov 11 2000 .
drwxr-xr-x 3  oracledba512       Nov 16 11:00 ..
-rw-r--r-- 1  oracle  dba  209747968  Nov 16 10:58 .tstmp
-rw-r--r-- 1  oracledba209747968Nov 16 10:58 .tstab
lrwxrwxrwx 1  oracledba18        Nov 11 2000 tstmp -> \
                                    .tstmp::cdev:vxfs:
lrwxrwxrwx 1  oracledba 18         Nov 11 2000 tstab -> \
                                    .tstab::cdev:vxfs:
```

3. Run `dbv` tool against Quick I/O file `tstmp`:

```
$ dbv file=/tmp/ckpt_ro/db01/tstmp

DBVERIFY: Release 8.1.5.0.0 - Production on Sun Nov 16 11:53:33 2014

(c) Copyright 1999 Oracle Corporation.  All rights reserved.

DBVERIFY - Verification starting: FILE = \
  /tmp/ckpt_ro/db01/PROD/tstmp

DBVERIFY - Verification complete

Total Pages Examined         : 102400
Total Pages Processed (Data) : 5077
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 2049
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 1681
Total Pages Empty            : 93593
Total Pages Marked Corrupt   : 0
Total Pages Influx           : 0
```

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 kept on a separate medium. 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.

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.

▼ To back up a frozen database image using the command line

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

1. Create a Storage Checkpoint using the `dbed_ckptcreate` command:

```
    $ /opt/VRTS/bin/dbed_ckptcreate -S PROD -H /oracle/product -o
online
    Creating online Storage Checkpoint of database PROD.
    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
```

Note If the specified mount point directory does not exist, then `dbed_ckptmount` 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.

Note 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 incomplete recovery of the database to the point right before the control files have recorded the addition or removal of datafiles. To provide recovery options, a backup copy of the control file for the database is saved under the `/etc/vx/vxdba/`*`$ORACLE_SID`*`/checkpoint_dir/`*`CKPT_NAME`* directory immediately after a Storage Checkpoint is created. You can use this file to assist with database recovery, if necessary. If possible, both ASCII and binary versions of the control file will be left under the `/etc/vx/vxdba/`*`$ORACLE_SID`*`/checkpoint_dir/`*`CKPT_NAME`* directory. The binary version will be compressed to conserve space. 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. Ensure that the affected datafiles, tablespaces, or database are offline, and 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.

2. Start up the database instance if it is down.

3. Use **`recover database until cancel`**, **`recover database until change`**, or **`recover database until time`** to re-apply archive logs to the point before the table was deleted to recover the database to 4:00 p.m.

4. Open the database with **`alter database open resetlogs`**.

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

6. Create a new Storage Checkpoint.

Note For more information on storage rollback and recovering to Storage Checkpoints, see "Rolling Back the Database to a Storage Checkpoint" on page 364, "Rolling Back a Tablespace to a Storage Checkpoint" on page 367, and "Rolling Back Datafiles to a Storage Checkpoint" on page 372.

Cloning the Oracle Instance Using dbed_clonedb

You can use the VERITAS Storage Foundation *for Oracle* dbed_clonedb command to clone an Oracle instance using mountable and writable Storage Checkpoints to the same or different instance so the instance can coexist. You can also create a clone instance using a Storage Checkpoint that is not mounted.

You have the option to manually or automatically recover the Oracle database when using the dbed_clonedb command:

- Manual (interactive) recovery, which requires using the -i option, of the clone instance 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 dbed_clonedb command, recovers the entire database and replays all of the archive logs. You will not be prompted for any archive log names.

Prerequisites

- You may be logged in as either the database administrator or root.
- Make sure you have enough space to create a clone database on your system.

  A clone database takes up as much memory and machine resources as the primary database.
- You must first create a writable Storage Checkpoint. (See "Creating Storage Checkpoints Using dbed_ckptcreate" on page 438.)
- If you choose to use an existing Storage Checkpoint to create the clone database, the Storage Checkpoint needs to be online.

Usage Notes

- The dbed_clonedb command is used to create a copy of a Oracle database, cloning all existing database files to new locations. This is required when using mountable, writable Storage Checkpoints, where a new Oracle database needs to be started on the same host as an existing database.
- The utility requires that the current environment be configured correctly for the existing Oracle database which has had a Storage Checkpoint created underneath it. This means that the ORACLE_SID and ORACLE_HOME environment variables must be set correctly.
- It is assumed that the user has a basic understanding of the Oracle recovery process.
- See the dbed_clonedb(1M) manual page for more information.

Options

`-S` Specifies the name of the new Oracle SID, which will be the name of the new database instance.

`-m` Indicates the new mount point of the Storage Checkpoint.

`-c` 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)

-d This option is only for use 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.

`-o` The `-o umount` option shuts down the clone database and unmounts the Storage Checkpoint file system. The `-o restartdb` option mounts the Storage Checkpoint file system and starts the clone database.

▼ To clone an Oracle instance with manual Oracle recovery

7. Use the `dbed_clonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -m /local/oracle9/1 \
  -c Checkpoint_988813047 -i
  Primary Oracle SID is TEST9i
  New Oracle SID is NEW9
  Checkpoint_988813047 not mounted at /local/oracle9/1
  Mounting Checkpoint_988813047 at /local/oracle9/1
   Using environment-specified parameter file
        /local/oracle9/links/dbs/initTEST9i.ora
   Default Oracle parameter file found:
        /local/oracle9/links/dbs/initTEST9i.ora
   Copying /local/oracle9/links/dbs/initTEST9i.ora
        to /local/oracle9/1/testvol
   Control file 'ora_control2'
        path not explicitly specified in init file; assuming
        ORACLE_HOME/dbs

   All redo-log files found
   Copying initTEST9i.ora to initNEW9.ora
        in /local/oracle9/1/testvol
   Altering db_name in initNEW9.ora
   Altering control file locations in initNEW9.ora
   Creating new link for clone database init file
   Creating archive log directory

   About to start up new database and begin reconfiguration

   Database NEW9 is being reconfigured
   Altering clone database archive log directory
   Updating log_archive_dest in clone database init file
   Found archive log destination at /testvol

   The latest archive log(s) must now be applied. To apply
    the logs, open a new window and perform the following steps:

1. copy required archive log(s) from primary to clone:
   primary archive logs in /testvol
   clone archive logs expected in /local/oracle9/1/testvol
2. ORACLE_SID=NEW9; export ORACLE_SID # sh and ksh, OR
   setenv ORACLE_SID NEW9 #csh
3. /local/oracle9/links/bin/sqlplus /nolog
4. CONNECT / AS SYSDBA
5. RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
6. enter the archive log(s) you wish to apply
7. EXIT

Press <Return> after you have completed the above steps.
<Return>

   Resetting logs on new database NEW9
   Database instance NEW9 is up and running
```

▼ To clone an Oracle instance with automatic Oracle recovery

8. Use the dbed_clonedb command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -m /local/oracle9/1 \
  -c Checkpoint_988813047
   Primary Oracle SID is TEST9i
   New Oracle SID is NEW9
   Checkpoint_988813047 not mounted at /local/oracle9/1
   Mounting Checkpoint_988813047 at /local/oracle9/1
   Using environment-specified parameter file
        /local/oracle9/links/dbs/initTEST9i.ora
   Default Oracle parameter file found:
        /local/oracle9/links/dbs/initTEST9i.ora
   Copying /local/oracle9/links/dbs/initTEST9i.ora
        to /local/oracle9/1/testvol
   Control file 'ora_control2'
        path not explicitly specified in init file; assuming
```

```
     ORACLE_HOME/dbs

All redo-log files found
Copying initTEST9i.ora to initNEW9.ora
     in /local/oracle9/1/testvol
Altering db_name in initNEW9.ora
Altering control file locations in initNEW9.ora
Creating new link for clone database init file
Creating archive log directory

About to start up new database and begin reconfiguration
Database NEW9 is being reconfigured
Starting automatic (full) 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 /testvol
Mounting clone database
Resetting logs on new database NEW9
Database instance NEW9 is up and running
```

Guidelines for Oracle Recovery

For optimal Oracle recovery, follow these guidelines:

- Back up all control files before Storage Rollback 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. It is also a good idea to back up the control files before and after making structural changes to databases.

Note The VxDBA utility automatically saves control file and log information when you create a Storage Checkpoint. See "Creating Storage Checkpoints" on page 353 for more information.

- Make sure 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.

Note If you intend to roll back the database to recover from structural changes that you do not want to maintain, you may want to roll back control files. The VxDBA utility saves control file and log information and provides the capability to roll back control files. See "Managing Storage Rollback" on page 362 and "Showing the Backup Control File List" on page 378 for more information.

- Make sure 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.

  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.

- To perform a complete media recovery:

  ```
  SET AUTORECOVERY ON;
  RECOVER DATABASE;
  ```

- To perform an incomplete media recovery, use one of the following:
  - **`RECOVER DATABASE UNTIL CANCEL;`**
  - **`RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss';`**

    (You can confirm the time of error by checking the `$ORACLE_HOME/rdbms/log/alert*.log` file.)
  - **`RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss' using \ backup controlfile;`**
  - **`RECOVER DATABASE UNTIL CHANGE scn;`**

- To open the database after an incomplete media recovery, use the following:
  - **`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.

Caution 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 VxDBA Utility or GUI to Perform Storage Checkpoint-Related Operations

Use the VxDBA utility or GUI to create Storage Checkpoints and then roll back an entire database, a single tablespace, or any set of datafiles using any of the previously created Storage Checkpoints. The VxDBA utility presents a menu of administrative operations and guides you through these tasks, prompting you for the data or information needed to complete each task.

See “Using the VxDBA Utility” on page 333 for a complete description of VxDBA operations and “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267 for detailed information about GUI operations.

Using the Space Capacity Planning Utility for Storage Checkpoints 9

The VERITAS Space Capacity Planning utility for Storage Checkpoints is one of the operations available from the VERITAS Storage Foundation graphical user interface (GUI) and VxDBA utility menus. This chapter describes how to use Storage Checkpoint Capacity Planning using the VxDBA utility and to plan for adequate file system space needed for Storage Checkpoints.

Note This is the last major release of VERITAS Storage Foundation *for Oracle* to support the Space Capacity Planning utility.

Topics include:

Planning File System Space for Storage Checkpoints

VxFS file systems need extra disk space to store Storage Checkpoints. Because VxFS can remove Storage Checkpoints when a file system runs out of space, it is important to ensure that you have adequate space for Storage Checkpoints. The extra space needed depends on how the Storage Checkpoints are used, the number of VxFS changed blocks recorded in the Storage Checkpoints, the frequency with which you plan to create Storage Checkpoints, and how many Storage Checkpoints you want to retain on your system at any given time.

You can use Storage Checkpoint Capacity Planning to simulate various Storage Checkpoint creation and retention models in your production environment, collect the associated file system space-usage information based on these models, and use this information to proactively determine how much additional storage space is needed for Storage Checkpoints.

Storage Checkpoint Capacity Planning uses the `cron` command as the underlying mechanism to run the Capacity Planning schedules you create. You must have the proper access and permissions to create a `cron` job or the Capacity Planning schedule will fail to run.

All Storage Checkpoint Capacity Planning activities, including the file-level block change information, is logged into the `/etc/vx/vxdba/logs/ckptplan.log` log file.

Starting the Storage Checkpoint Capacity Planning Utility

The Storage Checkpoint Capacity Planning utility operations can be run by the Oracle Database Administrator (typically, the user ID `oracle`) of the database instance.

Prerequisites

- You must have the appropriate permissions to run the VxDBA utility. The VxDBA utility requires permission changes to allow database administrators to access it. If you did not make these permission changes when prompted during installation, you can grant administrators access to the VxDBA utility now. See "Starting VxDBA" on page 338.

Usage Notes

- You can *only* use the Storage Checkpoint Capacity Planning utility in an environment that contains *no* Storage Checkpoints created by other tools or products, including:
    - VxDBA utility
    - Command line interface using the `dbed_ckptcreate` command
    - VERITAS NetBackup

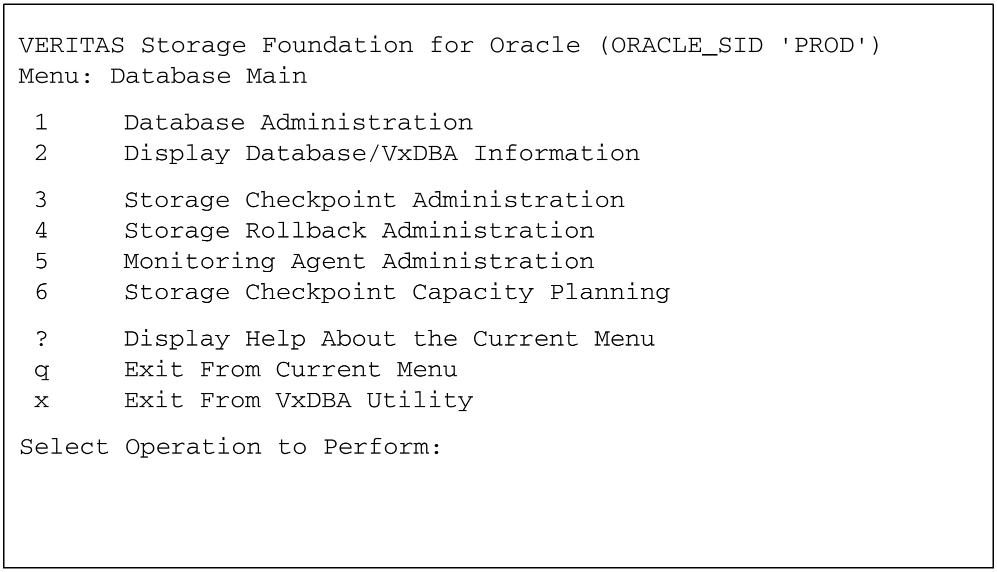
  Each time `cron` attempts to create a Storage Checkpoint at the time designated in the schedule you create, the Storage Checkpoint Capacity Planning utility checks for the presence of Storage Checkpoints created by other tools or products and fails if it detects *any* of these other Storage Checkpoints.

▼ To start the Storage Checkpoint Capacity Planning utility

1. At the administrative prompt, enter:

```
$ /opt/VRTSdbed/bin/vxdba
```

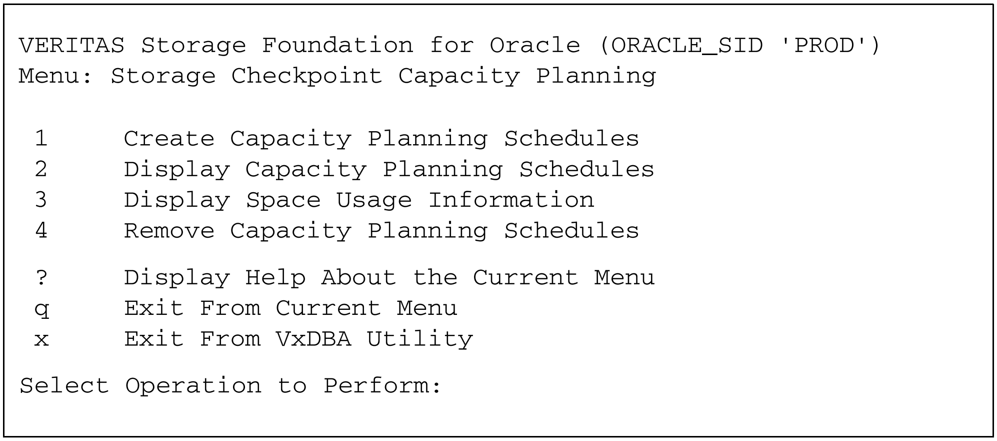
VxDBA starts up and displays the main menu.



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Database Main

 1     Database Administration
 2     Display Database/VxDBA Information

 3     Storage Checkpoint Administration
 4     Storage Rollback Administration
 5     Monitoring Agent Administration
 6     Storage Checkpoint Capacity Planning

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

2. Type **6** to select **Storage Checkpoint Capacity Planning**.

   VxDBA displays the Storage Checkpoint Capacity Planning utility operations:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Capacity Planning

 1     Create Capacity Planning Schedules
 2     Display Capacity Planning Schedules
 3     Display Space Usage Information
 4     Remove Capacity Planning Schedules

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Creating Capacity Planning Schedules

The VERITAS Storage Foundation GUI and the Storage Checkpoint Capacity Planning utility prompt you through the entire schedule-creation process.

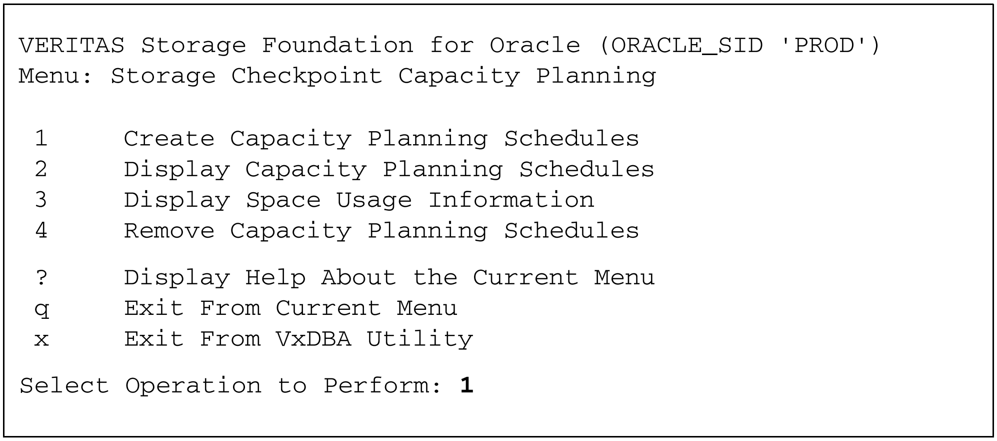
Prerequisites

◆ You must have the appropriate permissions to create and execute a `cron` job to create Capacity Planning schedules.

  For more information on setting up and using `cron`, see the `cron`(1) and `crontab`(1) manual pages.

▼ To create Capacity Planning schedules using the VxDBA utility

1. From the Storage Checkpoint Capacity Planning menu, type **1** to select **Create Capacity Planning Schedules**:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Capacity Planning

 1     Create Capacity Planning Schedules
 2     Display Capacity Planning Schedules
 3     Display Space Usage Information
 4     Remove Capacity Planning Schedules

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform: 1
```

2. Select the type of schedule you want to create:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Create Capacity Planning Schedules

 1     Create Quick Planning Schedule (Current Instance)
 2     Create Custom Planning Schedule (List of File Systems)
 3     Create Complete Planning Schedule (All File Systems)

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform: 1
```

Select from the following operations:

Operation	**Description**
Create Quick Planning Schedule	Creates a schedule for the VxFS file systems associated with the current database instance's datafiles.
Create Custom Planning Schedule	Creates a schedule for the VxFS file systems listed in a user-supplied list file.
Create Complete Planning Schedule	Creates a schedule for all VxFS file systems on the host.

3. Supply the schedule-creation information when prompted. When you finish entering the schedule information, the Storage Checkpoint Capacity Planning utility displays the schedule you created and lets you confirm or edit it.

Example

To create a Quick Planning Schedule, type **1** on the **Create Capacity Planning Schedules** menu:

```
----------------------------------------
VxDBA: Create Quick Planning Schedule
----------------------------------------

NOTICE: To correctly create schedule on current database instance,
you must have up-to-date tablespace information stored in VxDBA's
database.
Are you certain the tablespace information is up-to-date? [y,n,q,?] y

How often do you want to create Storage Checkpoints?
[d(daily),w(specify days of week),m(specify days of month),q]
(default: d) d

Specify the hours of the day to create Storage Checkpoints,
where 00 is midnight and 23 is 11:00 p.m..
Use ',' to separate multiple entries.  [00-23,q] (default: 00) 01

On what date do you want to start this Capacity Planning schedule?
[yyyy-mm-dd,q] (default: 2001-12-10) 2001-12-15

On what date do you want to end this Capacity Planning schedule?
[yyyy-mm-dd,q] (default: 2001-12-10) 2001-12-15

Do you want to remove all Storage Checkpoints created when
this Capacity Planning schedule ends? [y,n,q,?] (default: y) y

You created the following schedule for Capacity Planning:

  Start date: 2001-12-15 End date: 2001-12-15
  You set the remove Storage Checkpoints option to: 'y'
  You specified Storage Checkpoints are to be created as follows:
  Daily
  On the following hour(s) of the day:  (1 a.m.)

Press <Return> to confirm this schedule, 'm' to modify
the schedule, or 'q' to quit. [<Return>,m,q] <Return>

Press <Return> to activate this schedule, 'a' to add a new
schedule, or 'q' to quit. [<Return>,a,q]? <Return>
```

For information on creating capacity planning schedules using the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

Displaying Capacity Planning Schedules

Use the GUI, the VxDBA utility, or the Storage Checkpoint Capacity Planning utility to display all the Capacity Planning schedules you created.

▼ To display Capacity Planning schedules using the VxDBA utility

From the Storage Checkpoint Capacity Planning menu, type **2** to select **Display Capacity Planning Schedules**:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Capacity Planning

 1     Create Capacity Planning Schedules
 2     Display Capacity Planning Schedules
 3     Display Space Usage Information
 4     Remove Capacity Planning Schedules

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform: 2
```

The Storage Checkpoint Capacity Planning utility displays all the schedules you created.

Example

To display the Quick Planning Schedule you created in "Creating Capacity Planning Schedules" on page 187, type **2**:

```
---------------------------------------------
VxDBA: Display Capacity Planning Schedules
---------------------------------------------

#    Start Date  End Date    Schedule Summary
--- ----------- ----------- ---------------------------------------
1   2001-12-15  2001-12-15   Daily at 1 a.m.

Press <Return> to continue...
```

For information on displaying capacity planning schedules using the GUI, see "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267.

Displaying File System Space Usage Information

Use the GUI, the VxDBA utility, or the Storage Checkpoint Capacity Planning utility to display space-usage information for VxFS file systems and the associated Storage Checkpoints. You can monitor this space-usage information as your Storage Checkpoint-creation schedules progress.

Usage Notes

- If a Storage Checkpoint is created using other tools and products (for example, through the VxDBA utility menus or VERITAS NetBackup), the following will occur when a Storage Checkpoint Capacity Planning schedule is in progress:
  - The Storage Checkpoint Capacity Planning utility will fail the next time `cron` attempts to create a Storage Checkpoint at the time designated in the schedule you created.
  - The Storage Checkpoint Capacity Planning utility will display the following error when displaying the Storage Checkpoint space information using the **Display Space Usage Information** operation:

    ```
    DBED1007: Non-Capacity Planning Storage Checkpoints detected.
    ```

▼ To display VxFS file system and Storage Checkpoint space usage using the VxDBA utility

1. From the Storage Checkpoint Capacity Planning menu, type **3** to select **Display Space Usage Information**:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Capacity Planning

 1     Create Capacity Planning Schedules
 2     Display Capacity Planning Schedules
 3     Display Space Usage Information
 4     Remove Capacity Planning Schedules

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform: 3
```

2. Select the kind of space-usage information you want to display:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Display Space Usage Information

 1     Display Space Usage for the Current Instance
 2     Display Space Usage for a List of File Systems
 3     Display Space Usage for All File Systems
 4     Display Space Usage by Schedule

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform: 1
```

Select from the following operations:

Operation	**Description**
Display Space Usage for the Current Instance	Displays space-usage information for the VxFS file systems contained in the current database instance.
Display Space Usage for a List of File Systems	Displays space-usage information for the VxFS file systems listed in a user-supplied list file. You are prompted for the list file name when you select this operation.
Display Space Usage for All File Systems	Displays space-usage information for all VxFS file systems on the host.
Display Space Usage by Schedule	Displays space-usage information for the VxFS file systems by Storage Checkpoint Capacity Planning schedule. You are prompted for the schedule number for which you want to display the space usage.

For information on displaying file system space usage using the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

Example

To display VxFS file system and Storage Checkpoint space-usage information for the current database instance, type **1** on the **Display Space Usage Information** menu:

```
File System (1K block)         FS Size       Used     Avail  %Full
---------------------------------------  --------  --------  -----
/db01                           10.0GB      5.3GB   4.5GB    53.4%

    Storage Checkpoint       Creation Time             Space Needed
   ------------------------ ------------------------- -------------
   Planning_00001_956765641 Wed Oct 27 09:14:01 2001  82.0KB
   Planning_00001_956762040 Wed Oct 27 08:14:00 2001  4.0KB
   Planning_00001_956758441 Wed Oct 27 07:14:01 2001  4.0KB

Total space required by 3 Storage Checkpoint(s) is 90.0KB

Press <Return> to continue...
```

In addition to providing space-usage information for the current database instance's underlying file systems, **Display Space Usage Information for the Current Instance** shows the following information about each Storage Checkpoint it detects:

Field	Description
Storage Checkpoint	All Storage Checkpoints created by the Storage Checkpoints Capacity Planning utility are named using the following conventions: ◆ Prefixed with `Planning_` ◆ Followed by the five digit schedule number, for example `00001_` ◆ Followed by a timestamp sequence number, for example `956758441`
Creation Time	**Creation Time** is the time that the `cron` creates the Capacity Planning Storage Checkpoint.
Space Needed	**Space Needed** is storage space consumed by the Storage Checkpoint based on the changed blocks recorded in the Storage Checkpoint.

Removing Capacity Planning Schedules

Use the GUI, the VxDBA utility, or the Storage Checkpoint Capacity Planning utility to remove Capacity Planning schedules at any time. You do not need to wait until the expiration date that you supplied when creating a schedule.

Usage Notes

- The Storage Checkpoint Capacity Planning utility stores the space-usage information it collects during the schedule duration in the log file `/etc/vx/vxdba/logs/plan`*`xxxxx`*`.out` (where *`xxxxx`* is the schedule ID number). This space-usage information remains available to you in the log file even after the schedule is removed.
- During the schedule-removal operation, you are asked if you want to remove the Storage Checkpoints that were created by the Storage Checkpoint Capacity Planning utility during the schedule duration. If you answer no (**`n`**) to this question, you can remove the associated Storage Checkpoints manually at a later time using the VxDBA utility. See “Removing Storage Checkpoints” on page 361 for more information.

▼ To remove Capacity Planning schedules using the VxDBA Utility

From the Storage Checkpoint Capacity Planning menu, type **4** to select **Remove Capacity Planning Schedules**:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Capacity Planning

 1     Create Capacity Planning Schedules
 2     Display Capacity Planning Schedules
 3     Display Space Usage Information
 4     Remove Capacity Planning Schedules

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform: 4
```

The Storage Checkpoint Capacity Planning utility displays all the existing Storage Checkpoint Capacity Planning schedules, so that you can remove a particular schedule, a range of schedules, or all schedules.

For information on removing Capacity Planning schedules using the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

Example

To remove the Quick Planning Schedule you created in “Creating Capacity Planning Schedules” on page 187, type **4** on the **Storage Checkpoint Capacity Planning** menu:

```
-------------------------------------------
VxDBA: Remove Capacity Planning Schedules
-------------------------------------------

ID Start Date  End Date    Schedule Summary
-- ----------- ---------- ----------------------------------------
1   2001-12-15 2001-12-15   Daily at 1 a.m.

Do you want to delete any of these Capacity Planning schedules?
[y,n,q,?] (default: y) y

Enter a schedule number or a range of numbers to delete.
You can also enter 'all' to remove the entire list of
Capacity Planning schedules. [<number>,<number>-<number>,all,q] 1

ID Start Date  End Date    Schedule Summary
-- ----------- ---------- ----------------------------------------
1   2001-12-15 2001-12-15   Daily at at 1 a.m.

Do you want to delete schedule #1? [y,n,q,?] y

Do you want to remove the Storage Checkpoints created by
Capacity Planning schedule #1? [y,n,q,?] (default: y) y

Generating the output ‘/etc/vx/vxdba/logs/plan00001.out’, please
wait...
Removed schedule ‘00001’ successfully.
```

Using Database FlashSnap for Backup and Off-Host Processing 10

This chapter describes how to use VERITAS Database FlashSnap to create a point-in-time copy of a database for backup and off-host processing. Database FlashSnap allows you to make backup copies of your volumes online with minimal interruption to users.

VERITAS Database FlashSnap is included with VERITAS Storage Foundation Enterprise Edition.

Database FlashSnap lets you capture an online image of an actively changing database at a given instant, known as a snapshot. You can perform backups and off-host processing tasks on snapshots while providing continuous availability of your critical data.

Database FlashSnap commands can be executed from either the command line or the GUI.

Topics covered in this chapter include:

How VERITAS Database FlashSnap Works

You can use VERITAS Database FlashSnap to implement point-in-time copy solutions for enterprise databases. 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.

Database FlashSnap allows database administrators to create a consistent copy of a database without root privileges by creating 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 also allows 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*.

Database FlashSnap can significantly reduce the time it takes to backup your database, increase the availability of your production database, and still maintain your production database's performance.

Note To use Database FlashSnap, you must have VERITAS Storage Foundation Enterprise Edition 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. See "Preparing Hosts and Storage for Database FlashSnap" on page 204, for more information.

Typical Problems Database FlashSnap Solves

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 not central to their job function or 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 it does not require root privileges, Database FlashSnap overcomes these obstacles by enabling database administrators to create consistent snapshots of the database more easily. The snapshots can be utilized for repetitive use.

Database FlashSnap Applications

The following are typical applications of VERITAS Database FlashSnap:

- *Database Backup and Restore*: Enterprises require 24/7 online data availability. They cannot afford the downtime involved in backing up critical data offline. By creating a clone database or a duplicate volume snapshot of data, and then using it to back up your data, your business-critical applications can continue to run without extended down time or impacted performance. After a clone database or snapshot volume is created, it can be used as a source to back up the original database.
- *Decision-Support Analysis and Reporting*: Operations such as decision-support analysis and business reporting may not require access to real-time information. You can direct such operations to use a clone database that you have created from snapshots using VERITAS Database FlashSnap, rather than allowing them to compete for access to the primary volume or database. When required, you can quickly resynchronize the clone database with the primary database to get up-to-date information.
- *Application Development and Testing*: Development or service groups can use a clone database created with Database FlashSnap as a test database for new applications. A clone database provides developers, system testers, and quality assurance groups with a realistic basis for testing the robustness, integrity, and performance of new applications.
- *Logical Error Recovery*: Logical errors caused by an administrator or an application program can compromise the integrity of a database. You can recover a database by restoring the database files from a volume snapshot or by recovering logical objects (such as tables, for example) from a clone database created from volume snapshots using Database FlashSnap. These solutions are faster than fully restoring database files from tape or other backup media.

Using Database FlashSnap

The system administrator needs to configure storage according to the requirements specified in the snapplan. For information on configuring storage, see "Preparing Hosts and Storage for Database FlashSnap" on page 204.

Database FlashSnap allows you to check the storage setup against requirements set forth in the snapplan. Depending on the results, the database administrator may need to modify the snapplan or the system administrator may need to adjust the storage configuration. Properly configuring storage is the only aspect of using Database FlashSnap that requires the system administrator's participation.

To use Database FlashSnap, a database administrator must first define their snapshot requirements. For example, they need to determine whether off-host processing is required and, if it is, which host should be used for it. In addition, it is also important to consider how much database downtime can be tolerated. Database snapshot requirements are defined in a file called a *snapplan*. The snapplan specifies snapshot options that will be used when creating a snapshot image (such as whether the snapshot mode will be `online`, `offline`, or `instant`). For further details, see "Creating a Snapplan (dbed_vmchecksnap)" on page 223.

After creating the snapplan, the database administrator must validate it to ensure that it is correct. During validation the snapplan is copied to the repository before using it to create a snapshot. Depending on the validation results, the database administrator may need to modify the snapplan or the system administrator may need to adjust the storage configuration. Properly configuring storage is the only aspect of using Database FlashSnap that requires the system administrator's participation.

After storage is configured as specified in the snapplan and the snapplan has been validated, the database administrator can create snapshots of the database and create database clones based on the snapshots on either the same host or a secondary one.

A database clone can be used on a secondary host for off-host processing, including decision-support analysis and reporting, application development and testing, database backup, and logical error recovery. After a user has finished using the clone on a secondary host, the database administrator can shut down the clone and move the snapshot database back to the primary host. Regardless of whether a snapshot is used on the primary or secondary host, it can be resynchronized with the primary database using Database FlashSnap. Database FlashSnap utilizes VERITAS Volume Manager FastResync to quickly resynchronize the changed section between the primary and snapshot. Refer to the *VERITAS Volume Manager 4.1 User's Guide* for details about the Volume Manager FastResync.

Database FlashSnap can also be used to recover the primary copy of the database if it becomes corrupted by overwriting it with the snapshot. You can recover the primary database with a snapshot using the reverse resynchronization functionality of Database FlashSnap.

Database FlashSnap Commands

The Database FlashSnap feature consists of three commands:

- `dbed_vmchecksnap` (used on the primary host)

  Creates and validates the snapshot plan used to create a snapshot image of an Oracle database. You can also use `dbed_vmchecksnap` to copy, list, or remove a snapplan or make sure the storage is configured properly for the task. `dbed_vmchecksnap` is also used on the secondary host to list the snapplan.

- `dbed_vmsnap` (used on the primary host)

  Creates a snapshot image of an Oracle database by splitting the mirror volumes used by the database. You can also use `dbed_vmsnap` to resynchronize snapshot volumes with their original volumes. The command also allows you to resynchronize the original volumes from the data in the snapshot volumes, which is useful if the original volumes become corrupted. Resynchronizing the original volumes from the snapshot volumes is known as *reverse resynchronization*.

- `dbed_vmclonedb` (used on the primary or secondary host)

  Mounts and starts a clone database using snapshot volumes. It can also shut down a clone database and deport its volumes, as well as restart a clone database that has been shut down. The snapshot image can be brought up on the same host running the primary database or on a secondary host.

All of these commands can be executed by the Oracle database administrator and do not require superuser (`root`) privileges.

Note Database FlashSnap operations can be executed from either the command line or the GUI.

Database FlashSnap Options

Database FlashSnap offers three options for creating database snapshots. The option you choose is specified in the snapplan.

- *online*

  With this option, the tablespaces are put into online backup mode before the snapshot is created. This type of snapshot is also a valid database backup. Select this option if you are performing a point-in-time recovery from logical errors.

- *instant*

  With this option, the database can be up and running, and the tablespaces do not need to be put into online backup mode before the snapshot is created. However, all the file systems used by the database (including those containing the online redo logs and control files) are temporarily frozen and the cache is flushed before the snapshot is created. By freezing the file systems, the snapshot will be a consistent point-in-time image of the database from which a database clone can be created.

An instant snapshot can be used to guard against data corruption or for off-host decision-support queries. However, it *is not* a valid database backup and cannot be used to perform a point-in-time recovery or off-host backup since tablespaces are not put into online backup mode before the snapshot is created. The instant option is much faster than the online option.

- *offline*

  The `offline` option can be used to clone or back up a database. With this option, the database must be shut down when the snapshot is created. and online redo logs are required. This type of snapshot is a valid database backup.

Note In this release of VERITAS Storage Foundation for Oracle, Database FlashSnap supports third mirror break-off snapshots only. Third mirror break-off snapshots are fully synchronized, full-sized snapshots. See the *VERITAS Volume Manager Administrator's Guide* for more information.

Planning Considerations

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?

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

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

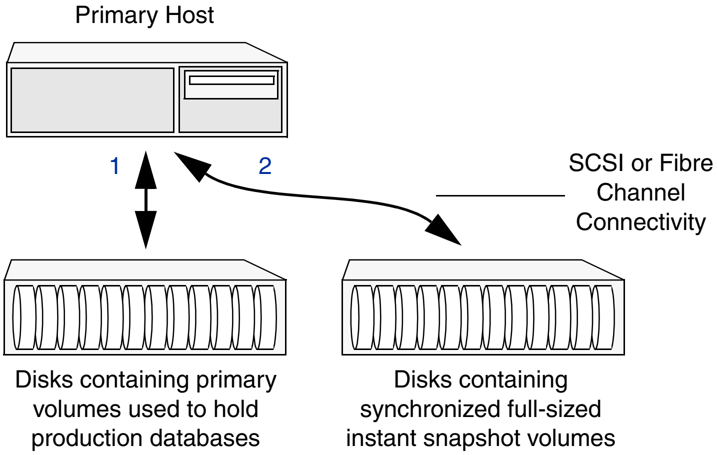
Setting Up Hosts

Database FlashSnap requires sufficient VERITAS Volume Manager disk space, and can be used on the same host that the database resides on (the primary host) or on a secondary host. 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.

Single-Host Configuration

The following figure, "Example of a Database FlashSnap Solution on a Primary Host" on page 204shows the suggested arrangement for implementing Database FlashSnap solutions on the primary host to avoid disk contention.

Example of a Database FlashSnap Solution on a Primary Host

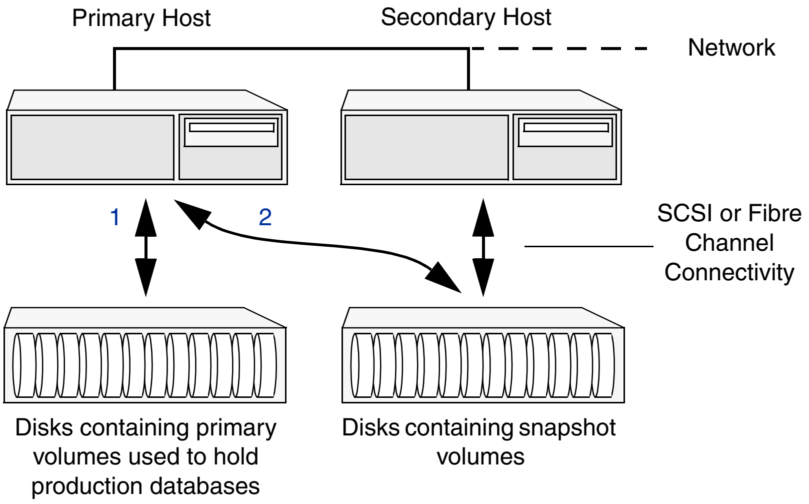

Primary Host
1
2
SCSI or Fibre Channel Connectivity
Disks containing primary volumes used to hold production databases
Disks containing synchronized full-sized instant snapshot volumes

Two-Host Configuration

As shown in the figure below, “Example of an Off-Host Database FlashSnap Solution” on page 205,a Database FlashSnap configuration with two hosts 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. A two-host configuration also allows the snapshot database to avoid contending for I/O resources on the primary host.

For off-host processing applications, both the primary and secondary hosts need to share the storage in which the snapshot database is created. Both the primary and secondary hosts must be able to access the disks containing the snapshot volumes.

Example of an Off-Host Database FlashSnap Solution


Primary Host
Secondary Host
Network
1
2
SCSI or Fibre Channel Connectivity
Disks containing primary volumes used to hold production databases
Disks containing snapshot volumes

Host and Storage Requirements

Before using Database FlashSnap, ensure that:

- 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 on a different VxFS file system than Oracle datafiles 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 disk group.

In addition, before attempting to use Database FlashSnap with two hosts, ensure that:

- The versions of VERITAS Storage Foundation *for Oracle* 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 a VERITAS Storage Foundation 4.1 *for Oracle* Enterprise Edition 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.

You can use the vxsnap CLI command or the GUI to create a snapshot mirror. Creating a snapshot mirror using the GUI is relatively easy. Because the time required to synchronize a snapshot mirror can be long, using the command line is recommended when resynchronizing snapshot mirrors.

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.

Caution The procedure given in this section is for existing volumes without existing snapshot plexes or associated snapshot volumes.

▼ To create a snapshot mirror of a volume or volume set

Note In the following procedure, `volume_name` is the name of either a volume or a volume set.

1. To prepare the volume for being snapshot, use the `vxsnap prepare` command:

```
# vxsnap -g diskgroup prepare volume \
alloc="storage_attribute ..."
```

Note 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.

Note For enabling persistent FastResync on a volume in VxVM 4.1, 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
```

Note There is no option for creating multiple mirrors at the same time. Only one mirror can be created at a time.

5. List the available mirrors:

```
# vxprint -g diskgroup -F%name -e"pl_v_name in \"volume_name\""
```

Note The following two steps enable database FlashSnap to locate the correct mirror plexes when creating snapshots.

6. Set the dbed_flashsnap 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
   ```

7. 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

The following example shows the steps involved in creating a snapshot mirror for the volume `data_vol` belonging to the disk group `PRODdg`.

Prepare the volume `data_vol` for mirroring:

```
# vxsnap -g PRODdg prepare data_vol alloc=PRODdg01
```

Verify that FastResync is enabled:

```
# vxprint -g PRODdg -F%fastresync data_vol
on
```

Verify that a DCO and a DCO log are attached to the volume:

```
# vxprint -g PRODdg -F%hasdcolog data_vol
on
```

Create a snapshot mirror of `data_vol`:

```
# vxsnap -g PRODdg addmir data_vol alloc=PRODdg02
```

List the data plexes:

```
# vxprint -g PRODdg -F%name -e"pl_v_name in \"data_vol\""
data_vol-01
data_vol-02
```

Note 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.

Decide which data plex you want to use and set the *dbed_flashsnap* tag for it:

```
# vxedit -g PRODdg set putil2=dbed_flashsnap data_vol-02
```

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
```

To verify that the snapshot volume was created successfully, use the `vxprint -g <dg>` command as follows:

```
# vxprint -g PRODdg
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 4.1

The procedure described in this section describes how to upgrade a volume created using a version older than VxVM 4.1 so that it can take advantage of database FlashSnap.

Note 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.

Note Existing snapshot volumes created by the `vxassist` command are not supported. A combination of snapshot volumes created by `vxassist` and `vxsnap` are not supported.

▼ 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.

Note 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

In this example, the volume, `data_vol`, is upgraded to make use of VxVM 4.1 features.

Upgrade the disk group, `PRODdg`.

```
# vxdg upgrade PRODdg
```

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
```

Reattach any snapshot volume back to the primary volume to be upgraded.

```
# vxsnap -g PRODdg snapback SNAP-data_vol
```

Turn off FastResync on the volume to be upgraded.

```
# vxvol -g PRODdg set fastresync=off data_vol
```

Disassociate and remove any older DCO object and DCO volumes.

```
# vxassist -g PRODdg remove log  data_vol logtype=dco
```

Upgrade the volume by associating a new DCO object and DCO volume.

```
# vxsnap -g PRODdg prepare data_vol alloc="PRODdg01 PRODdg02"
```

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
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
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
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.

```
# vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 convert
state=SNAPDONE data_vol-04
```

```
# vxprint -g PRODdg
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    -        -       -
```

Upgrading from VERITAS Database Edition 3.5 for Oracle with Database FlashSnap

In this release of VERITAS Storage Foundation *for Oracle*, Database FlashSnap does not support snapshots of `vxdbavol` and `ORACLE_HOME`. If you have upgraded from VERITAS Database Edition 3.5 *for Oracle* with Database FlashSnap, you must remove the volume plexes for `vxdbavol` and `ORACLE_HOME`, and revalidate the snapplan before using Database FlashSnap with this release of VERITAS Storage Foundation *for Oracle*.

▼ To remove the volume plexes for vxdbavol and ORACLE_HOME

1. As root, snapback the snapshot plexes.

```
# vxsnap [-g diskgroup] snapback snapvol
```

2. Turn off FastResync.

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

3. Remove the DCO object.

```
# vxassist [-g diskgroup] remove log volume logtype=dco
```

4. Remove the volume plexes for `vxdbavol` and `ORACLE_HOME`.

```
# vxplex -g diskgroup -o rm dis plex_name
```

5. Log in as the DBA user and revalidate your snapplan.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -H ORACLE_HOME \
-f SNAPPLAN -o validate
```

Summary of Database Snapshot Steps

You can use Database FlashSnap commands to create a snapshot of your entire database on the same host or on a different one. Three types of snapshots can be created: `online`, `offline`, or `instant`.

If the `SNAPSHOT_MODE` specified in the snapplan is set to `online`, `dbed_vmsnap` 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.

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.

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.

Both online and offline snapshots provide a valid backup copy of the database. You can use the snapshot as a source for backing up the database or creating a clone database for decision-support purposes. Instant snapshots *do not represent* a valid backup copy for point-in-time recovery.

The sections that follow explain 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. For more information about the `vxsnap` command, see *VERITAS Volume Manager Administrator's Guide*.

Note Make sure the volumes used by the database are configured properly before attempting to take a snapshot. This requires superuser (`root`) privileges.

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 `dbed_update`.

Note Database FlashSnap commands must be run by the Oracle database administrator.

▼ To create a snapshot image of a database

1. Perform the steps in "Creating a Snapshot Mirror of a Volume or Volume Set Used by the Database" on page 206.

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 VxDBA repository (dbed_vmchecksnap -o remove -f *SNAPPLAN*).
   - Copy a snapplan from the VxDBA repository to your local directory (dbed_vmchecksnap -o copy -f *SNAPPLAN*).

   For information about the snapplan file, see "Creating a Snapplan (dbed_vmchecksnap)" on page 223.

3. Use the dbed_vmsnap command to create snapshot volumes for the database. See "Creating a Snapshot (dbed_vmsnap)" on page 236 for more information.

4. On the secondary host, use the dbed_vmclonedb command to create a clone database using the disk group deported from the primary host. See "Cloning a Database (dbed_vmclonedb)" on page 245 for more information.

   If the primary and secondary hosts specified in the snapplan are different, the dbed_vmclonedb command imports the disk group that was deported from the primary host, recovers the snapshot volumes, mounts the file systems, recovers the database, and brings the database online with a different Oracle SID name than the primary host. If the secondary host is different, the database name can be same. 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.

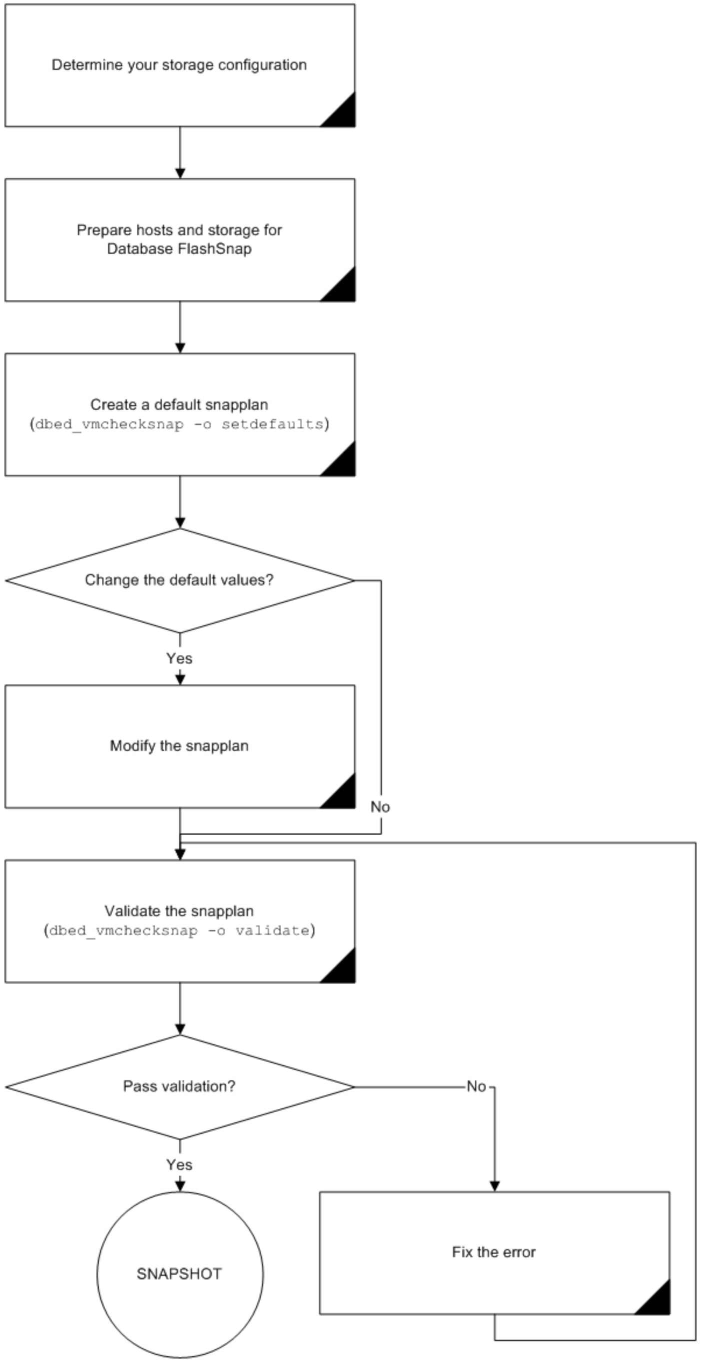
   You can also create a clone on the primary host. Your snapplan settings specify whether a clone should be created on the primary or secondary host.

5. You can now use the clone database to perform database backup and other off-host processing work.

6. 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.

The following flow chart depicts the sequence of steps leading up to taking a snapshot using Database FlashSnap.

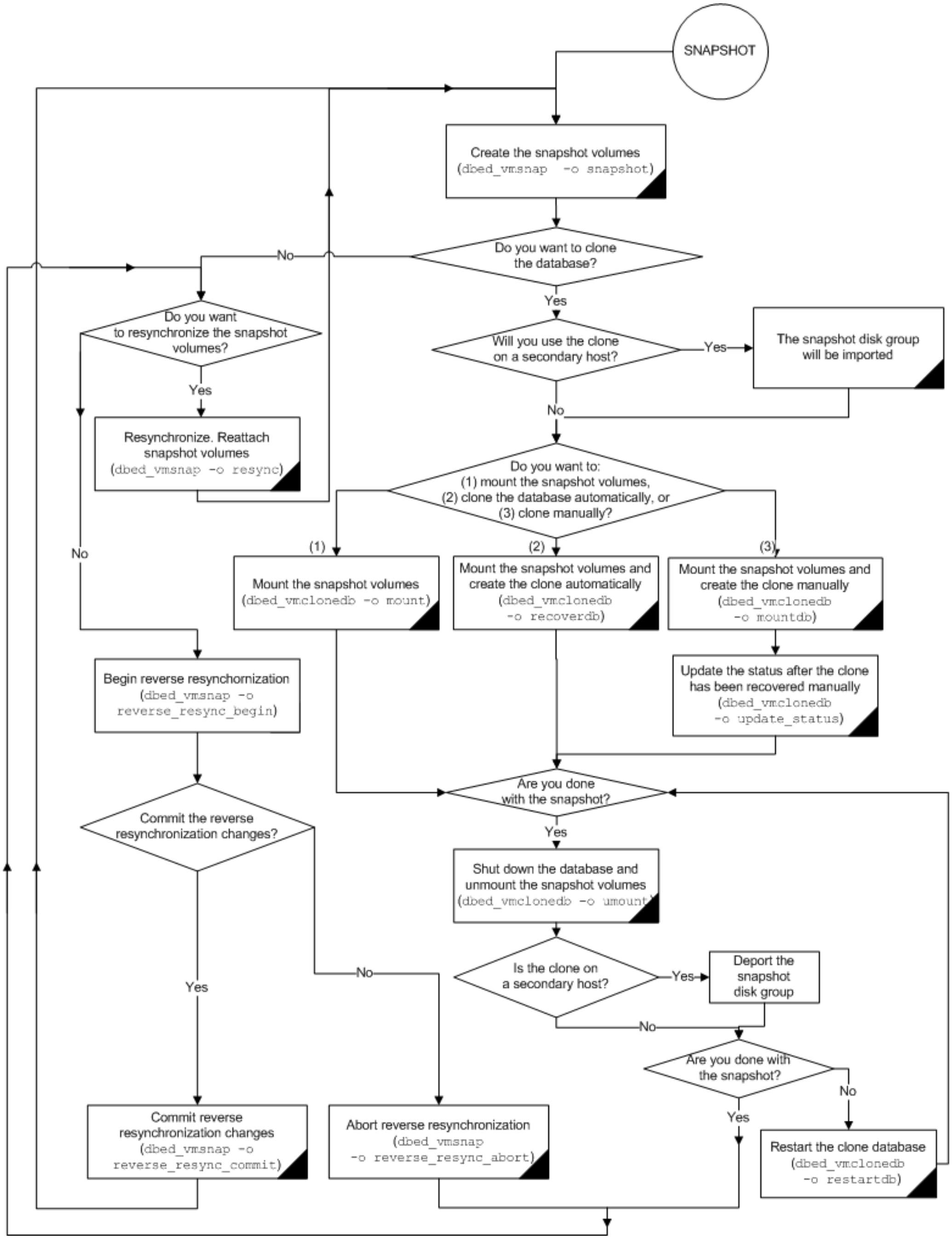
Prerequisites for Creating a Snapshot of your Database


Determine your storage configuration
Prepare hosts and storage for Database FlashSnap
Create a default snapplan (dbed_vmchecksnap -o setdefaults)
Change the default values?
Yes
No
Modify the snapplan
Validate the snapplan (dbed_vmchecksnap -o validate)
Pass validation?
No
Yes
SNAPSHOT
Fix the error

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. In the event of primary database failure, you can recover it by reverse resynchronizing the snapshot volumes.

The following flow chart depicts the actions you can perform after creating a snapshot of your database using Database FlashSnap.

Actions you can perform after Creating a Snapshot of your Database


SNAPSHOT
Create the snapshot volumes (dbed_vmsnap -o snapshot)
Do you want to clone the database?
No
Yes
Will you use the clone on a secondary host?
Yes
The snapshot disk group will be imported
No
Do you want to resynchronize the snapshot volumes?
Yes
Resynchronize. Reattach snapshot volumes (dbed_vmsnap -o resync)
No
Do you want to: (1) mount the snapshot volumes, (2) clone the database automatically, or (3) clone manually?
(1)
(2)
(3)
Mount the snapshot volumes (dbed_vmclonedb -o mount)
Mount the snapshot volumes and create the clone automatically (dbed_vmclonedb -o recoverdb)
Mount the snapshot volumes and create the clone manually (dbed_vmclonedb -o mountdb)
Update the status after the clone has been recovered manually (dbed_vmclonedb -o update_status)
Begin reverse resychnorization (dbed_vmsnap -o reverse_resync_begin)
Are you done with the snapshot?
Yes
Commit the reverse resynchronization changes?
Shut down the database and unmount the snapshot volumes (dbed_vmclonedb -o umount)
Is the clone on a secondary host?
Yes
Deport the snapshot disk group
No
Yes
No
Are you done with the snapshot?
No
Yes
Commit reverse resynchronization changes (dbed_vmsnap -o reverse_resync_commit)
Abort reverse resynchronization (dbed_vmsnap -o reverse_resync_abort)
Restart the clone database (dbed_vmclonedb -o restartdb)

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 (such as 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.

When using dbed_vmchecksnap to create or validate a snapplan, the following parameters are set:

Parameter	**Value**
SNAPSHOT_VERSION	Specifies the snapshot version for this major release of VERITAS Storage Foundation *for Oracle*.
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	The name of the VxVM disk group used by the primary database.
SNAPSHOT_DG	The name of the disk group containing the snapshot volumes. 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.
ORACLE_SID	The name of the Oracle database. By default, the name of the Oracle database is not included in the snapplan.
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 muliplexing 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**.

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 **yes**.
SNAPSHOT_MODE	**online** or **offline** or **instant** Specifies whether the database snapshot should 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. **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.
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.

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.
SNAPSHOT_MIRROR	Specifies the number of plexes to be snapshot. The default value is 1.

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.

Note You cannot access Database FlashSnap commands (dbed_vmchecksnap, dbed_vmsnap, and dbed_vmclonedb) with the VxDBA menu utility.

Prerequisites

- Storage must be configured as specified in "Preparing Hosts and Storage for Database FlashSnap" on page 204.
- 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 (init.ora).
- The Oracle database must have at least one mandatory archive destination. For further details, see "Establishing a Mandatory Archive Destination" on page 229.
- *ORACLE_HOME* cannot reside on disk which will be used for snapshot.

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.

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]
   ```

3. Open the snapplan file in a text editor and modify it as needed.

For information on creating a snapplan using the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

Example

In this example, a snapplan, `snap1`, is created for a snapshot image in a single-host configuration and default values are set. The host is named `host1` and the working directory is `/export/snap_dir`.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/9i -f snap1 -o setdefaults -t host1
Snapplan snap1 for PROD.
=====================================================
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_PRODdg
ORACLE_SID=PROD
```

```
ARCHIVELOG_DEST=/prod_ar
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnapSNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Example

In the following example, a snapplan, snap2, is created for a snapshot image in a two-host configuration, and default values are set. The primary host is host1, the secondary host is host2, and the working directory is /export/snap_dir.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/9i -f snap2 -o setdefaults  -t host2
Snapplan snap2 for PROD.
=====================================================
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_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.

Example

In the following example, the -p option is used with setdefaults to assign my_tag as the SNAPSHOT_PLEX_TAG value.

```
# dbed_vmchecksnap -S $ORACLE_SID -H $ORACLE_HOME -O setdefaults \
  -p my_tag -f snap1 -t PROD
Snapplan snap1 for PROD
=====================================================
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Creating Multi-mirror Snapshots

To make 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 gets disabled. Snapshot volumes can be mounted and the entire database snapshot is usable even if one of the mirror gets disabled. The multi-mirror snapshots are enabled via SNAPSHOT_MIRROR=<n> in the snapplan.

Note There are no changes to the Command Line usage or arguments for the Flashsnap tools.

Note Before taking the snapshot, make sure all tagged snapshot mirrors are in SNAPDONE state.

The following sample explains the setup and the procedure for taking multi-mirror snapshots:

1. Add the second mirror and DCO log. When allocating storage for the second mirror and DCO logs, make sure the snap volumes are splittable. If snap volumes are not splittable, dbed_vmchecksnap fails with appropriate errors.

   Tag the newly added mirror with the same tag as that of the first mirror.

Assume that the volume has `fastresync = on`, has `dcolog = on`, and already has one `SNAPDONE` mirror and is tagged with `dbed_flashsnap`.

```
# vxsnap -g dg_a  addmir dg_a_vol1 alloc=dg_a03
# vxedit -g dg_a set putil2=dbed_flashsnap dg_a_vol1-03
```

2. Add `SNAPSHOT_MIRROR` keyword to the snapplan. Here is a sample snapplan.

```
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=2
```

Establishing a Mandatory Archive Destination

When cloning a database using Database FlashSnap (see “Cloning a Database (dbed_vmclonedb)” on page 245 for details), the Oracle database must have at least one mandatory archive destination, otherwise `dbed_vmchecksnap` results in this error message:

```
SFORA dbed_vmchecksnap ERROR V-81-5677 Could not find a mandatory,
primary and valid archive destination for database PROD.

Please review the LOG_ARCHIVE_DEST_n parameters and check
v$archive_dest.
```

This example shows how to establish a mandatory archive destination using SQL*Plus:

```
alter system set log_archive_dest_1 =
'LOCATION=/ora_mnt/oracle/oradata/PROD/archivelogs MANDATORY
[REOPEN]’ [scope=both];
```

For more information about Oracle parameters for archiving redo logs, see your Oracle documentation.

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.

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.
- After validating the snapplan, you have the option of modifying the snapplan file to meet your storage configuration requirements.
- When using `dbed_vmchecksnap` 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.
- The `dbed_vmchecksnap` command can be used on the primary or secondary host.
- See the `dbed_vmchecksnap`(1M) manual page for more information.

▼ 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
```

Note In HA environment, you must modify the default snapplan, use the virtual host name defined for the resource group for the PRIMARY_HOST and/or SECONDARY_HOST, and run validation.

For information on validating a snapplan using the GUI, see "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267.

Example

In the following example, a snapplan, snap1, is validated for a snapshot image in a single-host configuration. The primary host is host1 and the working directory is /export/snap_dir.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/9i \
-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 no

The snapplan snap1 has been created.
```

Example

In the following example, a snapplan, snap2, is validated for a snapshot image in a two-host configuration. The primary host is host1, the secondary host is host2, and the working directory is /export/snap_dir.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H \
/oracle/product/9i -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/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 yes

The snapplan snap2 has been created.
```

Displaying, Copying, and Removing a Snapplan (dbed_vmchecksnap)

This section explains how to list all snapplans for a specific Oracle database, display a snapplan file, and copy and remove snapplans.

Usage Notes

- If the local snapplan is updated or modified, you must re-validate it.
- If the database schema or disk group is modified, you must revalidate it after running dbed_update.

▼ 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
```

Example

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
```

Note 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 explanations of the various statuses that may appear for SNAP_STATUS and DB_STATUS, refer to “VERITAS Database FlashSnap Status Information” on page 495

▼ 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
```

Example

In the following example, the snapplan snap1 is displayed.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o list
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=yes
SNAPSHOT_MIRROR=1
STORAGE_INFO
PRODdg02
SNAP_PLEX=prod_ar-02

STATUS_INFO
SNAP_STATUS=init_full
DB_STATUS=init
LOCAL_SNAPPLAN=/export/snap_dir/snap1
```

▼ To copy a snapplan from the VxDBA repository to your current directory

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 VxDBA repository to your current directory, the snapplan must not already be present in the current directory.

Use the dbed_vmchecksnap command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID \
-f SNAPPLAN -o copy
```

Example

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'
```

▼ To remove a snapplan from the VxDBA repository

A snapplan can be removed from a local directory or repository if the snapplan is no longer needed.

Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -f\
SNAPPLAN -o remove
```

Example

In the following example, the snapplan, `snap1`, is removed from the VxDBA 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. dbed_vmsnap ensures that a backup control file is created when the snapshot database is created, which allows for complete data recovery, if needed.

For a complete list of all snapshot and database statuses, see "VERITAS Database FlashSnap Status Information" on page 495.

Note You cannot access Database FlashSnap commands (dbed_vmchecksnap, dbed_vmsnap, and dbed_vmclonedb) with the VxDBA menu utility.

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.

▼ To create a snapshot

1. Change directories to the working directory in which your snapplan is stored:

```
$ cd /working_directory
```

2. If `SNAPSHOT_MODE` is set to `offline` in the snapplan, shut down the database.

3. Create the snapshot image using the `dbed_vmsnap` command:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \
-o snapshot [-F]
```

Note 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 VERITAS Storage Foundation *for Oracle* commands. (That is, the volumes were synchronized without using VERITAS Storage Foundation *for Oracle* 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.

Note 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 VxDBA repository volume for a two-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 VxDBA repository.

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. For details, see "Backing Up the Database from Snapshot Volumes (dbed_vmclonedb)" on page 239. You can also create another Oracle database for decision-support purposes. See "Cloning a Database (dbed_vmclonedb)" on page 245 for more information.

For information on creating a snapshot using the GUI, see "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267.

Example

In this example, a snapshot image of the database, PROD, is created for a single-host configuration. In this case, the SECONDARY_HOST parameter is set the same as the PRIMARY_HOST parameter in the snapplan.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot

dbed_vmsnap started at 2004-04-02 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 2004-04-02 14:16:11
```

Example

In this example, a snapshot image of the primary database, PROD, is created for a two-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 2004-04-09 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:

dbed_vmsnap ended at 2004-04-09 23:02:58
```

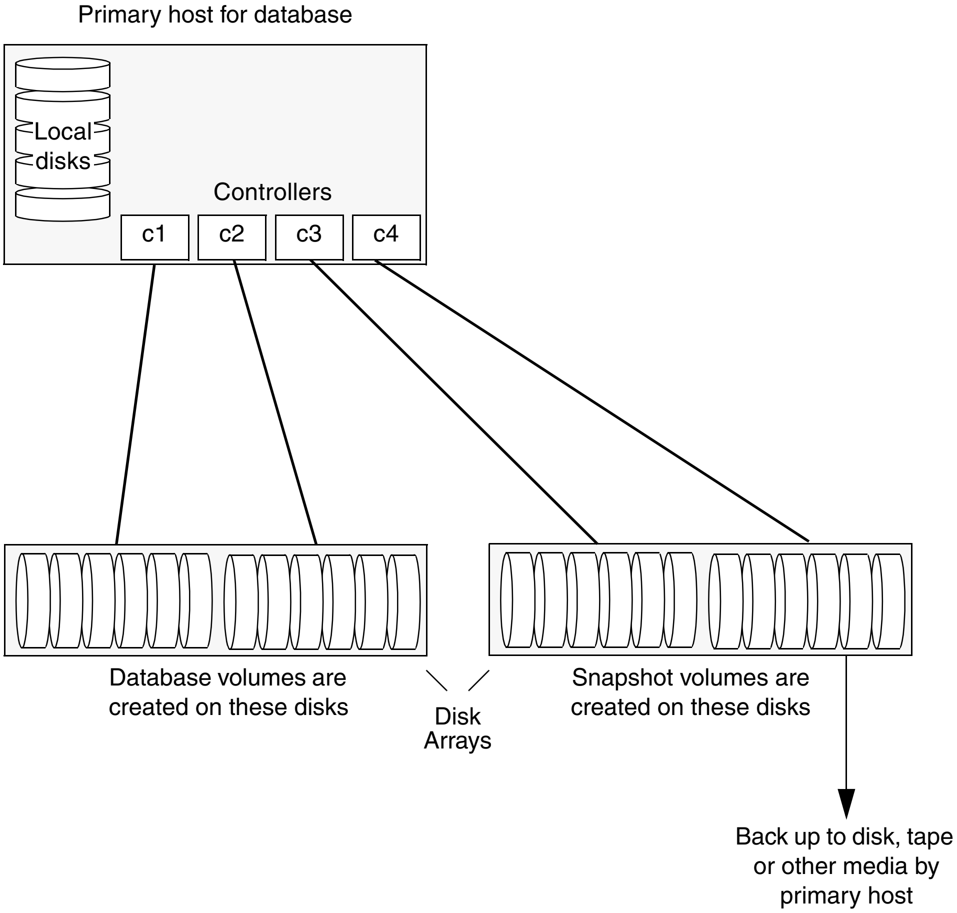
Backing Up the Database from Snapshot Volumes (dbed_vmclonedb)

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.

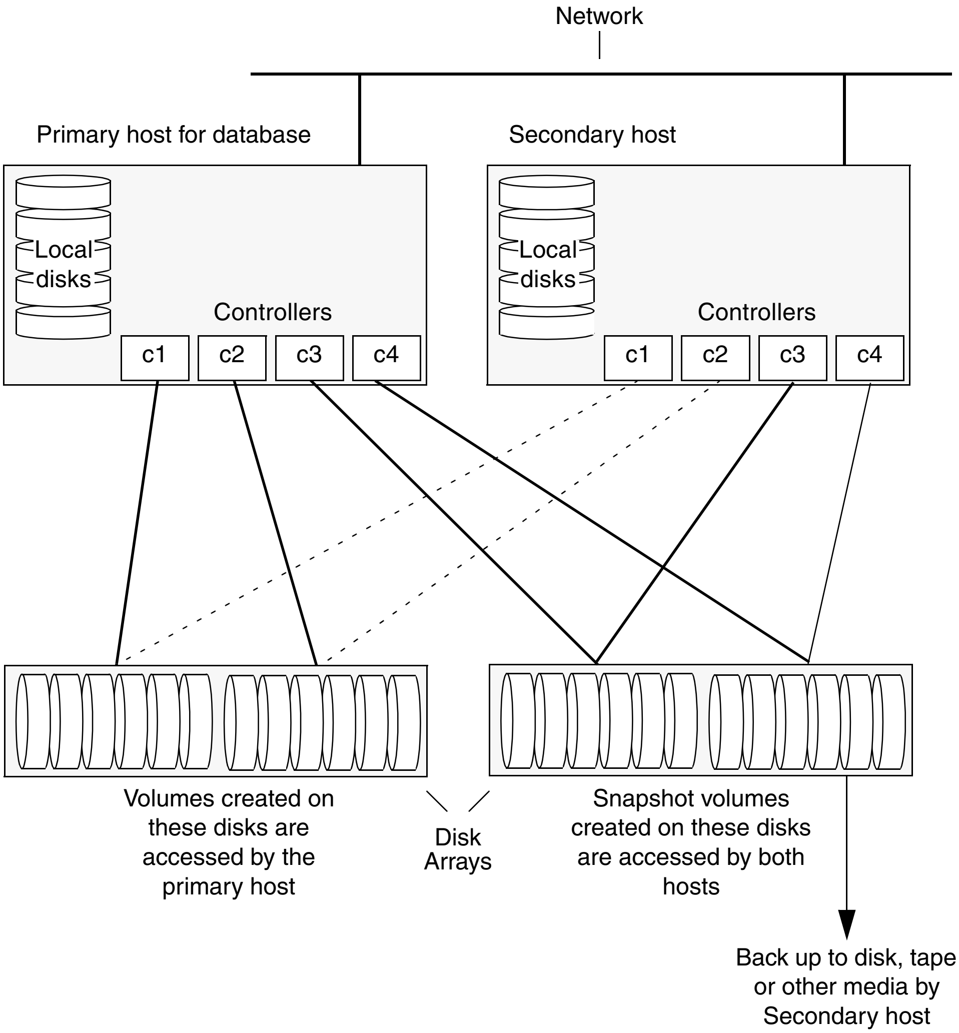
The figure below, "Example System Configuration for Database Backup on the Primary Host," shows a typical configuration when snapshot volumes are located on the primary host.

Example System Configuration for Database Backup on the Primary Host


Primary host for database
Local disks
Controllers
c1
c2
c3
c4
Database volumes are created on these disks
Disk Arrays
Snapshot volumes are created on these disks
Back up to disk, tape or other media by primary host

The following figure, "Example System Configuration for Database Backup on a Secondary Host," shows a typical configuration when snapshot volumes are used on a secondary host.

Example System Configuration for Database Backup on a Secondary Host


Network
Primary host for database
Secondary host
Local disks
Controllers
c1
c2
c3
c4
Local disks
Controllers
c1
c2
c3
c4
Volumes created on these disks are accessed by the primary host
Disk Arrays
Snapshot volumes created on these disks are accessed by both hosts
Back up to disk, tape or other media by 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 complete the steps in "Summary of Database Snapshot Steps" on page 217, "Validating a Snapplan (dbed_vmchecksnap)" on page 230, and "Creating a Snapshot (dbed_vmsnap)" on page 236.
- 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.

Usage Notes

- The dbed_vmclonedb command can be used on the secondary host.
- In a single-host configuration, the primary and secondary hosts are the same.
- In a single-host configuration, -r relocate_path is required.
- In a two-host configuration, the vxdbavol=*vol_name* option is required.
- If SNAPSHOT_MODE is set to offline or instant, a two-host configuration is required and -r *relocate_path* is not allowed.
- See the dbed_vmclonedb(1M) manual page for more information.

Note You cannot access Database FlashSnap commands (dbed_vmchecksnap, dbed_vmsnap, and dbed_vmclonedb) with the VxDBA menu utility.

Mounting the Snapshot Volumes and Backing Up

Before using the snapshot volumes to do a backup, you must first mount them as follows.

▼ To mount the snapshot volumes

Use the dbed_vmclonedb command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID -g snap_dg \
-o mount,new_sid=new_sid -f SNAPPLAN [-H ORACLE_HOME] \
[-r relocate_path]
```

You can now backup an individual file or a group of files under a directory onto the backup media.

Example

In this example, snapshot volumes are mounted.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o mount,new_SID=NEWPROD -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
```

▼ 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:

   ```
   $ /opt/VRTS/bin/dbed_ckptdisplay -S ORACLE_SID -n
   ```

2. You can mount one of the listed Storage Checkpoints using:

   ```
   $ /opt/VRTS/bin/dbed_ckptmount -S ORACLE_SID -c CKPT_NAME \
     -m MOUNT_POINT
   ```

Limitations

- Any mounted Storage Checkpoints must be unmounted before running the following commands:

  ```
  $ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=new_sid \
    -f SNAPPLAN
  ```

- It is only possible to mount a Storage Checkpoint carried over with the snapshot volumes in a two-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 backup the database using the snapshot

Copy the snapshot volumes to tape or other appropriate backup media.

Note If you use the Oracle online backup method, you must also backup all the archived log files in order to do a complete restore and recovery of the database.

Restoring from Backup

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. See "Resynchronizing Your Database to the Snapshot" on page 255 for more information.

Cloning a Database (dbed_vmclonedb)

This section explains how to create a clone database using the 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 239 for more information. You can backup 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 single-host 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 single-host 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 a two-host configuration, the `dbed_vmclonedb` command imports the snapshot disk group `SNAP_dg`, mounts the file systems on the snapshot volumes, 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. When creating the clone off host, `-o vxdbavol=vol_name` is required.

Caution 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 complete the steps in "Summary of Database Snapshot Steps" on page 217, "Validating a Snapplan (dbed_vmchecksnap)" on page 230, and "Creating a Snapshot (dbed_vmsnap)" on page 236.
- 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.
- The Oracle database must have at least one mandatory archive destination. For further details, see "Establishing a Mandatory Archive Destination" on page 229.

Usage Notes

- The `dbed_vmclonedb` command can be used on the secondary host.
- In a single-host configuration, `-r` *`relocate_path`* is required. This command is also needed if the name of the clone database is different than the primary database.
- In a two-host configuration, the `vxdbavol=`*`vol_name`* option is required.
- 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.

▼ 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 -g snap_dg \
-o mountdb,new_sid=new_sid[,vxdbavol=vol_name] -f SNAPPLAN \
[-H ORACLE_HOME] [-r relocate_path]
```

2. Recover the database manually.

3. Update the snapshot status information for the clone database in the VxDBA repository:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,new_sid=new_sid \
-f SNAPPLAN [-r relocate_path]
```

Example

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.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o mountdb,new_sid=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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.
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 2004-04-02 15:34:59
```

The database is recovered manually using `dbinitdb`.

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 \
-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
```

Example

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.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S -g SNAP_PRODdg \
-o mountdb,new_sid=NEWPROD,vxdbavol=SNAP_arch -f snap2
dbed_vmclonedb started at 2004-04-09 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.
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 2004-04-09 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 \
-f snap2
dbed_vmclonedb started at 2004-04-09 23:34:01
The snapshot status has been updated.
dbed_vmclonedb ended at 2004-04-09 23:34:35
```

▼ To clone the database automatically

Use the dbed_vmclonedb command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID -g snap_dg \
-o recoverdb,new_sid=new_sid[,vxdbavol=vol_name] -f SNAPPLAN \
[-H ORACLE_HOME] [-r relocate_path]
```

Where:

- *ORACLE_SID* is the name of the Oracle database used to create the snapshot.
- *snap_dg* is the name of the diskgroup that contains all the snapshot volumes.
- new_sid specifies the *ORACLE_SID* for the clone database.
- vxdbavol is the volume that contains the snapplan data. This name is provided after you run dbed_vmsnap -o snapshot.
- *SNAPPLAN* is the name of the snapplan file.
- *ORACLE_HOME* is the *ORACLE_HOME* setting for the *ORACLE_SID* database.
- *relocate_path* is the name of the initial mount point for the snapshot image.

Note 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.

For information on cloning a database using the GUI, see "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267.

Example

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 -g SNAP_PRODdg \
-o recoverdb,new_sid=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2004-04-02 14:43:05
```

Example

In the following example, a clone of the primary database is automatically created on a secondary host.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o recoverdb,new_sid=NEWPROD,vxdbavol=SNAP_arch -f snap2
dbed_vmclonedb started at 2004-04-09 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.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2004-04-09 23:04:50
```

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=new_sid \
-f SNAPPLAN [-r relocate_path]
```

Example

In this example, the clone database is shut down and file systems are unmounted for a clone on the same host as the primary database (a single-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD \
-f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 15:11:22
NOTICE: Umounting /clone/prod_db.
NOTICE: Umounting /clone/prod_ar.
dbed_vmclonedb ended at 2004-04-02 15:11:47
```

Example

In this example, the clone database is shut down, file systems are unmounted, and the snapshot disk group is deported for a clone on a secondary host (a two-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD \
-f snap2
dbed_vmclonedb started at 2004-04-09 23:09:21
NOTICE: Umounting /clone/arch.
NOTICE: Umounting /clone/prod_db.
dbed_vmclonedb ended at 2004-04-09 23:09:50
```

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 -g snap_dg \
-o restartdb,new_sid=new_sid -f SNAPPLAN [-H ORACLE_HOME] \
[-r relocate_path]
```

For information on restarting the clone database using the GUI, see "Using the VERITAS Storage Foundation for Oracle Graphical User Interface" on page 267.

Example

In this example, the clone database is re-started on the same host as the primary database (a single-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o restartdb,new_sid=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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 2004-04-02 15:15:19
```

Example

In this example, the clone database is re-started on the secondary host (a two-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o restartdb,new_sid=NEWPROD,vxdbavol=SNAP_arch -f snap2
dbed_vmclonedb started at 2003-04-09 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 2003-04-09 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, `dbed_vmclonedb` will display the `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 $ORACLE_SID is at
/tmp/add_tf.$ORACLE_SID.sql.
```

Note *$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.

Note 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.

There are two choices when resynchronizing the data in a volume:

- Resynchronizing the snapshot from the original volume. This option 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.

Prerequisites

- You must be logged in as the Oracle database administrator.
- Before you can resynchronize the snapshot image, you must complete the steps in "Summary of Database Snapshot Steps" on page 217, "Validating a Snapplan (dbed_vmchecksnap)" on page 230, and "Creating a Snapshot (dbed_vmsnap)" on page 236.
- 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 249.

Usage Notes

- The `dbed_vmsnap` command can only be executed on the primary host.
- In a two-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.
- You cannot access the Database FlashSnap commands (`dbed_vmchecksnap`, `dbed_vmsnap`, and `dbed_vmclonedb`) with the VxDBA menu utility.

▼ To resynchronize the snapshot image

Use the dbed_vmsnap command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN -o resync
```

For information on resynchronizing the snapshot using the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

Example

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 2004-04-02 16:19:05
The option resync has been completed.
dbed_vmsnap ended at 2004-04-02 16:19:26
```

Now, you can again start creating snapshots.

Resynchronizing Your Database to the Snapshot

If your database becomes corrupted, you can use reverse resynchronization to recover the database from a clone. The reverse resynchronization feature of VERITAS Database FlashSnap enables you to resynchronize the primary database or volume with a clone database or snapshot volume.

Reverse resynchronization requires the primary database to be offline so that it remains unchanged.

Caution 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. The `dbed_vmsnap -o reverse_resync_commit` command cannot be undone and should be used with extreme caution.

Prerequisites

- You must be logged in as the Oracle database administrator.
- Before you can reverse resynchronize the snapshot image, you must complete the steps in "Summary of Database Snapshot Steps" on page 217 and "Creating a Snapshot (dbed_vmsnap)" on page 236.
- The mount point for the primary database must be created by and owned by the Oracle DBA user before mounting the VxFS file system.
- If a clone database has been created, you must shut it down and unmount the file systems using the `dbed_vmclonedb -o umount` command before you can reverse resynchronize the snapshot image. 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 249.
- The primary database must be offline.

Usage Notes

- The `dbed_vmsnap` command can only be executed on the primary host.
- If the Oracle authentication password is used, you need to recreate it using the `ORAPWD` utility after executing `dbed_vmsnap -o reverse_resync_commit`.

Note You cannot access Database FlashSnap commands (`dbed_vmchecksnap`, `dbed_vmsnap`, and `dbed_vmclonedb`) from the VxDBA menu utility.

▼ To begin reverse resynchronization

The `-o reverse_resync_begin` option of the `dbed_vmsnap` command imports the disk group that was deported from the secondary host (in a two-host configuration) and joins it back to the original disk group. The command unmounts the original volumes, mounts the snapshot volumes with the file systems that are configured for the primary database, and brings up the database snapshot image as the primary database. This operation requires the primary database to be offline so that its contents remain unchanged.

Note 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 begin reverse resynchronization, use the `-o reverse_resync_begin` option as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \
-o reverse_resync_begin
```

Note Any mounted storage checkpoints must be unmounted before running `dbed_vmsnap -o reverse_resync`.

After executing `reverse_resync_commit`, checkpoints created on the original database or clone database will be deleted.

Limitation

Reverse resynchronization requires the primary database to be down. However, in an HA environment, when VCS detects that the primary database is down, it starts the fail over process, VxDBA repository gets unmounted, `dbed_vmsnap` dies.

Use the following workaround:

1. To temporarily freeze the VCS Resource Group for the database, enter:

   ```
   # hagrp -freeze ResourceGroup
   ```

2. Shutdown the primary database.

3. Run `reverse_resync`.

4. When `reverse_resync` is completed, start up the database. Make sure it is in `archivelog` mode.

5. To unfreeze the Resource Group, enter:

```
# hagrp -unfreeze ResourceGroup
```

▼ To abort reverse resynchronization

The `-o reverse_resync_abort` option aborts `-o reverse_resync_begin`, unmounts the snapshot volumes, and mounts the original volumes back with the file systems that are configured to use the volume. This operation is only allowed after `-o reverse_resync_begin` has been executed and cannot be used after reverse resynchronization has been committed (`-o reverse_resync_commit`).

To abort reverse resynchronization, use the `-o reverse_resync_begin` option as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \
-o reverse_resync_abort
```

▼ To commit reverse resynchronization changes

The `-o reverse_resync_commit` option commits the reverse resynchronization changes after you have verified that they are acceptable. The operation resynchronizes the original volume from the data in the snapshot.

To commit the reverse resynchronization changes, use the `-o reverse_resync_commit` option as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \
-o reverse_resync_commit
```

Then, restart the primary database in `ARCHIVELOG` mode so that it is ready to take another snapshot.

If the Oracle authentication password file is used for the database, it needs to be recreated using the `ORAPWD` utility after the `reverse_resync_commit` operation is performed.

For information on resynchronizing the snapshot using the GUI, see “Using the VERITAS Storage Foundation for Oracle Graphical User Interface” on page 267.

Example

Note The following example is valid only for the `online_snapshot` mode.

Reverse resynchronization is started on the primary host.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 \
-o reverse_resync_begin
dbed_vmsnap started at 2004-04-02 15:53:45
Database PROD (SID=PROD) is running.
dbed_vmsnap ended at 2004-04-02 15:54:29
```

Reverse resychronization is aborted on the primary host.

```
$ /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
```

Reverse resychronization changes are committed on the primary host.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 \
-o reverse_resync_commit
dbed_vmsnap started at 2004-04-02 16:18:44
The option reverse_resync_commit has been completed.
dbed_vmsnap ended at 2004-04-02 16:18:56
```

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 root.
- 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:

a. 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:

b. 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 -D db -f snapplan -o remove
```

Example

To 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
```

Using Database FlashSnap in an HA Environment

VERITAS Storage Foundation 4.1 *for Oracle* supports FlashSnap in the HA environment.

When using FlashSnap in the HA environment, observe the following limitations:

- Modify the default snapplan to use the virtual host name defined for the database resource group for the `PRIMARY_HOST` and/or the `SECONDARY_HOST` parameters and validate the snapplan before creating a snapshot by running the following command:

  ```
  dbed_vmchecksnap -S ORACLE_SID -H ORACLE_HOME \
  -f SNAPPLAN -o validate
  ```

- The primary database must be down before you perform reverse resychronization (`dbed_vmsnap -S ORACLE_SID -f SNAPPLAN -o reverse_resync_begin`). When VERITAS Cluster Server (VCS) detects that the primary database is down, it starts the failover process and the VxDBA repository is unmounted and the `dbed_vmsnap` command is aborted.

To avoid the VCS failover process, perform the following steps:

1. As root, temporarily freeze the VCS resource group for the database:

   ```
   # hagrp -freeze ResourceGroup
   ```

2. Shut down the primary database.

3. Run reverse resynchronization:

   ```
   # dbed_vmsnap -S ORACLE_SID -f SNAPPLAN -o\
    reverse_resync_begin
   ```

4. After reverse reynchronization changes are committed (`-o reverse_resync_commit`), verify that the database has been started in `ARCHIVELOG` mode. This information is provided in the status messages that appear after running committing reverse resynchronization changes.

5. Unfreeze the resource group:

   ```
   # hagrp -unfreeze ResourceGroup
   ```

Using VERITAS NetBackup for Database Backup 11

VERITAS NetBackup™ is a separately licensed product and is not included with VERITAS Storage Foundation *for Oracle*. The information included here is for reference only.

For information about using third-party software for backing up files, see "Using Third-Party Software to Back Up Files" on page 489.

Topics include:

Using VERITAS NetBackup for Backup and Restore

VERITAS NetBackup provides for high performance, online (hot) backups of databases that must be available on a 24x7 basis, as well as offline (cold) database backups. VERITAS NetBackup lets you back up and restore database files and directories. You can set up schedules for automatic, unattended, online, and offline database backup, as well as full or incremental backup. These backups are managed entirely by the NetBackup server. You can also manually back up database files from any of the NetBackup clients. Client users can perform database backups and restores from their client systems on demand.

Using VERITAS NetBackup to Backup and Restore Quick I/O Files

The information in this section assumes that you are *not* using NetBackup *for Oracle* to back up and restore Quick I/O files. If you *are* using NetBackup *for Oracle*, instead refer to "Using Oracle RMAN to Back Up and Restore Quick I/O Files" on page 490.

VERITAS NetBackup does not follow symbolic links when backing up files. Typical backup management applications are designed this way to avoid backing up the same data twice. This would happen if both the link and the file it points to were included in the list of files to be backed up.

A Quick I/O file consists of two components: a hidden file with the space allocated for it, and a link that points to the Quick I/O interface of the hidden file. Because NetBackup does not follow symbolic links, you must specify both the Quick I/O link and its hidden file in the list of files to be backed up.

Example

To view all files and their attributes in the db01 directory:

```
$ ls -la /db01
total 2192
drwxr-xr-x 2 root    root          96   Oct 20 17:39 .
drwxr-xr-x 9 root    root        8192   Oct 20 17:39 ..
-rw-r--r-- 1 oracle  dba      1048576   Oct 20 17:39 .dbfile
lrwxrwxrwx 1 oracle  dba           22   Oct 20 17:39 dbfile ->\
                                        .dbfile::cdev:vxfs:
```

In the example above, you must include both the symbolic link dbfile and the hidden file .dbfile in the file list of the backup class.

If you want to back up all Quick I/O files in a directory, you can simplify the process by just specifying the directory to be backed up. In this case, both components of each Quick I/O file will be properly backed up. In general, you should specify directories to be backed up unless you only want to back up some, but not all files, in those directories.

Because VERITAS NetBackup is integrated with VERITAS Storage Foundation *for Oracle*, VERITAS NetBackup backs up extent attributes of a Quick I/O file and restores them accordingly. Quick I/O files can then be backed up and restored as regular files using VERITAS NetBackup, while preserving the Quick I/O file's extent reservation. Without this feature, restoring the file could cause the loss of contiguous reservation, which can degrade performance.

When restoring a Quick I/O file, if both the symbolic link and the hidden file already exist, VERITAS NetBackup will restore both components from the backup image. If either one or both of the two components are missing, VERITAS NetBackup creates or overwrites as needed.

Note The Oracle backup and restore utility cannot be used to back up and restore Quick I/O files.

Note Some back up software may not be able to back up and restore VxFS extent attributes. See the `qio_recreate`(1M) online manual page for information on how to restore missing Quick I/O files.

Using VERITAS NetBackup to Back Up and Restore Oracle Disk Manager Files

Oracle allocates Oracle Disk Manager files with contiguous extent layouts for good database performance. When you back up your database using VERITAS NetBackup, extent attributes are backed up automatically. 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 not using NetBackup or 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. Refer to the `odmmkfile`(1M) manual page for more information.

Example

To preallocate an Oracle datafile with size 100 M, assuming the Oracle database block size is 8K use the `odmmkfile` command and enter:

```
$ /opt/VRTS/bin/odmmakefile -h 8k -s 100m filename
```

Using the VERITAS Storage Foundation for Oracle Graphical User Interface 12

You can access VERITAS Storage Foundation *for Oracle*, VERITAS Volume Manager, and VERITAS File System functions through the VERITAS Storage Foundation *for Oracle* graphical user interface (GUI). This chapter describes only how to use the GUI to perform various storage management tasks for your Oracle database.

Topics covered in this chapter include:

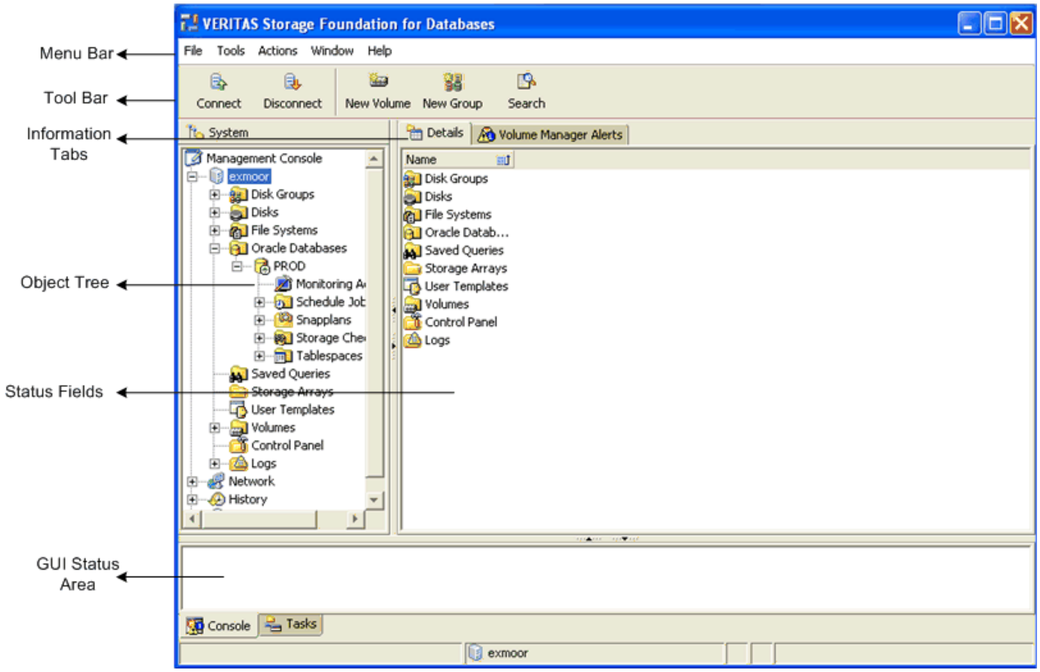
VERITAS Storage Foundation for Oracle GUI

The graphical user interface (GUI) allows you to perform storage management duties for Oracle, such as monitoring the database, using Storage Checkpoints and Storage Rollback, and file system space planning. This chapter describes the components of the GUI.

Note The GUI uses VxDBA to perform many actions. For information on setting up VxDBA and the GUI in a high availability environment, see "Setting Up VxDBA in an HA Environment" on page 405.

The GUI runs in a client-server environment. The server is located on a host that runs VERITAS Storage Foundation *for Oracle*. The client can run on any Solaris, Windows NT, Windows 98, Windows 2000, Windows Me, and Windows XP machine that supports the Java Runtime Environment.

Within the GUI, you can perform tasks from the main menu bar or you can right click an object on the navigational (left) side of the screen, as seen in the following graphic:



There are multiple sections within the GUI. The following list describes the sections shown in the graphic, which depicts the Main Window:

- **Menu Bar** - allows you to perform various VxDBA operations. The options in the Menu Bar will vary according to the object in the object tree that you have selected.

Note To access online help from the Menu Bar, click **Help** > **Contents**.

- **Tool Bar** - provides shortcuts to various operations available in the Menu Bar. The Tool Bar is icon-based and dynamically changes when you select something from the Object Tree. When you use your mouse to point at an icon, a description of the icon appears.
- **Information Tabs** - allow you to view different information about the same object in the Object Tree. For example, if you are viewing details about a database, you can click a different tab to view different information about that database.
- **Object Tree** - is a dynamic hierarchical display of VERITAS Storage Foundation *for Oracle*, VERITAS Volume Manager, and VERITAS File System objects and other objects on the system.
- **Status Fields** - indicate the status of the object you are viewing. To change your view, click one of the Information Tabs at the top of the window.
- **GUI Status Area** - displays GUI status, which is provided through VERITAS Enterprise Administrator. See the VERITAS Enterprise Administrator documentation for more information.

Within the GUI, options that are not available are displayed as grayed-out and are not clickable. Also, you can point to an object on the screen and a description of the object is displayed in a pop-up field.

The following table describes terms associated with the use of the mouse:

Term	Definition
Click	Press and release the mouse button.
Double-click	Click the mouse button twice (quickly).
Right-click	Press and release the right mouse button.
Press and Hold	Press and continue to hold down the mouse button.
Point	Move the tip of the pointer onto an item on the screen.

Term	Definition
Select	Click the mouse button while the pointer is directly over the item to be selected.
Drag	Slide the mouse while pressing a mouse button.

Overview of GUI Functions

The GUI allows you to perform many storage management tasks for Oracle. This section offers an overview of those tasks. The tasks that you can perform dynamically from the main menu bar and pop-up menu depend on what is highlighted on the object tree. For example, if you have the Oracle database highlighted in the tree, you can open a database instance from the Oracle menu.

Note Some operations require that you must be logged in as `root`. Please read all "Prerequisites" and "Usage Notes" before starting a procedure.

GUI functionality includes:

- Database operations

  You can start, stop, or duplicate a database instance through the GUI.

- Storage Checkpoint Management

  You can create and roll back to Storage Checkpoints. You can also mount, unmount, and remove Storage Checkpoints. Storage Checkpoints can be used to clone the database.

- Database FlashSnap

  Database FlashSnap functionality is available through the GUI. With Database FlashSnap, you can create online or offline snapshots of your database, which can be used as backups. You can also use these snapshots to recover your database if it becomes corrupt.

- System configuration maintenance

  You can convert datafiles to or from Quick I/O files and you can view topology and statistic information. You can also check and save your system configuration.

- Monitoring Agent administration

  You can use the GUI to set up, configure, start, and stop a monitoring agent, which can be used to send alarm notifications and automatically grow file systems when space usage exceeds user-defined thresholds.

Prerequisites

- Before running the GUI, you must have appropriate permission to access the GUI. You must make permission changes to allow database administrators to access these tools. The default settings for the `/opt/VRTSdbed` directory at installation time allows only superuser (`root`) access to the directory. To make the appropriate permission changes, you will need to use the `vxdbedusr` utility. See the steps for granting administrative permissions in the section “Starting VxDBA” on page 338 for more information.

VERITAS Enterprise Administrator Service

To use the VERITAS Storage Foundation *for Oracle* GUI, VERITAS Enterprise Administrator (VEA) Service must be running on the server. The VEA Service is started when you install the software. Occasionally, you may need to manually start or stop the service or add users to the VEA Service console registry.

Prerequisites

- Use the `/opt/VRTS/bin/vxdbedusr` utility to create login names for anyone who needs to run the GUI. To run `/opt/VRTS/bin/vxdbedusr`, you must have superuser (`root`) privileges. For more information, see "Adding Users to the VERITAS Enterprise Administrator Service Console Registry" on page 273.
- You must have superuser (`root`) privileges to execute the `vxsvc` command.

Note You must update the database using the `dbed_update` command at least once on each database before you can manage the database(s) through the GUI.

Adding Users to the VERITAS Enterprise Administrator Service Console Registry

You may want to add users to the VEA Service console registry to allow access to the interface to users other than `root`. You also have the option to give database administrators `root` privileges.

Having `root` privileges means that you can access the volume, disk, and file system objects in the system.

▼ To add users other than root to the VERITAS Enterprise Administrator Service console registry

1. Make sure that the optional GUI package was installed.

```
# pkginfo -l VRTSorgui

 PKGINST:  VRTSorgui
    NAME:  VERITAS Storage Foundation Graphical User Interface for
            Oracle
CATEGORY:  application
    ARCH:  Sparc
 VERSION:  4.1
  VENDOR:  VERITAS Software
    DESC:  VERITAS Storage Foundation Graphical User Interface for
            Oracle
```

```
   PSTAMP:  test.020322155315
 INSTDATE:  Dec 03 2004 15:55
  HOTLINE:  1-800-342-0652
    EMAIL:  support@veritas.com
   STATUS:  completely installed
    FILES:     29 installed pathnames
                8 shared pathnames
               13 directories
                4 executables
            13514 blocks used (approx)
PKGINST: VRTSorgui
```

2. To give `root` privileges within the GUI to the database administrator, use the `vxdbedusr` command as follows.

   ```
   # /opt/VRTS/bin/vxdbedusr -a user [-A] [-f] -n user_name
   ```

   where:

   `-a user` adds a user to the registry.

   `-A` grants the user `root` access.

   `-f` allows the user to be a user other than the `/opt/VRTSdbed` owner.

   `-n` indicates the name of the user that will be added or removed.

   For example, to add a database administrator with the name “oracle” as a user with `root` privileges, enter the following:

   ```
   # /opt/VRTS/bin/vxdbedusr -a user -A -f -n oracle
   ```

3. To add a user without `root` privileges, use the `vxdbedusr` command as follows.

   ```
   # /opt/VRTS/bin/vxdbedusr -a user -n user_name
   ```

   where `-a` adds a user to the registry.

   For example, to add “oracle” as a user, enter the following:

   ```
   # /opt/VRTS/bin/vxdbedusr -a user -n oracle
   ```

4. To add a group to the console registry, use the `vxdbedusr` command as follows.

   ```
   # /opt/VRTS/bin/vxdbedusr -a group -n group_name
   ```

   where `-a` adds the user group to the registry.

   For example, to add “dba” as a group, enter the following:

   ```
   # /opt/VRTS/bin/vxdbedusr -a group -n dba
   ```

Note Make sure you shutdown and then restart the VEA server to update the registration information.

Removing Users from the VERITAS Enterprise Administrator Service Console Registry

You may need to restrict access to the VEA Service console registry. You can remove users or user groups from the registry if they have been previously added.

Note You cannot remove `root` from the VEA Service console registry.

▼ To remove users other than root from the VERITAS Enterprise Administrator Service console registry

1. Make sure that the optional GUI package was installed.

```
   # pkginfo -l VRTSorgui
 PKGINST: VRTSorgui
    NAME:  VERITAS Storage Foundation Graphical User Interface for
            Oracle
CATEGORY:  application
    ARCH:  Sparc
 VERSION:  4.1
  VENDOR:  VERITAS Software
    DESC:  VERITAS Storage Foundation Graphical User Interface for
            Oracle
  PSTAMP:  test.020322155315
INSTDATE:  Dec 03 2004 15:55
 HOTLINE:  1-800-342-0652
   EMAIL:  support@veritas.com
  STATUS:  completely installed
   FILES:     29 installed pathnames
               8 shared pathnames
              13 directories
               4 executables
           13514 blocks used (approx)
 PKGINST: VRTSorgui
```

2. To remove a user, use the `vxdbedusr` command as follows.

```
# /opt/VRTS/bin/vxdbedusr -r user -n user_name
```

where `-r` removes the user from the registry.

For example, to remove the user "oracle," enter the following:

```
# /opt/VRTS/bin/vxdbedusr -r user -n oracle
```

3. To remove a group, use the `vxdbedusr` command as follows.

```
# /opt/VRTS/bin/vxdbedusr -r group -n group_name
```

where `-r` removes the user group from the registry.

For example, to remove the group "dba," enter the following:

```
# /opt/VRTS/bin/vxdbedusr -r group -n dba
```

Note Make sure you restart the VEA server to update the registration information.

Starting the VERITAS Enterprise Administrator Service

▼ **To manually start VERITAS Enterprise Administrator Service**

1. Verify the status of the VEA Service:

```
# /opt/VRTS/bin/vxsvcctrl status
Current state of server : NOT RUNNING
```

2. Start the VEA Service:

```
# /opt/VRTS/bin/vxsvcctrl start
Initializing Storage Foundation Provider 4.1 for Oracle
```

3. Again verify the status of the VEA Service:

```
# /opt/VRTS/bin/vxsvcctrl status
Current state of server : RUNNING
```

Shutting Down the VERITAS Enterprise Administrator Service

Although the VEA Service should remain running, you may need to shut it down manually.

▼ To manually shut down VERITAS Enterprise Administrator Service

1. Verify the status of the VEA Service:

   ```
   # /opt/VRTS/bin/vxsvcctrl status
   Current state of server : RUNNING
   ```

2. Stop the VEA Service:

   ```
   # /opt/VRTS/bin/vxsvcctrl stop
   Successfully unloaded the Storage Foundation Provider 4.1
   for Oracle
   ```

3. Again verify the status of the VEA Service:

   ```
   # /opt/VRTS/bin/vxsvcctrl status
   Current state of server : NOT RUNNING
   ```

Opening and Closing the VERITAS Storage Foundation for Oracle GUI

You can run the GUI from a Windows or UNIX client machine. You must have the client software installed before you can use the GUI.

Prerequisites

- Use the `/opt/VRTS/bin/vxdbedusr` utility to create login names for anyone (other than `root`) who needs to run the GUI. To run `/opt/VRTS/bin/vxdbedusr`, you must have superuser (`root`) privileges.
- The `dbed_update` command must be run at least once before you can manage a database through the GUI.

Opening the VERITAS Storage Foundation for Oracle GUI from a Windows Client

▼ To start the VERITAS Storage Foundation for Oracle GUI from a Windows client

1. Click on **Start**, then select **Programs** > **VERITAS** > **VERITAS Enterprise Administrator**.

2. In the Connection pop-up window, enter the host name for the server to which you are connecting and press the **Tab** key.

3. Enter your login name and password. Then click **OK**.

Note The VEA Service must be running on the server. If you need to start the VEA Service, see "Starting the VERITAS Enterprise Administrator Service" on page 276.

Opening the VERITAS Storage Foundation for Oracle GUI from a UNIX Client

▼ To start the VERITAS Storage Foundation for Oracle GUI from a UNIX client

1. From an open terminal window, type **`/opt/VRTSob/bin/vea`** and press Enter.

Note You can run this command only from `root`.

2. In the Connection pop-up window, enter the host name for the server to which you are connecting and press the **Tab** key.

3. Enter your login name and password. Then click **OK**.

Note VEA Service must be running on the server. If you need to start VEA Service, see "Starting the VERITAS Enterprise Administrator Service" on page 276.

Closing the VERITAS Storage Foundation for Oracle GUI

▼ To close the VERITAS Storage Foundation for Oracle GUI

1. From the menu bar, select **File** > **Exit**.

2. The GUI displays a message indicating that you will be disconnected from the host if you continue. Click **Yes** to continue or, click **No** to keep the GUI running.

Note Stopping the GUI is the same for a Windows or UNIX client.

Starting an Oracle Database Instance

You can start an Oracle database instance or a cloned Oracle database instance from the GUI. You must know the UNIX user name and password of the database owner.

The steps to restart an Oracle database instance are the same.

▼ To start an Oracle database instance or a cloned Oracle database instance

1. Click the **Oracle databases** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to start the database.

   ◆ From the menu bar, select **Oracle** > **Start Up Database**.

   or

   ◆ Right click the **Oracle databases** icon to bring up a pop-up menu. Then click **Start Up Database**.

   The Start Up Database wizard screen is then displayed.

3. Enter the Oracle SID, Oracle home, and Owner information. Click **Next** to continue.

4. Enter your password and optionally enter Oracle pfile information, then click **Start Database**. The UNIX user name field is populated automatically and the other fields on this screen are read-only.

5. At the confirmation prompt, click **Yes** to confirm that you want to start the database.

   If the Oracle database was successfully started, you will receive a confirmation message. Click **OK** to continue.

Starting an Oracle Snapshot Database Instance

If you created a snapshot using the Database FlashSnap feature, you can start an Oracle snapshot database instance from the secondary host via the GUI. This is the equivalent of cloning the database from the primary host. You must know the UNIX user name and password of the database owner and there must be an existing snapshot.

For more information on the Database FlashSnap feature within the GUI, see "Managing Snapshots with Database FlashSnap" on page 310.

Prerequisites

- Make sure you have enough space to create a clone database on your system.
- You must have an existing snapshot.

▼ To start an Oracle snapshot database instance

1. Click the **Oracle Databases** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to start the snapshot database.

   - From the menu bar, select **Oracle** > **Start Up Snapshot Database**.

     or

   - Right click the **Oracle Databases** icon to bring up a pop-up menu. Then click **Start Up Snapshot Database**.

   The Start Up Snapshot Database wizard screen is then displayed.

3. In the "connect as" section, enter the local Oracle home information. Then, enter your UNIX user name and password.

4. For the primary database information section, enter the Oracle SID information.

5. Select whether you are creating the database or restarting the database by clicking the appropriate radio button.

   Select "Startup database" if you are starting a new snapshot database based on a snapshot. Select "Restart database" if you are restarting a snapshot database that has already been created.

6. For the snapshot database information section, enter the new Oracle SID, Snapplan file, VxDBA volume name, Snapshot disk group name, and Relocate path in the appropriate fields.

Note To start an on-host snapshot database, the Relocate path is required. To start an off-host snapshot database, the VxDBA volume name field is required. You can obtain this information by viewing the details after you create a snapshot. If you did not retain the information from the snapshot creation, you can use the `dbed_vmchecksnap` command or use View Log via the GUI to retrieve the information. For more information, see "Creating and Working with Snapplans Using dbed_vmchecksnap" on page 459.

7. Click **Start Database**.

8. At the confirmation prompt, click **Yes** to confirm that you want to start the database.

   If the snapshot database was successfully started, you will receive a confirmation message. Click on **OK** to continue.

Shutting Down an Oracle Database Instance

The GUI lets you shut down an Oracle database instance. For example, you must shut down the database to perform a Storage Rollback of an entire database.

▼ To shut down an Oracle database instance

1. Click the actual Oracle database in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to stop the database.

   ◆ From the menu bar, select **Oracle** > **Shutdown Database**.

   or

   ◆ Right click the database to bring up a pop-up menu. Then click **Shutdown Database**.

3. Verify the database information, such as the Oracle SID and Oracle Home. Then, click **Next** to continue.

4. Verify your UNIX user name, then enter your password in the **Password** field.

5. In the dialog box, select the type of shutdown you want to perform:

   ◆ **Normal**

   Use this option to shut down the Oracle database instance in normal situations. If you shut down using this option, no new database connections are allowed. Oracle waits for all currently connected users to disconnect from the database, and then closes and dismounts the database before shutting down the instance. The next database start up does not require an instance recovery.

   ◆ **Transactional**

   Use this option to shut down the Oracle database instance immediately upon completion of all transactions in progress. If you shut down using this option, no client can start a new transaction on this instance, and a client is disconnected when the transaction in progress ends. The next database start up does not require an instance recovery.

- **Immediate**

  Use this option to shut down the Oracle database instance immediately. Use this option in situations where the database, or some application, is running irregularly or a power shutdown is about to occur. If you shut down using this option, all current client SQL statements are terminated immediately, any uncommitted transactions are rolled back, and all connected users are disconnected. Oracle closes and dismounts the database before shutting down the instance. The next database start up does not require an instance recovery.

- **Abort**

  Use this option to shut down the Oracle database instance instantaneously by aborting the database's instance. Use this option with extreme caution and only when normal or immediate shutdown does not work, you experience problems when starting the instance, or you need to shut down the instance instantaneously. If you shut down using this option, all connected users are disconnected, current client SQL statements are terminated immediately, uncommitted transactions are not rolled back, and the instance is terminated without closing the files. The next database start up requires an instance recovery.

Note In this release, the umount option to shutdown or umount the FlashSnap clone database has been removed. Use the **Remove Clone Database** option to shutdown or umount the FlashSnap clone database. See "Removing a Clone Database" on page 291 for details.

6. Click **Shutdown instance** at the bottom of the screen.

7. At the confirmation prompt, click **Yes** to confirm that you want to shut down the database.

   If the database instance was successfully shut down, you will receive a confirmation message. Click **OK** to continue.

Creating Clone Database

The Create Clone Database wizard guides you through the process of creating a clone of the database using either Storage Checkpoints or Volume Manager Flashsnap. Cloning the database helps you perform operations without affecting your production database.

Prerequisites

- The Oracle database must have at least one mandatory archive destination. For further details, see "Establishing a Mandatory Archive Destination" on page 229.

▼ To create a clone database

1. Click the database in the object tree.

2. Select one of the following methods to create a clone database.

   - From the menu bar, click **Oracle > Create Clone Database**.

     or

   - Right click the database object to bring up a pop-up menu. Then, click **Create Clone Database**.

   The Create Clone Database wizard is then displayed. Click **Next**.

3. A read-only dialog with information about the current database instance is displayed. Click **Next** to continue.

4. You are required to select whether you want to create a clone database using a Storage Checkpoint (see "Using Storage Checkpoint" on page 285) or Volume Manager Flashsnap (see "Using Volume Manager FlashSnap" on page 286). After making your selection, click **Next**.

Using Storage Checkpoint

▼ To create a clone database using a Storage Checkpoint

1. In the **Clone Database Type** dialog, select the option to clone the database using a Storage Checkpoint. Click **Next**.

2. In the **New Database Information** dialog, enter the New Oracle SID and the Mount Point.

3. Select whether you would want to clone the database using a new Storage Checkpoint or use an existing Storage Checkpoint.

   If you choose to clone the database using an existing Storage Checkpoint, you must select a Storage Checkpoint from the drop-down list.

   Click **Create Clone Database**.

4. When restarting a Storage Checkpoint clone database that is already created, make sure you check the **Restart Database** option or you would be starting a new Storage Checkpoint clone database.

   Note For restarting a Storage Checkpoint clone database, you must select an existing Storage Checkpoint. You are not required to enter the mount point.

   If the database was successfully cloned, you will receive a confirmation message. Click **OK** to continue.

Using Volume Manager FlashSnap

Creating a clone database using Volume Manager FlashSnap goes through the following dialogs:

1. Create Snapplan
2. Validate/Modify Snapplan
3. Create Snapshot
4. Startup Clone Database

▼ To create a clone database using Volume Manager FlashSnap

1. In the **Clone Database Type** dialog, select create a clone database using Volume Manager FlashSnap. Click **Next**.

2. In the **VM FlashSnap Clone Database** dialog, select whether you would want to create a clone database using a new snapplan or an existing one. Click **Next**.

   If you choose **New Snapplan**, the **Create Snapplan** dialog is displayed. For details, see "Creating a Clone Database with a new snapplan" on page 287.

   If you choose **Existing Snapplan**, you are required to select a snapplan from the drop-down list, see "Creating a Clone Database with an Existing Snapplan" on page 289.

Creating a Clone Database with a new snapplan

▼ To create a Clone Database with a new snapplan

1. Confirm the Oracle SID. This is a read-only field.

2. Enter the following values:

   - Full path of the snapplan file
   - Name of the secondary host
   - Snapshot plex tag

3. Click **Next** to continue. The **Validate/Modify Snapplan** screen is displayed with default values set.

4. If needed, modify any incorrect settings. Then, click **Next** to validate the snapplan.

Note For more information regarding the snapplan parameters, see "Creating and Working with Snapplans Using dbed_vmchecksnap" on page 459.

   If the snapplan was successfully validated, you will receive a confirmation message.

5. To see the snapplan details, click the **Show details** checkbox. The snapplan details are displayed in the pop-up window.

   When you have finished reviewing them, click **OK**. The following example is a sample of the snapplan details that you should see:

```
PRIMARY_HOST is host1

SECONDARY_HOST is host2

The version of PRIMARY_DG-PRODdg is 120.

SNAPSHOT_DG is SNAP_PRODdg

SNAPSHOT_MODE is online

The database is running in archivelog mode.

ARCHIVELOG_DEST is /prod_db

SNAPSHOT_PLAN_FOR is database

SNAPSHOT_ARCHIVE_LOG is yes

ARCHIVELOG_DEST=/prod_db is mount on /dev/vx/dsk/PRODdg/ORACLE1.

Examining Oracle volume and disk layout for snapshot.

Volume ORACLE1 on PRODdg is ready for snapshot.
```

```
Original plex and DCO log for ORACLE1 is on PRODdg02.
Snapshot plex and DCO log for ORACLE1 is on PRODdg01.

Volume oracon on PRODdg is ready for snapshot.
Original plex and DCO log for oracon is on PRODdg02.
Snapshot plex and DCO log for oracon is on PRODdg01.

SNAP_PRODdg for snapshot will include: PRODdg01

ALLOW_REVERSE_RESYNC is yes

The snapplan sp3 has been created.
```

6. In the **Create Snapshot** dialog, verify the snapplan location. (This is a read-only field.)

7. If you need to force the snapshot creation, click the **Force snapshot creation** checkbox.

Note **Force snapshot creation** can be used after a snapshot operation has failed and the problem was fixed without using VERITAS Storage Foundation *for Oracle* commands. (That is, the volumes were synchronized without using VERITAS Storage Foundation *for Oracle* commands.) In this situation, the status of the snapplan will appear as unavailable for creating a snapshot. The **Force snapshot creation** option ignores the unavailable status, checks for the availability of volumes, and creates the snapshot after the volumes pass the availability check.

8. Click **Next** to continue.

9. If the snapshot creation was successful, you will receive a confirmation message.

10. Click the **Show details** checkbox to view the actions taken. The following example is a sample of the snapplan details that you should see:

```
dbed_vmsnap started at 2004-05-13 17:09:40
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 sp2 is used for the snapshot.
VxDBA repository volume is SNAP_ORACLE1.
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.
  /oracon.

dbed_vmsnap ended at 2003-05-13 17:10:29
```

11. Click **OK** to continue. You can now use the snapshot as a database backup.

Note You will need this information when you start your snapshot database.

12. The **Startup Snapshot Database** dialog is displayed. In the "connect as" section, confirm the local host name and enter the Oracle home information. Then, enter your UNIX user name and password.

Caution Make sure that the VEA service is running on the secondary host. Also, make sure that the database owner of the secondary host is registered on the VEA Service Console Registry. See "Adding Users to the VERITAS Enterprise Administrator Service Console Registry" on page 273 and "Starting the VERITAS Enterprise Administrator Service" on page 276 for further details.

13. For the primary database information section, confirm the Oracle SID. It is a read-only field.

14. Select whether you are creating the database or restarting the database by clicking the appropriate radio button.

    Select "Startup database" if you are starting a new snapshot database based on a snapshot. Select "Restart database" if you are restarting a snapshot database that has already been created.

15. For the snapshot database information section, verify the snapplan file, VxDBA volume name, and Snapshot disk group name. These are read-only fields. Enter the new Oracle SID and Relocate path in the appropriate fields.

Note To start an on-host snapshot database, the Relocate path is required. Make sure the Relocate path has the database owner's permission.

16. Click **Create Clone Database**. If the snapshot database was successfully started, you will receive a confirmation message. Click **OK** to continue.

Creating a Clone Database with an Existing Snapplan

▼ To create a Clone Database with an existing Snapplan

Select **Existing Snapplan**. You are required to select a snapplan from the drop-down list. The snapplan displays its status, for example, `sp00 [init_db] /ora_home/snapplan/sp00`, where INIT_DB is the status of the snapplan.

- If you select a snapplan with INIT_DB status, the **Validate/Modify Snapplan** dialog is displayed. Follow the steps from step 4 on page 287 in the "Creating a Clone Database with a new snapplan".
- If you select a snapplan with INIT_FULL status, the **Create Snapshot** dialog is displayed. Follow the steps from step 6 on page 288 in the "Creating a Clone Database with a new snapplan" section.
- If you select a snapplan with SNAPSHOT_END status displays the **Startup Snapshot Database** dialog is displayed. Follow the steps from step 12 on page 289 in the "Creating a Clone Database with a new snapplan" section.

Removing a Clone Database

In releases prior to 4.1, for removing a cloned database you were required to go through separate menus to shut down and unmount clone databases. VERITAS Storage Foundation *for* Oracle 4.1, provides the Remove Clone Database wizard, that can be accessed from the database instance node. The Remove Clone Database wizard helps in shutting down and unmounting the clone database. It also removes the Storage Checkpoint or resynchronizes the snapshot, if needed.

Removing a VM FlashSnap Clone Database

Removing a VM FlashSnap clone database unmounts the FlashSnap database. If needed, it also resynchronizes the snapshot.

Caution Make sure that the VEA service is running on the secondary host. Also, make sure that the database owner of the secondary host is registered on the VEA Service Console Registry. See “Adding Users to the VERITAS Enterprise Administrator Service Console Registry” on page 273 and “Starting the VERITAS Enterprise Administrator Service” on page 276 for further details.

1. In the object tree, click a clone database that was created using VM FlashSnap. You may need to expand the object tree to find the clone database.

2. Select one of the following methods to remove a clone database.

   From the menu bar, click Oracle > Remove Clone Database.

   or

   Right click the clone database object to bring up a pop-up menu. Then, click Remove Clone Database.

3. The **Remove Clone Database** dialog is displayed. In the Authentication section, verify the UNIX user name and enter the password.

   Verify the Snapplan File name and the Relocate path.

   You may check the **Resync Snapshot** option to resynchronize the snapshot.

   In the **Primary Database Information** section, verify the following information:

   - Primary Host Name
   - Database Name
   - UNIX User Name

Note You are required to enter the Password only if you check the **Resync Snapshot** option.

4. Click the Remove Clone Database button.

5. In the **Confirmation** dialog, click Yes to proceed with removing the clone database.

Removing a Storage Checkpoint Clone Database

Removing a Storage Checkpoint clone database allows you to unmount the clone database. It also gives you the option to remove the checkpoint.

1. In the object tree, click a clone database that was created using a Storage Checkpoint. You may need to expand the object tree to find the clone database.

2. Select one of the following methods to remove the clone database.

   From the menu bar, click **Oracle** > **Remove Clone Database**.

   or

   Right click the database object to bring up a pop-up menu. Then, click **Remove Clone Database**.

3. The **Remove Clone Database** screen is displayed. Verify the read-only information.

4. Check the **Remove Storage Checkpoint** option if you want to remove the Storage Checkpoint along with removing the cloned database.

5. Click the **Remove Clone Database** button.

6. In the **Confirmation** dialog, click Yes to proceed.

Using the Monitoring Agent

You can use a Monitoring Agent to manage and monitor VxFS file systems, Oracle tablespaces, and datafile space usage.

The Monitoring Agent monitors the file system space, and when the space usage reaches a configured threshold value, a predefined action script grows the file system automatically.

The agent can be enabled at boot-time. Each file system monitored has three settings that the Monitoring Agent needs to know about:

- *Warning Threshold* is a percent value (% of file system used) that determines when the agent begins warning the administrator of space shortage
- *Grow Threshold* is a percent value (% of file system used) that determines when the agent is to attempt to grow the file system (when space usage is at a critical level)
- *Amount* is either a percentage or a value in megabytes by which to grow file systems when the Grow Threshold is reached or exceeded

The VxDBA Monitoring Agent operations are driven from the following files:

- `/opt/VRTSdbed/lib/dbed_mon_config.base`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_config.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_fslist.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_oralist.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/include`

Understanding the Monitoring Agent Files

The `/opt/VRTSdbed/lib/dbed_mon_config.base` file contains the site-level configuration settings for monitoring all file systems and databases recognized. This configuration file specifies how often to check for file system and database configuration changes, how often to check the file space usage, where space usage information gets logged, and the thresholds for warning and automatically growing the file system.

For example, if you are monitoring a database named `PROD`, the database-specific file would be `/etc/vx/vxdba/PROD/dbed_mon_config.PROD`. This is the first file opened when the agent is started and contains the default settings for monitoring file systems at the database level. The Monitoring Agent cannot start without this file. Modify this configuration file if you want to change the preconfigured settings carried over from the `dbed_mon_config.base` file to maintain a different set of settings at the database level.

The files /etc/vx/vxdba/$ORACLE_SID/dbed_mon_fslist.$ORACLE_SID and /etc/vx/vxdba/$ORACLE_SID/dbed_mon_oralist.$ORACLE_SID are created and are used for restarting the Monitoring Agent. These files specify the status of the database. The files also specify the space monitoring and alarm information for each file system, tablespace, and datafile. You can edit these files manually to change settings, and then restart the Monitoring Agent.

The Monitoring Agent uses the /etc/vx/vxdba/$ORACLE_SID/include file to check that all files are up-to-date and are being monitored. This file is created by VERITAS Storage Foundation for Oracle and should not be edited.

Occasionally, Monitoring Agents ignore Storage Checkpoints. This happens when a Storage Checkpoint is not owned by the current Oracle instance. These Storage Checkpoints will not be used to calculate thresholds and potential removal candidates. Storage Checkpoints that are not considered part of the current database instance's data set are logged as such in the file /var/log/dbed_mon/dbed_mon.prune_ckpt_log.$ORACLE_SID when the Monitoring Agent is looking for potential removal candidates. A Storage Checkpoint must have an entry in the /etc/vx/vxdba/$ORACLE_SID/checkpoint_dir directory before it is considered owned by the database. This is done automatically by the provided VxDBA(1M) and dbed_ckptcreate(1M) utilities and ensures that, if multiple databases share the same file system(s), the policy for one database does not affect another.

Starting a Monitoring Agent

Prerequisites

- The dbed_update command must be run at least once before you can use the monitoring agent utility.
- Make sure you are logged on as root to perform Monitoring Agent operations.

▼ To start a Monitoring Agent

1. Click the **Monitoring Agent** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to start the Monitoring Agent.

   - From the menu bar, select **Monitoring Agent** > **Start Agent**.

     or

   - Right click the Monitoring Agent to bring up a pop-up menu. Then click **Start Agent**.

You will see a message prompting you for your `root` password.

3. Enter your `root` password. Then click **OK**.

   If the Monitoring Agent was successfully started, you will receive a confirmation message. Click **OK** to continue.

Enable / Disable the Monitoring Agent at Boot-Time

4. Optionally, if you would like to set up the Monitoring Agent to become enabled at boot-time, right click the Monitoring Agent in the object tree and select **Enabled at Boot Time**.

   or

   If you would like to set up the Monitoring Agent to become disabled at boot-time, right click the Monitoring Agent in the object tree and select **Disabled at Boot Time**.

5. Enter your `root` password. Then click **OK**.

   If the Monitoring Agent was successfully changed to be either enabled or disabled at boot time, you will receive a confirmation message. Click **OK** to continue.

   You will receive a confirmation message that the agent will either be enabled or disabled when you boot the system.

6. Click **OK** to continue.

Note To change the default Monitoring Agent values, see "Viewing or Changing Monitoring Agent Values" on page 296.

Stopping a Monitoring Agent

Prerequisites

- The `dbed_update` command must be run at least once before you can use the monitoring agent utility.
- Make sure you are logged on as `root` to perform Monitoring Agent operations.

▼ To stop a monitoring agent

1. Click the **Monitoring Agent** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to stop the Monitoring Agent.

   - From the menu bar, select **Monitoring Agent** > **Stop Agent**.

     or

   - Right click the Monitoring Agent to bring up a pop-up menu. Then click **Stop Agent**.

   You will then see a message prompting you for your `root` password.

3. Enter your `root` password. Then click **OK**.

   If the Monitoring Agent was successfully stopped, you will receive a confirmation message. Click **OK** to continue.

Viewing or Changing Monitoring Agent Values

You can view the settings of your active Monitoring Agent. You also have the option to change the default values of the Monitoring Agent, if needed.

Prerequisites

- The `dbed_update` command must be run at least once before you can use the monitoring agent utility.
- Make sure you are logged on as `root` to perform Monitoring Agent operations.
-

▼ To view the Monitoring Agent values

Click the **Monitoring Agent** icon in the object tree. (You may need to expand the tree view to find the icon.)

The values associated with the Monitoring Agent are displayed on the screen.

▼ To change the Monitoring Agent values

1. Click the **Monitoring Agent** icon in the object tree. (You may need to expand the tree view to find the icon.) The Monitoring Agent values appear on the right side of the screen.

2. On the right side of the screen, double-click the field you want to change and enter a new value. You may continue to double-click the fields you want to change and enter new values.

3. Click **Save** to save your changes.

Managing Storage Checkpoints

A Storage Checkpoint is like an online backup of a database that contains a point-in-time database image. Storage Checkpoints can later be used to restore the image of a datafile, a tablespace, or the entire database to any earlier state recorded by the Storage Checkpoints.

VERITAS Storage Foundation *for Oracle* uses the VxDBA repository to determine the list of tablespaces, datafiles, and file systems for Storage Checkpoint creation and removal.

For more information about Storage Checkpoints, see “Understanding Storage Checkpoints and Storage Rollback” on page 156.

Creating a Storage Checkpoint

You have the option to create an offline, online, or instant Storage Checkpoint. If the database is online when the Storage Checkpoint is created, VERITAS Storage Foundation *for Oracle* switches the database to online backup mode before creating the Storage Checkpoint. Once the Storage Checkpoint is created, VERITAS Storage Foundation *for Oracle* switches the database back to its normal operation mode.

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. After the rollback is complete, you need to perform Oracle database instance recovery. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

In addition to creating a Storage Checkpoint, the GUI automatically backs up the associated control files, initialization file, and log information. Suppose that you made a structural change to your database, and then needed to roll back the database to a Storage Checkpoint that was created before the structural change. The Storage Rollback would only be successful if you could also reconstruct the database to the same structure that it was when the Storage Checkpoint was created. You can recreate the previous database structure using the control files, initialization file, and log information that were backed up when the Storage Checkpoint was created.

Prerequisites

- Enable `ARCHIVELOG` mode before taking online Storage Checkpoints. See “Backing Up and Recovering the Database Using Storage Checkpoints” on page 169and“Using Storage Checkpoints and Storage Rollback for Backup and Restore” on page 156.

▼ To create a Storage Checkpoint

1. Click the **Storage Checkpoints** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to create a Storage Checkpoint.

   ◆ From the menu bar, select **Storage Checkpoints** > **Create Storage Checkpoint**.

   or

   ◆ Right click the **Storage Checkpoints** icon to bring up a pop-up menu. Then, click **Create Storage Checkpoint**.

   The Create a Storage Checkpoint wizard is displayed.

3. Verify the Oracle SID, which is the Oracle System Identifier, for which you are creating the Storage Checkpoint. This is a read-only field.

4. Select one of the following options:

   ◆ To create an online Storage Checkpoint, click the **Online** button.

   ◆ To create an offline Storage Checkpoint, click the **Offline** button.

   ◆ To create an instant Storage Checkpoint, click the **Instant** button.

5. If you want to remove the Storage Checkpoint when the file system becomes full, click the **Remove This Storage Checkpoint** button.

   or

   If you want to retain the Storage Checkpoint when the file system becomes full, click the **Retain This Storage Checkpoint** button.

6. Click **Create** to continue.

7. At the prompt, click **Yes** to proceed with creating the Storage Checkpoint.

   If the Storage Checkpoint was successfully created, you will receive a confirmation message. To see details, click the **Show Details** checkbox. The details are displayed in a pop-up window.

   Click **OK** to continue.

Viewing Storage Checkpoint Details

▼ To view the details of a Storage Checkpoint

1. Click a specific **Storage Checkpoint** in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

   The Storage Checkpoint information is displayed on the right side of the window.

2. To view file system quota information for the Storage Checkpoint, click the **File System Quota** tab at the top of the window, just above the Storage Checkpoint details.

Note The file system quota information is only available for disk layout version 6.

Mounting a Storage Checkpoint

You can mount and write to Storage Checkpoints just as you can do with any file system. See "Using Storage Checkpoints and Storage Rollback for Backup and Restore" on page 156 for more information.

▼ To mount a Storage Checkpoint

1. Click a specific Storage Checkpoint in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

2. Select one of the following methods to mount the Storage Checkpoint.

   - From the menu bar, select **Storage Checkpoint** > **Mount Storage Checkpoint**.

     or

   - Right click the Storage Checkpoint you want to mount to bring up a pop-up menu. Then click **Mount Storage Checkpoint**.

   The Mount a Storage Checkpoint wizard then displays.

3. Verify that you are mounting the correct Storage Checkpoint and click **Next** to continue. If you selected the wrong Storage Checkpoint, click **Cancel**. The information on this screen is read-only.

4. On the second screen, enter the mount point (absolute path) where the Storage Checkpoint should be mounted.

Note The directory containing the mount point must be writable by the database administrator group. You should have created this group during installation. If not, create the group before mounting the Storage Checkpoint.

5. To mount the Storage Checkpoint as read-only, click **Read Only**.

   or

   To mount the Storage Checkpoint as read-write, click **Read/Write**. This will allow you to make changes to the Storage Checkpoint.

Note When you select the **Read/Write** option, the GUI creates an identical Storage Checkpoint with the same name plus a `wr001` suffix, where *001* is a sequential number. The GUI mounts the new Storage Checkpoint and leaves the original Storage Checkpoint unmounted. This allows you to roll back to the original Storage Checkpoint.

6. Click **Mount** to mount the Storage Checkpoint.

7. At the prompt, click **Yes** to proceed with mounting the Storage Checkpoint.

   When you have finished viewing the details in the Confirmation dialog, click **OK** to continue.

Unmounting a Storage Checkpoint

▼ To unmount a Storage Checkpoint

1. Click a specific Storage Checkpoint in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

Note If you want to unmount a Storage Checkpoint that was originally mounted with the **Read/Write** option, you should unmount the new Storage Checkpoint that was automatically created by the GUI, which is the Storage Checkpoint that contains the `wr001` suffix, where *001* is a sequential number, at the end of the name.

2. Select one of the following methods to unmount the Storage Checkpoint.

   ◆ From the menu bar, select **Storage Checkpoint** > **Unmount a Storage Checkpoint**.

     or

- Right click the Storage Checkpoint you want to unmount to bring up a pop-up menu. Then click **Unmount Storage Checkpoint**.

3. Verify that you are unmounting the correct Storage Checkpoint and click **Unmount** to continue. If you selected the wrong Storage Checkpoint, click **Cancel**. The information on this screen is read-only.

4. At the prompt, click **Yes** to proceed with unmounting the Storage Checkpoint.

5. A confirmation dialog is displayed. Click **OK** to continue.

Removing a Storage Checkpoint

Occasionally, you may need to manually remove Storage Checkpoints that are no longer needed. For example, you can remove a Storage Checkpoint on a file system to free up needed space.

Prerequisites

- Before you can remove a mounted Storage Checkpoint, you must first unmount it.

▼ To remove a Storage Checkpoint

1. Click a specific Storage Checkpoint in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

2. Select one of the following methods to remove the Storage Checkpoint.

   - From the menu bar, select **Storage Checkpoint** > **Remove Storage Checkpoint**.

     or

   - Right click the Storage Checkpoint you want to remove to bring up a pop-up menu. Then click **Remove Storage Checkpoint**.

3. At the prompt, click **Yes** to remove the Storage Checkpoint.

   If the Storage Checkpoint was successfully removed, you will receive a confirmation message. Click **OK** to continue.

Rolling Back to a Storage Checkpoint

You can roll back a database file, a list of database files, a single tablespace, or the entire database to a Storage Checkpoint. To perform a Storage Rollback, you must have a valid Storage Checkpoint. This option is not available with an instant Storage Checkpoint.

Note The GUI does not automatically roll back the control file associated with a Storage Checkpoint. See "Guidelines for Oracle Recovery" on page 179 for information on database recovery.

Note You must be the Database Administrator to perform Storage Rollback operations. You must shut down the instance to perform full Storage Rollback of the database, or you can choose to leave the database up to roll back a datafile or tablespace. In this situation, VERITAS Storage Foundation *for Oracle* checks to see if the target database objects are offline before proceeding. See "Backing Up and Recovering the Database Using Storage Checkpoints" on page 169 and"Guidelines for Oracle Recovery" on page 179 for more information.

Storage Checkpoints can only be used to roll back files that are damaged due to a software error or a human error (for example, accidental deletion of a table). Because Storage Checkpoints reside on the same physical disks as the primary file system, when a file is corrupted due to a media failure, the file on the Storage Checkpoints will not be available either. In this case, you need to restore files from a tape backup.

After the files are rolled back, you may need to follow the recovery procedure described in your Oracle manuals to recover the database before the database can be used.

Note Some database changes, made after a Storage Checkpoint was taken, may make it impossible to run Storage Rollback successfully. For example, you cannot successfully run Storage Rollback if the control files for the database have recorded the addition or removal of datafiles. To provide recovery options, a backup copy of the control file for the database is saved under the `/etc/vx/vxdba/`*`ORACLE_SID`*`/checkpoint_dir/`*`CKPT_NAME`* directory just after a Storage Checkpoint is created. You can use this file to assist with database recovery, if necessary. If possible, both an ASII and binary version of the control file will be left under the `/etc/vx/vxdba/`*`ORACLE_SID`*`/checkpoint_dir/`*`CKPT_NAME`* directory, with the binary version being compressed to conserve space. Use extreme caution when recovering your database using alternate control files.

Rolling Back the Database to a Storage Checkpoint

Rolling back the entire database rolls back all the datafiles used by the database, except the redo logs and control files, to a Storage Checkpoint.

Note While the Storage Rollback process is running, it creates a temporary file, `/`*`filesystem`*`/.VRTSstrb.lock`, in each file system. Do not remove these temporary lock files.

▼ To rollback the database to a Storage Checkpoint

1. Shut down the Oracle database from the GUI.

2. Click the a specific Storage Checkpoint in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

3. Select one of the following methods to rollback to the selected Storage Checkpoint.

   ◆ From the menu bar, select **Storage Checkpoint** > **Roll Back Storage Checkpoint**.

   or

   ◆ Right click the Storage Checkpoint to which you want to rollback to bring up a pop-up menu. Then, click **Roll Back Storage Checkpoint**.

   The Rollback a Storage Checkpoint wizard is displayed.

4. Verify that you are rolling back to the correct Storage Checkpoint and click **Next** to continue. If you selected the wrong Storage Checkpoint, click **Cancel**. The information on this screen is read-only.

5. On the second screen, use the drop-down menu to select the appropriate buffer size in the Rollback Buffer Size field. The default buffer size is 128K.

Note The buffer size configured for reads and writes when performing a Storage Rollback can affect performance. Vary the size to determine the optimal setting for your system.

6. Use the drop-down menu to select the appropriate number of threads in the Number of Threads field. The default number of threads is four.

Note Depending on the number of CPUs available on your system and the type of volume on which the file system is located, this default setting may specify too few or too many threads.

7. Click the **Roll back a database** button to indicate that you are rolling back the entire database to the Storage Checkpoint.

8. Click **Next** to continue. Click **Roll Back** to continue.

   If the Storage Rollback was successful, a confirmation message is displayed.

9. To see the Storage Rollback details, click the **Show details** checkbox. The details are displayed in the pop-up window.

   When you have finished viewing the details, click **OK**.

10. Click **Yes** to roll back the tablespace or tablespaces. Perform any necessary Oracle recovery. See "Guidelines for Oracle Recovery" on page 179 for more information. (You cannot recover your database through the GUI.)

11. Restart the database from the GUI.

Rolling Back a Tablespace to a Storage Checkpoint

If a tablespace is corrupted or removed due to a software error or a human mistake, this operation rolls back all of the files of the corrupted or removed tablespace to a Storage Checkpoint.

This option is only available for offline and online Storage Checkpoints. You cannot roll back a tablespace to an instant Storage Checkpoint.

Rolling back a tablespace is used for complete recovery of the tablespace. It is not designed for point-in-time (incomplete) tablespace recovery, which is more complicated and requires interaction with Oracle Customer Support. The tablespace point-in-time recovery requires using a clone database. See "Creating Clone Database" on page 285 for more information.

Note You can perform this operation while the database is online as long as the tablespace is offline.

▼ To roll back a tablespace to a Storage Checkpoint

1. Verify that the tablespace to which you want to roll back is offline.

2. Click the a specific Storage Checkpoint in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

3. Select one of the following methods to rollback to the selected Storage Checkpoint.

   ◆ From the menu bar, select **Storage Checkpoint** > **Rollback a Storage Checkpoint**.

   or

- ◆ Right click the Storage Checkpoint to which you want to rollback to bring up a pop-up menu. Then, click **Rollback a Storage Checkpoint**.

If the database is online, you will receive a prompt asking you if you want to continue. Click **Yes** to continue the Storage Rollback.

The Rollback a Storage Checkpoint wizard is displayed.

4. Verify that you are rolling back to the correct Storage Checkpoint and click **Next** to continue. If you selected the wrong Storage Checkpoint, click **Cancel**. The information on this screen is read-only.

5. On the second screen, use the drop-down menu to select the appropriate buffer size in the Rollback Buffer Size field. The default buffer size is 128K.

Note The buffer size configured for reads and writes when performing a Storage Rollback can affect performance. Vary the size to determine the optimal setting for your system.

6. Use the drop-down menu to select the appropriate number of threads in the Number of Threads field. The default number of threads is four.

Note Depending on the number of CPUs available on your system and the type of volume on which the file system is located, this default setting may specify too few or too many threads.

7. Click the **Rollback a Tablespace** button to indicate that you are rolling back the tablespace to the Storage Checkpoint.

8. Click **Next** to continue. Select the tablespace, or tablespaces, you would like to roll back by clicking the appropriate checkbox in the Tablespace list.

9. Click **Roll Back** to continue.

10. If the Storage Rollback was successful, you will receive a confirmation message. Click **OK** to continue.

11. Click **Yes** to rollback the datafile or datafiles. Perform any necessary Oracle media recovery. See "Guidelines for Oracle Recovery" on page 179 for more information. (You cannot recover database information through the GUI.)

12. Put the recovered tablespace back online.

Rolling Back Datafiles to a Storage Checkpoint

Rolling back datafiles rolls back database files to a Storage Checkpoint. You can also use this operation to roll back more than one datafile.

This option is only available for offline and online Storage Checkpoints. You cannot roll back a datafile to an instant Storage Checkpoint.

Note You can perform this operation while the tablespace is online as long as the datafile is offline.

▼ To roll back datafiles to a Storage Checkpoint

1. Verify that the datafile to which you want to roll back is offline.

2. Click the a specific Storage Checkpoint in the object tree. (You may need to expand the tree view to find the Storage Checkpoint.)

3. Select one of the following methods to rollback to the selected Storage Checkpoint.

   ◆ From the menu bar, select **Storage Checkpoint** > **Rollback a Storage Checkpoint**.

   or

   ◆ Right click the Storage Checkpoint to which you want to rollback to bring up a pop-up menu. Then click **Rollback a Storage Checkpoint**.

   If the database is online, you will receive a prompt asking you if you want to continue. Click **Yes** to continue the Storage Rollback.

   The Rollback a Storage Checkpoint wizard is displayed.

4. Verify that you are rolling back to the correct Storage Checkpoint and click **Next** to continue. If you selected the wrong Storage Checkpoint, click **Cancel**. The information on this screen is read-only.

5. On the second screen, use the drop-down menu to select the appropriate buffer size in the Rollback Buffer Size field. The default buffer size is 128K.

Note The buffer size configured for reads and writes when performing a Storage Rollback can affect performance. Vary the size to determine the optimal setting for your system.

6. Use the drop-down menu to select the appropriate number of threads in the Number of Threads field. The default number of threads is four.

Note Depending on the number of CPUs available on your system and the type of volume on which the file system is located, this default setting may specify too few or too many threads.

7. Click the **Roll back a Datafile** button to indicate that you are rolling back the datafile to the Storage Checkpoint.

8. Click **Next** to continue.

9. Select the datafile, or datafiles, you would like to roll back by clicking on the appropriate checkbox in the Datafile list.

10. Click **Roll Back** to continue.

11. If the Storage Rollback was successful, you will receive a confirmation message. Click **OK** to continue. You are then returned to the rollback window.

12. Perform any necessary Oracle media recovery. See “Guidelines for Oracle Recovery” on page 179 for more information.

13. Put the recovered datafile back online.

Using a Storage Checkpoint Policy for Space Management

A Storage Checkpoint policy establishes how many Storage Checkpoints you would like to keep in the event your file system becomes full. You can also determine what to do with old Storage Checkpoints when you run out of space. You should set up a Storage Checkpoint policy before creating any Storage Checkpoint schedules.

Prerequisites

◆ You must enable the Monitoring Agent to use the Storage Checkpoint policy.

Creating a Storage Checkpoint Policy

▼ To create a Storage Checkpoint policy

1. Click the **Storage Checkpoints** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to create a Storage Checkpoint policy.

Right click the Storage Checkpoint icon to bring up a pop-up menu. Then, click **Policy**.

The **Storage Checkpoint Policy** window is displayed.

3. Enter the maximum number of Storage Checkpoints to keep on the file system in the Maximum field.

4. To enable the policy, click the **Enable** button.

5. To remove old Storage Checkpoints if the maximum number is exceeded, click **Remove Old Storage Checkpoints**.

   or

   If you would like to keep old Storage Checkpoints after the maximum number is exceeded, click **Retain Old Storage Checkpoints**. (This option will prevent any new Storage Checkpoints from being created if the file system runs out of space.)

6. Click **Policy Update** to save your changes.

   If the Storage Checkpoint policy was successfully created, you will receive a confirmation message. Click **OK** to continue.

Note If your Monitoring Agent is disabled, you will receive warning stating that you must active the Monitoring Agent in order to use the Storage Checkpoint policy.

Disabling a Storage Checkpoint Policy

▼ To disable a Storage Checkpoint policy

1. Click the **Storage Checkpoints** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to disable a Storage Checkpoint policy.

   Right click the Storage Checkpoint icon to bring up a pop-up menu. Then, click **Policy**.

   The **Storage Checkpoint Policy** window is displayed.

3. To disable the policy, click the **Disable** button.

4. Click **Policy Update** to save your changes.

   If the Storage Checkpoint policy was successfully disabled, you will receive a confirmation message. Click **OK** to continue.

Managing Snapshots with Database FlashSnap

With VERITAS Database FlashSnap, you can create a point-in-time copy of a database for backup and off-host processing. For more information, see “Using Database FlashSnap for Backup and Off-Host Processing” on page 197.

From the GUI, you can create snapshots of your database using snapplans. You can also resynchronize snapshots to your database and your database to a snapshot. The following sections describe the Database FlashSnap features within the GUI.

Database FlashSnap functionality is not supported on duplicated (cloned) databases.

Prerequisites

- Before you can create a snapshot, a snapshot mirror of a volume must exist. For details, see “Creating a Snapshot Mirror of a Volume or Volume Set Used by the Database” on page 206.

Creating a Snapplan

A snapplan specifies snapshot scenarios for an Oracle database (such as online, offline, and instant). The snapplan is used as a basis for creating a snapshot. You must either create or validate a snapplan before you can create the snapshot image.

Prerequisites

- You must be logged in as the Oracle database administrator.
- The disk group must be version 110 or higher.
- Persistent FastResync must be enabled.
- To set up your snapplan for online snapshots, the database must be in archive log mode.

Usage Notes

- Database FlashSnap functionality is not supported on cloned (duplicated) databases.
- The snapplan name is user-defined.
- Each entry in the snapplan is a line in *parameter=argument* format.
- See the `dbed_vmsnapplan`(1M) and `dbed_vmchecksnap`(1M) manual page for more information.

▼ To create a snapplan

1. Click the **Snapplans** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to create a snapshot plan.

   ◆ From the menu bar, select **Snapplans** > **Create Snapplan**.

   or

   ◆ Right click the Snapshot Plans icon to bring up a pop-up menu. Then click **Create Snapplan**.

   The Create Snapplan wizard is then displayed.

3. Confirm the Oracle SID. This is a read-only field.

4. Enter the full path of the snapplan file.

5. Enter the name of the secondary host.

6. Enter the snapshot plex tag.

7. Click **Next** to continue.

   The Validate/Modify Snapplan screen is displayed with default values set.

8. If needed, modify any incorrect settings. Then, click **Validate** to ensure that the settings have been configured correctly.

Note For more information regarding the snapplan parameters, see "Creating and Working with Snapplans Using dbed_vmchecksnap" on page 459.

9. At the confirmation prompt, click **Yes** to confirm that you want to validate the snapplan.

   If the snapplan was successfully validated, you will receive a confirmation message.

10. To see the snapplan details, click the **Show details** checkbox. The snapplan details are displayed in the pop-up window.

    When you have finished reviewing them, click **OK**. The following example is similar to what you should see:

```
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 /prod_db

SNAPSHOT_PLAN_FOR is database

SNAPSHOT_ARCHIVE_LOG is yes

ARCHIVELOG_DEST=/prod_db is mount on /dev/vx/dsk/PRODdg/ORACLE1.

Examining Oracle volume and disk layout for snapshot.

Volume ORACLE1 on PRODdg is ready for snapshot.
Original plex and DCO log for ORACLE1 is on PRODdg02.
Snapshot plex and DCO log for ORACLE1 is on PRODdg01.

Volume oracon on PRODdg is ready for snapshot.
Original plex and DCO log for oracon is on PRODdg02.
Snapshot plex and DCO log for oracon is on PRODdg01.

SNAP_PRODdg for snapshot will include: PRODdg01

ALLOW_REVERSE_RESYNC is yes

The snapplan sp3 has been created.
```

Validating or Modifying a Snapplan

Before creating a snapshot, it is necessary to validate your snapplan to make sure the information is correct and that the snapshot will succeed. Occasionally, you may also need to modify a snapplan. For example, you would need to modify your snapplan if you wanted to change the primary disk group that is included in the snapshot.

Prerequisites

- You must be logged in as the Oracle database administrator.
- A DCO log must be attached to the snapshot plex.

Usage Notes

- Each entry in the snapplan is a line in *parameter=argument* format.

▼ To validate or modify a snapplan

1. Click a specific snapplan in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to validate or modify a snapshot plan.

   - From the menu bar, select **Snapplan** > **Modify/Validate Snapplan**.

     or

   - Right click the snapplan to bring up a pop-up menu. Then click **Modify/Validate Snapplan**.

   The Modify/Validate Snapplan wizard is displayed.

3. If incorrect, enter the path and file name of the snapplan in the **Snapplan file** field. Then, click **Next** to continue.

4. If you are modifying your snapplan, enter any new parameters that should be updated. You are not required to do anything if no changes are required.

Note For more information regarding the snapplan parameters, see "Creating and Working with Snapplans Using dbed_vmchecksnap" on page 459.

5. Click **Validate** to check that the settings have been configured correctly.

6. At the confirmation prompt, click **Yes** to confirm that you want to validate the snapplan.

   If the snapplan was successfully validated, you will receive a confirmation message.

7. To see the snapplan details, click the **Show details** checkbox. The snapplan details are displayed in the pop-up window.

   When you have finished reviewing them, click **OK**. The following example is similar to what you should see:

```
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 /prod_db
```

```
SNAPSHOT_PLAN_FOR is database

SNAPSHOT_ARCHIVE_LOG is yes

ARCHIVELOG_DEST=/prod_db is mount on /dev/vx/dsk/PRODdg/ORACLE1.

Examining Oracle volume and disk layout for snapshot.

Volume ORACLE1 on PRODdg is ready for snapshot.
Original plex and DCO log for ORACLE1 is on PRODdg02.
Snapshot plex and DCO log for ORACLE1 is on PRODdg01.

Volume oracon on PRODdg is ready for snapshot.
Original plex and DCO log for oracon is on PRODdg02.
Snapshot plex and DCO log for oracon is on PRODdg01.

SNAP_PRODdg for snapshot will include: PRODdg01

ALLOW_REVERSE_RESYNC is yes

The snapplan sp3 has been created.
```

Removing a Snapplan

You may need to remove a snapplan that is no longer necessary. However, you cannot create a snapshot if you do not have a snapplan.

Prerequisites

- You must be logged in as the Oracle database administrator.

▼ To remove a snapplan

1. Click the snapplan you want to remove in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to remove the snapplan.

   - From the menu bar, select **Snapplan** > **Remove Snapplan**.

     or

   - Right click the snapplan to bring up a pop-up menu. Then click **Remove Snapplan**.

     A pop-up window appears asking you to confirm whether you want to remove the snapplan.

3. At the prompt, click **Yes** to continue with removing the snapplan.

   If the snapplan was successfully removed, you will receive a confirmation message similar to the following:

   ```
   Successfully removed snapplan: sp1
   ```

4. Click the **Show details** checkbox to view the actions taken. You should see information similar to the following:

   ```
   The snapplan <filename> has been removed.
   ```

   Click **OK** to continue.

Creating a Snapshot

After having created and validated a snapplan, you can create a snapshot of your database. You can use the snapshot as a database backup or as a test database to perform operations without affecting your production database.

Prerequisites

- You must have a validated snapplan before creating a snapshot image.
- You must be logged in as the Oracle database administrator.

Usage Notes

- See the dbed_vmsnap(1M) manual page for more information.
- If you are creating an offhost snapshot, perform these steps on the secondary host.

▼ To create a snapshot

1. Click a snapplan in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to create a snapshot.

   - From the menu bar, select **Snapplan** > **Create Snapshot**.

     or

   - Right click the snapplan to bring up a pop-up menu. Then click **Create Snapshot**.

   The Create Snapshot wizard is displayed.

3. Verify the snapplan location. (This is a read-only field.)

4. If you need to force the snapshot creation, click the **Force snapshot creation** checkbox.

   Note **Force snapshot creation** can be used after a snapshot operation has failed and the problem was fixed without using VERITAS Storage Foundation *for Oracle* commands. (That is, the volumes were synchronized without using VERITAS Storage Foundation *for Oracle* commands.) In this situation, the status of the snapplan will appear as unavailable for creating a snapshot. The **Force snapshot creation** option ignores the unavailable status, checks for the availability of volumes, and creates the snapshot after the volumes pass the availability check.

5. Click **Finish** to continue.

6. If the snapshot creation was successful, you will receive a confirmation message.

7. Click the **Show details** checkbox to view the actions taken. You should see information similar to the following:

   ```
   dbed_vmsnap started at 2004-05-13 17:09:40
   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 sp2 is used for the snapshot.
   VxDBA repository volume is SNAP_ORACLE1.
   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.
     /oracon.

   dbed_vmsnap ended at 2003-05-13 17:10:29
   ```

   Click **OK** to continue.

   You can now use the snapshot as a database backup.

Note You will need this information when you start your snapshot database.

Creating a Clone Database With a Snapshot

After having created a snapshot, you can use the snapshot to create a clone database. The cloned database can be used for decision-making and testing that cannot be done on your production database.

Prerequisites

- Make sure you have enough disk space to create a clone database on your system.
- You must have an existing snapshot.

▼ To create a clone database with a snapshot

1. Click the **Oracle** icon in the object tree. (You may need to expand the tree view to find the icon.)
2. Select one of the following methods to start the snapshot database.
   - From the menu bar, select **Oracle Databases** > **Start Up Snapshot Database**.

     or
   - Right click the **Oracle Databases** icon to bring up a pop-up menu. Then click **Start Up Snapshot Database**.

   The Start Up Snapshot Database wizard screen is then displayed.
3. In the “connect as” section, enter the local path to Oracle home. Then, enter your Unix user name, password.
4. For the primary database, enter the Oracle SID information.
5. Select whether you are recovering the database or restarting the database by clicking the appropriate radio button.

   Select **Startup database** if you are restarting a new snapshot database based on a snapshot. Select **Restart database** if you are restarting a snapshot database that has already been created.
6. For the snapshot database, enter the new Oracle SID, Snapplan file, VxDBA volume name, Snapshot disk group name, and Relocate path in the appropriate fields.

Note To start an on-host snapshot database, the Relocate path is required. To start an off-host snapshot database, the VxDBA volume name field is required. You can obtain this information by viewing the details after you create a snapshot. If you did not retain the information from the snapshot creation, you can use the `dbed_vmchecksnap` command or use Viewlog via the GUI to retrieve the information. For more information, see “Creating and Working with Snapplans Using dbed_vmchecksnap” on page 459.

7. Click **Start Database**.

8. At the confirmation prompt, click **Yes** to confirm that you want to start the database.

   If the snapshot database was successfully started, you will receive a confirmation message. Click **OK** to continue.

Resynchronizing a Snapshot to a Database

Resynchronizing a snapshot to a database will refresh the snapshot so that it contains the most recent changes made to your production database.

Prerequisites

- You must be logged in as the Oracle database administrator.
- You must shut down the clone database and unmount the file systems.

Usage Notes

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

▼ To resynchronize a snapshot to a database

1. Click a snapplan, located under the **Snapplans** icon, in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to resynchronize the snapshot.

   - From the menu bar, select **Snapplan** > **Resync Snapshot**.

     or

   - Right click the snapplan to bring up a pop-up menu. Then click **Resync Snapshot**.

3. At the confirmation prompt, click **OK** to continue resynchronizing the snapshot to the database.

4. If the resynchronization was successful, you will receive a confirmation message. Click **OK** to continue.

5. Status information similar to the following example is displayed. Click **OK** when you have finished viewing the information.

```
dbed_vmsnap started at 2003-05-13 17:20:05
The option resync has been completed.
dbed_vmsnap ended at 2003-05-13 17:20:41
```

   To see the details, click the Show details checkbox. The details are displayed in a pop-up window.

Resynchronizing a Database to a Snapshot

Resynchronizing your database to a snapshot, also known as reverse resynchronization, reverts your database to a snapshot. Use this option if your database becomes corrupted and you need to restore your database to a previous point-in-time.

Prerequisites

- You must be logged in as the Oracle primary database administrator.
- You must shut down the primary database and the clone database; also, unmount the file systems.

Usage Notes

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

▼ To resynchronize a database to a snapshot

1. Click a snapplan, located under the **Snapplans** icon, in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to resynchronize the database to a snapshot.

   - From the menu bar, select **Snapplan** > **Reverse Resync Snapshot**.

     or

   - Right click the snapplan to bring up a pop-up menu. Then click **Reverse Resync Snapshot**.

   The Reverse Resync wizard is displayed.

3. Verify the Oracle SID and snapplan information. These are read-only fields.

4. Click the Begin button, then click **Finish** to start the reverse resynchronization process. To view details, click the **Show details** checkbox. The details are displayed in a pop-up window.

5. At the confirmation prompt, click **Yes** to continue.

   The Begin option performs the following actions:

   - Imports the disk group that was deported from the secondary host and joins it back to the original disk group.
   - Mounts the snapshot volumes.
   - Mounts the file systems that are configured for the primary database.
   - Brings up the database snapshot image as the primary database.

   Note The primary database must be offline to perform this action.

6. If the begin action was successful, you will receive a confirmation message. Click **Show details** to see the actions completed. When you are through, click **OK** to continue.

7. Again, click the snapplan on which you want to perform the reverse resynchronization.

8. Select one of the following methods to resynchronize the database to a snapshot.

   - From the menu bar, select **Snapplan** > **Reverse Resync Snapshot**.

     or

   - Right click the snapplan to bring up a pop-up menu. Then click **Reverse Resync Snapshot**.

   The Reverse Resync wizard is displayed.

9. Click the **Commit** button, then click **Finish** to commit the reverse resynchronization process.

10. At the confirmation prompt, click **Yes** to continue.

    The Commit option performs the following actions:

    - Commits the reverse resynchronization changes.
    - Resynchronizes the original volume from the data in the snapshot and then discards the content of the original database.

Note This action cannot be undone.

11. If the commit action was successful, you will receive a confirmation message. Click **Show details** to see the actions completed. When you are through, click **OK** to continue.

Aborting the Reverse Resychronization Operation

Occasionally, you may need to stop the reverse resynchronization process after you have begun. You can only abort the reverse resynchronization process after you have completed the Begin operation and before performing a Commit operation.

▼ To abort the reverse resynchronization operation

1. Click the snapplan for which the reverse resynchronization was begun.

2. Select one of the following methods to resynchronize the database to a snapshot.

   ◆ From the menu bar, select **Snapplan** > **Reverse Resync Snapshot**.

   or

   ◆ Right click the snapplan to bring up a pop-up menu. Then click **Reverse Resync Snapshot**.

   The Reverse Resync wizard is displayed.

3. Verify the Oracle SID and snapplan information. These are read-only fields.

4. Click the Abort button, then click **Finish** to abort the reverse resynchronization process.

5. At the confirmation prompt, click **Yes** to continue.

   The Abort option performs the following actions:

   ◆ Unmounts the snapshot volumes.

   ◆ Mounts the original volumes back with the file systems that are configured to use the volume.

Note This action can only be performed after a “begin” action has been completed and cannot be used after a reverse resynchronization has been committed.

6. If the abort action was successful, you will receive a confirmation message. Click **Show details** to see the actions completed. When you are through, click **OK** to continue.

Viewing the Snapplan Log

The Snapplan Log displays information about the Snapplan, disk group, snapshot Plex tag. It also displays the VxDBA volume name and the snapshot status.

▼ To view a Snapplan Log

1. In the object tree, click the snapplan whose log you wish to see. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to view the log of a snapplan.

   ◆ From the menu bar, select **Snapplan** > **View Log**.

   or

   ◆ Right click the snapplan to bring up a pop-up menu. Then click **View Log**.

   A pop-up window appears displaying the path of the Snapplan and the Log. The information displayed is similar to the following output:

```
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=exmoor
SECONDARY_HOST=exmoor
PRIMARY_DG=oradg
SNAPSHOT_DG=SNAP_oradg
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=yes

STORAGE_INFO
oradg03
SNAP_PLEX=prod_db-02 prod_ar-02

VXDBA_VOLUME_INFO
SNAP_prod_ar

STATUS_INFO
```

```
SNAP_STATUS=resync_start
DB_STATUS=init
LOCAL_SNAPPLAN=/oracle9i/snapplan/sp0
```

3. Click **OK** to continue.

Maintaining Your System Configuration

VERITAS Storage Foundation *for Oracle* maintains a repository that stores the pertinent information needed to display configuration information. This repository is located at `/etc/vx/vxdba/$ORACLE_SID`. When the database configuration changes, the information stored in the repository may not be up-to-date. You can rescan, or refresh, database instances, datafiles, and tablespaces to make sure system information is up-to-date.

The GUI also allows you to check and save the configuration of each Oracle database instance on your system. Information on all volumes, file systems and their types, and disk groups can be displayed.

Rescanning the Database Instance

1. Click the **Oracle databases** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to rescan the database instance.

   ◆ From the menu bar, select **Oracle** > **Rescan**.

   or

   ◆ Right click the Oracle database instance icon to bring up a pop-up menu. Then click **Rescan**.

3. If the rescan was successful, you will receive a confirmation message. Click **OK** to continue.

Displaying and Rescanning Tablespace Information

▼ To display tablespace information

Click a tablespace in the object tree. (You may need to expand the tree view to find the icon.) The tablespace information is displayed on the right side of the window.

▼ To rescan tablespace information

1. Click the **Tablespaces** icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to refresh the tablespace information.

   - From the menu bar, select **Tablespace** > **Rescan**.

     or

   - Right click the Tablespaces icon to bring up a pop-up menu. Then click **Rescan**.

3. If the refresh was successful, you will receive a confirmation message. Click **OK** to continue.

Viewing Oracle Datafile Topology or Statistics

1. Click a datafile in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to generate datafile statistics.

   - From the menu bar, select **Datafiles** > **Topology/Statistic**.

     or

   - Right click the datafile to bring up a pop-up menu. Then click **Topology/Statistic**.

   The Topology/Statistic page is displayed.

3. To view detailed information, click an object in the tree and then click the **Detailed Information** tab at the bottom of the page.

4. Optionally, if you want I/O statistics, enter a sampling time (in minutes) in the **Sampling interval (minutes)** field and then press **Get Statistics**.

   Information is displayed in the Operations, File Blocks, and Average Time (ms) columns of the page. A confirmation dialog is displayed. Click **OK** to continue.

5. When you are through, click the **Close** button to quit.

Determining the Datafile File Type

You can view the datafile properties to determine if the datafile is a Quick I/O file, an Oracle Disk Manager file, or a regular VxFS file.

▼ To determine the datafile file type

1. Click a datafile in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to view datafile properties.

    ◆ From the menu bar, select **File** > **Properties**.

    or

    ◆ Right click the datafile to bring up a pop-up menu. Then click **Properties**.

    The Properties pop-up window is displayed.

3. Click **OK** when you have finished viewing the datafile properties.

Converting Regular Datafiles to Quick I/O Files

VERITAS Storage Foundation *for Oracle* provides an option to convert your regular datafiles to Quick I/O files to improve performance.

Prerequisites

◆ Files you want to convert must be regular datafiles on VxFS file systems or links that point to regular VxFS files.

Usage Notes

◆ Converting existing datafiles to be Quick I/O files may not be the optimal thing to do if these files are fragmented.

▼ To convert Oracle datafiles to Quick I/O files

1. Shut down the database from the GUI.

2. Click a datafile in the object tree. (You may need to expand the tree view to find the icon.)

3. Select one of the following methods to generate datafile statistics.

   ◆ From the menu bar, select **Datafiles** > **Conversion**.

   or

   ◆ Right click the datafile to bring up a pop-up menu. Then click **Conversion**.

4. At the prompt, click **Yes** to convert the datafile to a Quick I/O file.

   If the datafile was successfully converted to a Quick I/O file, you will receive a confirmation message. Click **OK** to view the information.

Converting Quick I/O Files to Regular Datafiles

VERITAS Storage Foundation *for Oracle* provides an option to convert your Quick I/O files to regular Oracle datafiles. Use this option only if you are running Oracle9*i* and VERITAS Extension for Oracle Disk Manager.

Prerequisites

◆ Files you want to convert must be Quick I/O files on VxFS file systems or links that point to Quick I/O files.

▼ To convert Quick I/O files to regular datafiles

1. Shut down the database from the GUI.

2. Click a Quick I/O file in the object tree. (You may need to expand the tree view to find the icon.)

3. Select one of the following methods to generate datafile statistics.

   ◆ From the menu bar, select **Datafiles** > **Conversion**.

   or

   ◆ Right click the Quick I/O file to bring up a pop-up menu. Then click **Conversion**.

4. At the prompt, click **Yes** to convert the Quick I/O files to a regular datafile.

   If the Quick I/O file was successfully converted to a regular datafile, you will receive a confirmation message. Click **OK** to view the information.

Updating Rescan Intervals

VERITAS Storage Foundation *for Oracle* periodically scans the system for updated information. You can modify the rescan intervals to be faster or slower. A partial scan is a scan of existing known objects and a full scan is a scan of known and unknown objects, meaning that a search for new information is performed.

▼ To update a rescan interval

1. Click the Oracle Databases icon in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to update the rescan interval times.

   - From the menu bar, select **Oracle** > **Update Rescan Intervals**.

     or

   - Right click the Oracle Databases icon to bring up a pop-up menu. Then click **Update Rescan Intervals**.

3. To update a partial rescan interval, enter a new time (in minutes) in the appropriate field.

4. To update a full rescan interval, enter a new time (in minutes) in the appropriate field.

5. Click **Save** to save your changes. A confirmation dialog is displayed. Click **OK** to continue.

Checking the System Configuration

You can check the System Configuration only when using VERITAS Storage Foundation *for Oracle* Enterprise Edition.

▼ To scan the system configuration of a database

1. Click the Oracle database in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to view the database configuration information.

   - From the menu bar, select **Oracle** > **Check System Configuration**.

     or

   - Right click the database to bring up a pop-up menu. Then click **Check System Configuration**.

3. When the system has scanned the database, it will display a confirmation message. Click **Yes** or **No** to continue. The Scan Configuration Output window then displays. You should see output similar to the following:

```
Examining file system attributes.

NOTICE: All file systems are VxFS.
NOTICE: All file systems are VxFS Version 4 layout.

Examining Quick I/O settings.

NOTICE: All datafiles are Quick I/O files.
NOTICE: It appears that your system is ODM enabled.
NOTICE: All Quick I/O files should be converted to regular
files to use the ODM features.

Examining Cached Quick I/O settings.

NOTICE: No file systems have Cached Quick I/O enabled.

Examining datafiles fragmentation.

NOTICE: 0 files are fragmented.

Examining File System tunable settings.

NOTICE: Parameters for all VxFS file systems used by TEST9i.
Filesystem i/o parameters for /oracle9i
read_pref_io = 65536
read_nstream = 2
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 2
write_unit_io = 65536
pref_strength = 20
buf_breakup_size = 262144
discovered_direct_iosz = 262144
max_direct_iosz = 2097152
default_indir_size = 8192
qio_cache_enable = 0
write_throttle = 127232
max_diskq = 2097152
initial_extent_size = 8
max_seqio_extent_size = 2048
max_buf_data_size = 8192
hsm_write_prealloc = 0
```

```
Examining Oracle volume and file system layout.

NOTICE: Data for database TEST9i is contained in one volume
group.

Examining Oracle internal information.

Oracle Version is 9.0.0.0.0.

Control file /oracle9i/control1 is on file system /oracle9i.
WARNING: Control file is not mirrored using VxVM.

Control file /oracle9i/control2 is on file system /oracle9i.
WARNING: Control file is not mirrored using VxVM.

Total of 2 control files over 1 file systems.

WARNING: Control files are not spread over multiple file
systems. Spread control files over multiple file systems
for better redundancy.

Examining Oracle automatic extension of datafiles.

Total of 0 datafiles are configured to autoextend.
Total of 2 datafiles are defined to the database.

Examining Oracle log modes.

The database is running in archivelog mode.

The database is running in automatic log archiving mode.
```

4. Click **OK** to return to the main window.

Saving the System Configuration

You can save the System Configuration only when using VERITAS Storage Foundation *for Oracle* Enterprise Edition.

▼ To save the system configuration of a database

1. Click the Oracle database in the object tree. (You may need to expand the tree view to find the icon.)

2. Select one of the following methods to view the database configuration information.

- From the menu bar, select **Oracle** > **Save System Configuration**.

  or

- Right click the database to bring up a pop-up menu. Then click **Save System Configuration**.

The System Configuration wizard is then displayed.

3. Enter a path name, or directory, in the Path field to indicate where you would like to save the system configuration information.

4. Click **Save** to save the configuration information. A confirmation dialog is displayed. Click **OK** to continue.

Using the VxDBA Utility 13

This chapter describes how to use the VxDBA utility to support administrative tasks for Oracle database management. This is the last release of Storage Foundation *for Oracle* to support the VxDBA utility.

Topics include:

- "Overview of the VxDBA Menus" on page 334
- "Starting VxDBA" on page 338
- "Using VxDBA to Perform Administrative Operations" on page 340
- "Setting Up VxDBA in an HA Environment" on page 405

Overview of the VxDBA Menus

VxDBA Main Menu

The VxDBA utility main menu provides access to the following operations:

```
VERITAS Storage Foundation for Oracle(ORACLE_SID 'PROD')
Menu: Database Main

  1      Database Administration
  2      Display Database/VxDBA Information
  3      Storage Checkpoint Administration
  4      Storage Rollback Administration
  5      Monitoring Agent Administration
  6      Storage Checkpoint Capacity Planning

  ?      Display Help About the Current Menu
  q      Exit From Current Menu
  x      Exit From VxDBA Utility

  Select Operation to Perform:
```

Type the number, letter, or symbol of the operation you want to perform. You can also use the interrupt key (Ctrl-C) to end or break out of any operation and return to the system prompt.

The VxDBA main menu provides access to the following operations:

- **1 - Database Administration**

  Use this menu to perform basic database management operations.

  The **Database Administration** menu provides access to the following operations:

  - **Startup Database Instance**
  - **Shutdown Database Instance**
  - **Display/Update Tablespace Information**

- **2 - Display Database/VxDBA Information**

  Use this menu to display information about various aspects of your database environment, as well as examine and save configuration information for database recovery.

  The **Display Database/VxDBA Information** menu provides access to the following operations:

  - **Display Database Information**
  - **Display/Update Tablespace Information**
  - **Display Datafile /File System Information**
  - **Display VxDBA/Database Configuration Files**
  - **Examine Volume/File System/Database Configuration**
  - **Save Volume/File System/Database Configuration**

- **3 - Storage Checkpoint Administration**

  Use this menu to create, display, mount, unmount, and remove Storage Checkpoints. A Storage Checkpoint can be considered an online database backup that contains a snapshot image of the database when the Storage Checkpoint was created. You can then use these Storage Checkpoints to roll back the image of a database, a tablespace, or a datafile to some earlier state.

  The **Storage Checkpoint Administration** menu provides access to the following operations:

  - **Create New Storage Checkpoints**
  - **Display Storage Checkpoints**
  - **Mount Storage Checkpoints**
  - **Unmount Storage Checkpoints**
  - **Remove Storage Checkpoints**

- **4 - Storage Rollback Administration**

  Use this menu to roll back a database, a tablespace, a datafile, or a list of datafiles to a Storage Checkpoint and configure rollback options. You must shut down the instance to perform full Storage Rollback of the database, or you can elect to leave the database up for file or tablespace rollback. In this case, VxDBA will check if the target database objects are offline before proceeding.

The **Storage Rollback Administration** menu provides access to the following operations:

  - **Roll Back the Database to a Storage Checkpoint**
  - **Roll Back a Tablespace to a Storage Checkpoint**
  - **Roll Back Files to a Storage Checkpoint**
  - **Set Number of Storage Rollback Threads**
  - **Set Buffer Size for Storage Rollback**
  - **Show Backup Control File List**

- **5 - Monitoring Agent Administration**

  Use this menu to monitor and manage key aspects of your database environment. The primary function of VxDBA's Monitoring Agent is to monitor space usage of your database file systems, tablespaces, and datafiles. The Monitoring Agent can be configured to send alarm notifications and automatically grow file systems when space usage exceeds user-defined thresholds.

  The **Monitoring Agent Administration** menu provides access to the following operations:

  - **File System Space Administration**
  - **Oracle Tablespace/Datafile Space Administration**
  - **Configure Monitoring Agent Options**
  - **Configure Statistics Collection**
  - **Start/Stop Monitoring Agent**

- **6 - Storage Checkpoint Capacity Planning**

  Use this menu to plan the file system space requirements for Storage Checkpoints.

  The **Storage Checkpoint Capacity Planning** menu provides access to the following operations:

  - **Create Capacity Planning Schedules**
  - **Display Capacity Planning Schedules**
  - **Display Space Usage Information**
  - **Remove Capacity Planning Schedules**

Each VxDBA menu has the following standard operational and navigational controls:

- ? - **Display Help About the Current Menu**

  This menu option provides online help for the current VxDBA menu, listing the available operations and a definition of each.

- q - **Exit From Current Menu**

  This menu option returns you to the main menu if you are in one of the administration submenus or exits VxDBA if you are at the main menu level.

- x - **Exit From VxDBA Utility**

  This menu option exits the VxDBA utility.

VxDBA Submenu Operations

Most of the operations available from the VxDBA submenus are run as the Oracle database administrative user (typically, user ID `oracle`), which allows the VxDBA utility permission to connect directly to the database and gather information from the system catalog. This information includes the status of the database (for example, ACTIVE or INSTANCE DOWN), the number of tablespaces and datafiles in the current Oracle instance, and the number of file systems that these datafiles are spread across. When available, VxDBA displays these fields as header information on the submenus. For example:

```
Database Status : ONLINE
# File Systems  : (1)
# Tablespaces   : (3)
# Datafiles     : (4)
```

A subset of the VxDBA submenu operations requires superuser (`root`) privileges to interact with the more secure or "privileged" operations of the VERITAS Storage Foundation *for Oracle* product (for example, the VxDBA Monitoring Agent). When you run VxDBA operations as `root`, VxDBA cannot connect to and obtain information directly from the database, so the submenu **Database Status** header reports a permission error. For example:

```
Database Status : ORA-01031: insufficient privileges
# File Systems  :  (1)
# Tablespaces   :  (3)
# Datafiles     :  (4)
```

Because VxDBA cannot obtain information directly from the database at this time, the values for the number of file systems, tablespaces, and datafiles are enclosed in parentheses to indicate that this was the last value VxDBA was able to obtain from the system catalog.

Starting VxDBA

Most VxDBA utility operations can be run by the Oracle Database Administrator (typically, the user ID `oracle`) of the database instance. Some VxDBA utility operations, like many of the file system space management operations, require superuser (`root`) privileges. VxDBA prompts you for the `root` password when required.

Prerequisites

Before running VxDBA, you must:

- If you are using `SPFile`, run the `dbed_update` command. If you do not run the command, an error message displays.
- If you are using `$ORA_PFILE` instead of `SPFile`, you must specify the `init$ORACLE_SID.ora` parameter file in the environment variable `$ORA_PFILE` if the parameter file (`init$ORACLE_SID.ora`) is not located under `$ORACLE_HOME/dbs`.
- Have appropriate permission to run the utility. VxDBA requires permission changes to allow database administrators to access these tools. The default settings for the `/opt/VRTSdbed` directory at installation time allows only superuser (`root`) access to the directory. If you did not make these permission changes when prompted during installation, you can grant administrators access to VxDBA now.

▼ To grant administrative access to VxDBA

1. Use the `chown` and `chmod` commands to allow single user access to VxDBA. For example:

```
# chown oracle /opt/VRTSdbed
# chmod 500 /opt/VRTSdbed
```

2. Use the `chgrp` and `chmod` commands to allow group user access to VxDBA. For example:

```
# chgrp dba /opt/VRTSdbed
# chmod 550 /opt/VRTSdbed
```

▼ To start VxDBA

1. Before starting VxDBA, run the command `dbed_update` at least once to create or update the VxDBA repository.

2. To start VxDBA, enter the following command at the administrative prompt:

   ```
   $ /opt/VRTSdbed/bin/dbed_vxdba
   ```

VxDBA starts up and displays the main menu containing the available operations.

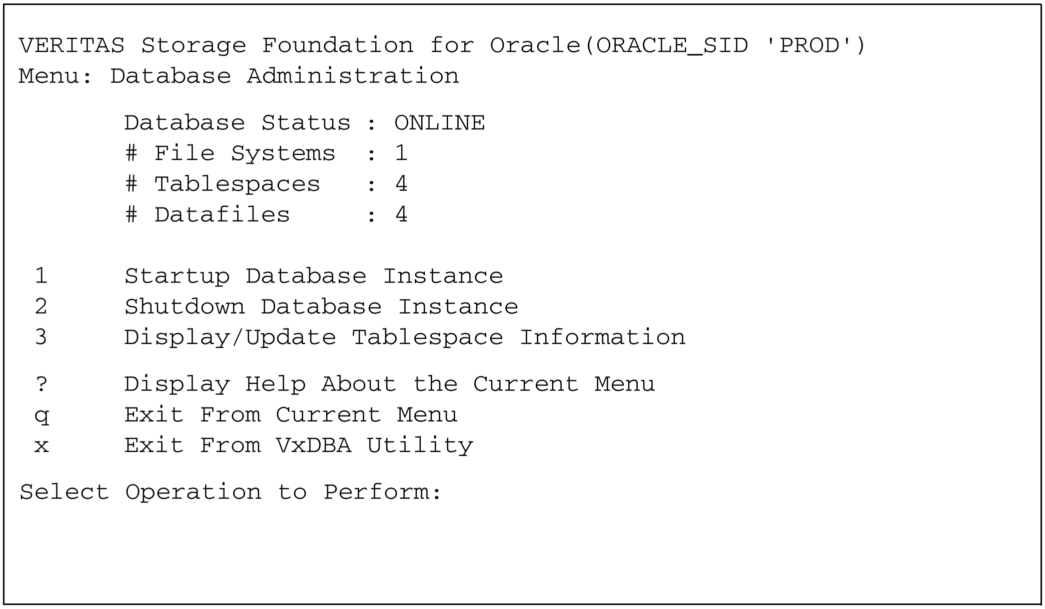
Using VxDBA to Perform Administrative Operations

The rest of this chapter details the administrative operations available through the VxDBA utility.

Managing Your Database

Use the **Database Administration** menu to perform basic database operations.

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle(ORACLE_SID 'PROD')
Menu: Database Administration

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

 1     Startup Database Instance
 2     Shutdown Database Instance
 3     Display/Update Tablespace Information

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Select from the following operations:

Startup Database Instance–Use this menu option to start up the database.

Shutdown Database Instance–Use this menu option to shut down the database.

Display/Update Tablespace Information–Use this menu option to update the tablespaces of an Oracle instance and their associated datafiles.

Starting Up a Database Instance

The VERITAS Storage Foundation *for Oracle* software package includes adaptable scripts that are run automatically when starting the database using the VxDBA utility. You can modify these scripts to run other tools and applications or to start and stop other services before and/or after database startup. For example, you could modify the scripts to ensure the database TCP listener is also correctly started.

The following scripts are included in the `/etc/vx/vxdba/$ORACLE_SID` directory:

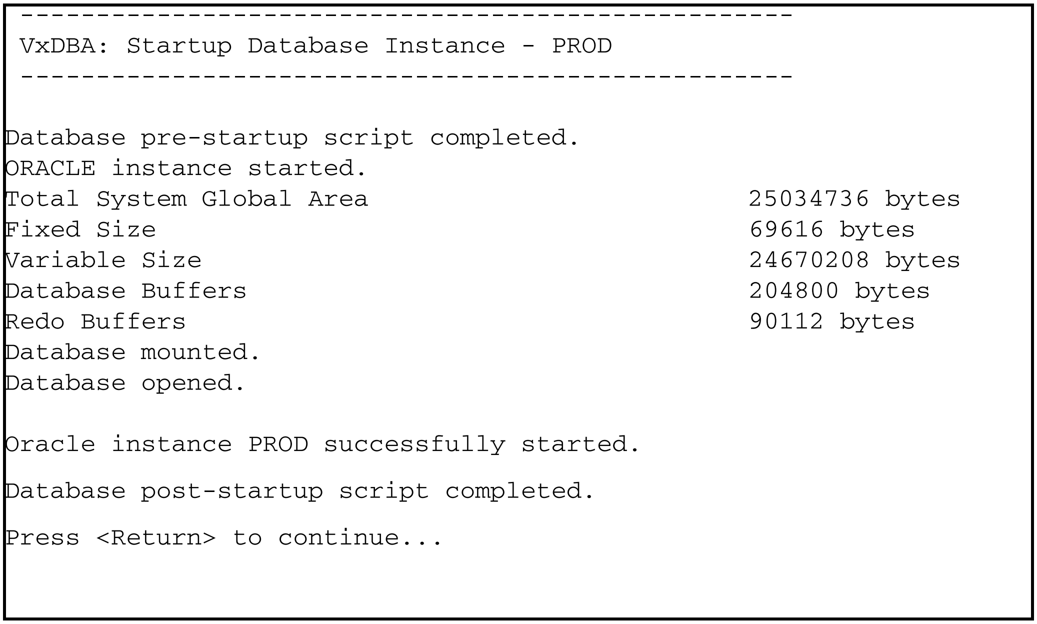
- startup.pre
- startup.post

These scripts are copied from the system-wide default files, located in `/opt/VRTSdbed/lib` directory:

- startup.pre.base
- startup.post.base

Use the **Startup Database Instance** menu option to bring the Oracle instance online. You must define the `$ORACLE_SID` and `$ORACLE_HOME` environment variables before attempting to start up the database instance.

This operation displays a screen similar to the following:



```
 ---------------------------------------------------
 VxDBA: Startup Database Instance - PROD
 ---------------------------------------------------

Database pre-startup script completed.
ORACLE instance started.
Total System Global Area                         25034736 bytes
Fixed Size                                       69616 bytes
Variable Size                                    24670208 bytes
Database Buffers                                 204800 bytes
Redo Buffers                                     90112 bytes
Database mounted.
Database opened.

Oracle instance PROD successfully started.

Database post-startup script completed.

Press <Return> to continue...
```

Shutting Down a Database Instance

The VERITAS Storage Foundation *for Oracle* software package includes adaptable scripts that are automatically run when stopping the database using the VxDBA utility. You can modify these scripts to run other tools and applications or to start and stop other services before and/or after database shutdown. For example, you could modify the scripts to ensure the database TCP listener is also correctly stopped. The following scripts are included in the `/etc/vx/vxdba/ORACLE_SID` directory:

- shutdown.pre
- shutdown.post

These scripts are copied from the system-wide default files, located in the `/opt/VRTSdbed/lib` directory:

- shutdown.pre.base
- shutdown.post.base

To perform some administrative tasks (for example, Storage Rollback), the database must be offline. Use the **Shutdown Database Instance** menu option to bring the database instance down.

This operation displays a screen similar to the following:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Shutdown Database Instance

  1       Shutdown NORMAL
  2       Shutdown TRANSACTIONAL
  3       Shutdown IMMEDIATE
  4       Shutdown ABORT

  ?       Display Help About the Current Menu
  q       Exit From Current Menu
  x       Exit From VxDBA Utility

Select Operation to Perform: 1
```

Select from the following shutdown methods:

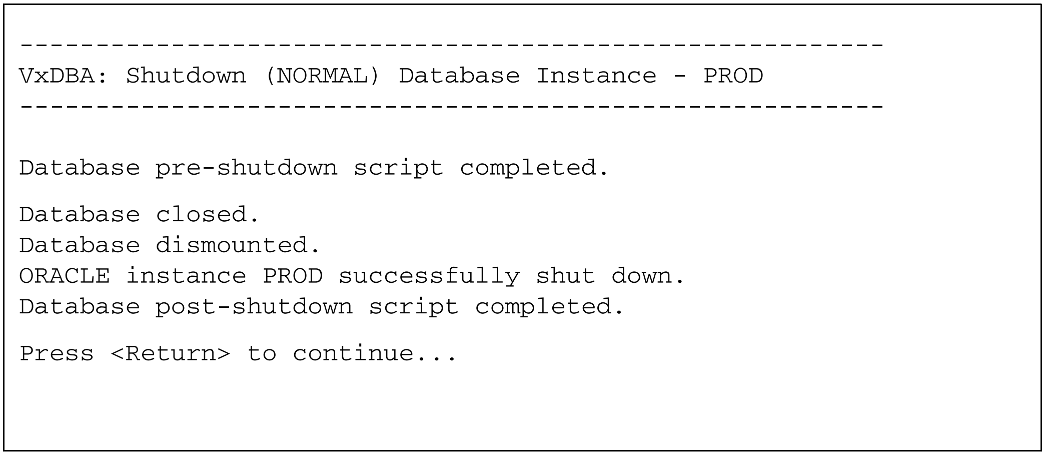
Shutdown NORMAL. Use this menu option to shut down the Oracle instance in normal situations. When this operation is selected, no new database connections are allowed. Oracle waits for all currently connected users to disconnect from the database, and then closes and dismounts the database before shutting down the instance. The next database startup does not require an instance recovery.

Shutdown TRANSACTIONAL. Use this menu option to shut down the Oracle instance immediately upon completion of all transactions in progress. When this operation is selected, no client can start a new transaction on this instance, and a client is disconnected when the transaction in progress ends. The next database startup does not require an instance recovery.

Shutdown IMMEDIATE. Use this menu option to shut down the Oracle instance immediately. Use this operation in situations where the database or some application is running irregularly or a power shutdown is about to occur. When this operation is selected, all current client SQL statements are terminated immediately, any uncommitted transactions are rolled back, and all connected users are disconnected. Oracle closes and dismounts the database before shutting down the instance. The next database startup does not require an instance recovery.

Shutdown ABORT. Use this menu option to shut down the Oracle instance instantaneously by aborting the database’s instance. Use this operation with extreme caution and only when normal shutdown or immediate shutdown does not work, you experience problems when starting the instance, or you need to shut down the instance instantaneously. When this operation is selected, all connected users are disconnected, current client SQL statements are terminated immediately, uncommitted transactions are not rolled back, and the instance is terminated without closing the files. The next database startup requires instance recovery.

The **Shutdown NORMAL** operation displays a screen similar to the following:



```
---------------------------------------------------------
VxDBA: Shutdown (NORMAL) Database Instance - PROD
---------------------------------------------------------


Database pre-shutdown script completed.

Database closed.
Database dismounted.
ORACLE instance PROD successfully shut down.
Database post-shutdown script completed.

Press <Return> to continue...
```

After the Oracle instance is shut down, VxDBA displays the **Database Administration** menu.

Displaying Database/VxDBA Information

Use the **Display Database/VxDBA Information** menu option to display information about various aspects of your database environment.

This operation displays a screen similar to the following:

```
VERITAS Storage Foundation for Oracle(ORACLE_SID 'PROD')
Menu: Display Database/VxDBA Information

 1     Display Database Information
 2     Display/Update Tablespace Information
 3     Display Datafile/File System Information
 4     Display VxDBA/Database Configuration Files
 5     Examine Volume/File System/Database Configuration
 6     Save Volume/File System/Database Configuration

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Select from the following operations:

Display Database Information. Use this menu option to display the database instance and status information.

Display/Update Tablespace Information. Use this menu option to display and update the tablespaces of Oracle instance and their associated datafiles.

Display Datafile/File System Information. Use this menu option to display the list of database files and the file systems used by the Oracle instance.

Display VxDBA/Database Configuration Files. Use this menu option to display and view the contents of various VxDBA and Oracle configuration files.

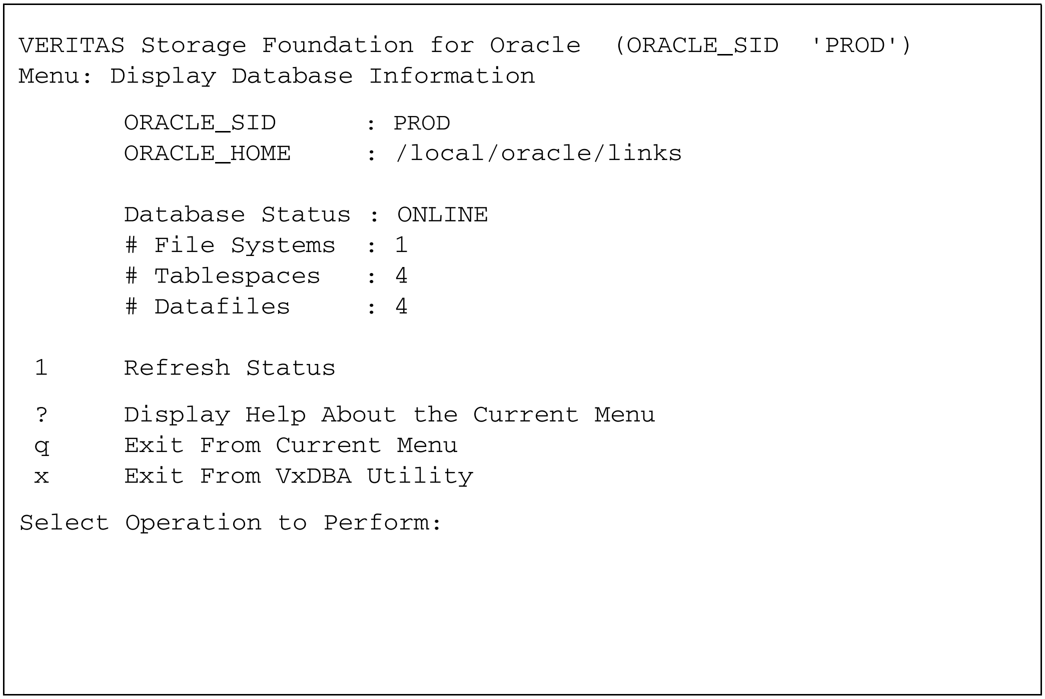
Examine Volume/File System/Database Configuration. Use this menu option to display general system configuration and database information, such as Quick I/O files, groups, control files, datafiles, and the layout and version of the file systems.

Save Volume/File System/Database Configuration. Use this menu option to save important system hardware, operating system, kernel tunables, database layout and control files, volume and file system configuration, packaging, and license information.

Displaying Database Information

Use the **Display Database Information** menu option to display the Oracle instance name (`$ORACLE_SID`), the Oracle home directory path (`$ORACLE_HOME`), and the status of the Oracle instance.

This operation displays a screen similar to the following:



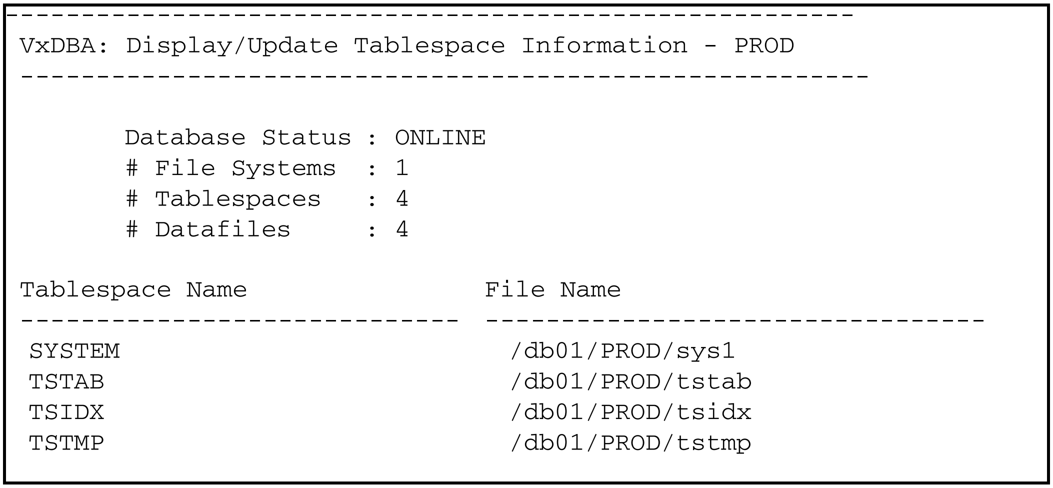
```
VERITAS Storage Foundation for Oracle  (ORACLE_SID  'PROD')
Menu: Display Database Information

       ORACLE_SID      : PROD
       ORACLE_HOME     : /local/oracle/links

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

 1     Refresh Status

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

If the display screen has been idle for a while, the status of the Oracle instance may have changed. Use the **Refresh Status** option to refresh the display with the up-to-date status of the Oracle instance.

Displaying and Updating Tablespace Information

Use the **Display/Update Tablespace Information** menu option to display the list of tablespaces of an Oracle instance and their associated datafiles. This menu option is also available on the **Database Administration** menu.

This operation displays a screen similar to the following:



```
--------------------------------------------------------
VxDBA: Display/Update Tablespace Information - PROD
--------------------------------------------------------


       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

Tablespace Name              File Name
-----------------------------  ---------------------------------
 SYSTEM                          /db01/PROD/sys1
 TSTAB                           /db01/PROD/tstab
 TSIDX                           /db01/PROD/tsidx
 TSTMP                           /db01/PROD/tstmp
```

VxDBA maintains a repository that stores the pertinent information needed to display configuration information. This repository is located at `/etc/vx/vxdba/$ORACLE_SID`.

When the database configuration changes, the information stored in the repository may not be up-to-date. When VxDBA detects that the repository is no longer consistent with Oracle's system catalog, VxDBA asks you if you want to update its repository with the latest configuration information.

Displaying Datafile and File System Information

Use the **Display Datafile/File System Information** menu option to display the list of database files and the file systems used by the Oracle instance.

This operation displays a screen similar to the following:

```
-----------------------------------------------------------
VxDBA: Display Datafile/File System Information - PROD
-----------------------------------------------------------

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

File                    File          File
System                  Type          Name
----------------------- ------------  --------------------------
/db01                   QIO       /db01/PROD/tstmp->.tstmp
/db01                   QIO       /db01/PROD/tsidx->.tsidx
/db01                   QIO       /db01/PROD/tstab->.tstab
/db01                   QIO       /db01/PROD/sys1->.sys1

Press <Return> to continue...
```

The datafile and file system information is also stored in VxDBA’s repository. When the configuration changes, VxDBA updates its repository with the latest configuration information.

Displaying VxDBA and Database Configuration Files

Use the **Display VxDBA/Database Configuration Files** menu option to display and then view the contents of various VxDBA and Oracle configuration or script files.

This operation displays a screen similar to the following:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Display VxDBA/Database Configuration Files

 1       Database Pre-Startup Script
 2       Database Post-Startup Script
 3       Database Pre-Shutdown Script
 4       Database Post-Shutdown Script
 5       VxDBA Settings File
 6       VxDBA Monitoring Agent Configuration File
 7       Oratab File (missing)
 8       Oracle Listener
 9       Oracle Configuration File

 ?       Display Help About the Current Menu
 q       Exit From Current Menu
 x       Exit From VxDBA Utility

Select Operation to Perform:
```

When you select an item from the list, VxDBA first displays the location of the file and then its contents. Use the space bar to page through the file contents. If VxDBA cannot find a configuration file, you will see a `(missing)` designation after the selection option description.

Note You cannot edit the configuration files from within VxDBA; you can only view their contents.

Examining Database Environment Information

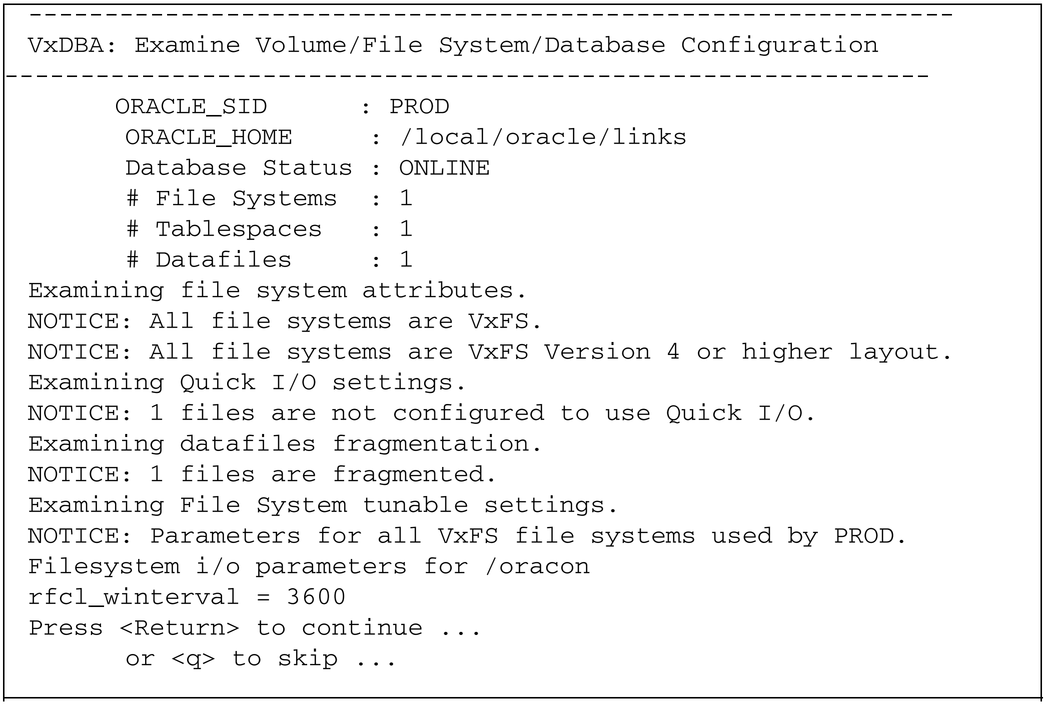
Use the **Examine Volume/File System/Database Configuration** menu option to display system configuration and database information that can assist you in determining if your database is configured properly on top of VERITAS products and the operating system.

Note The database must be running in archivelog mode.

By default, only users with superuser (`root`) privileges have permission to run `vxtunefs` and `vxdisk` commands. To allow database administrators to use these commands, you need to change these commands as follows:

```
# chown root:dba /opt/VRTSvxfs/sbin/vxtunefs
# chmod 4550 /opt/VRTSvxfs/sbin/vxtunefs
# chown root:dba /usr/sbin/vxdisk
# chmod 4550 /usr/sbin/vxdisk
```

This operation displays a screen similar to the following:



```
 -------------------------------------------------------------
 VxDBA: Examine Volume/File System/Database Configuration
-------------------------------------------------------------
       ORACLE_SID       : PROD
       ORACLE_HOME      : /local/oracle/links
       Database Status  : ONLINE
       # File Systems   : 1
       # Tablespaces    : 1
       # Datafiles      : 1
 Examining file system attributes.
 NOTICE: All file systems are VxFS.
 NOTICE: All file systems are VxFS Version 4 or higher layout.
 Examining Quick I/O settings.
 NOTICE: 1 files are not configured to use Quick I/O.
 Examining datafiles fragmentation.
 NOTICE: 1 files are fragmented.
 Examining File System tunable settings.
 NOTICE: Parameters for all VxFS file systems used by PROD.
 Filesystem i/o parameters for /oracon
 rfcl_winterval = 3600
 Press <Return> to continue ...
       or <q> to skip ...
```

Saving Database Environment Information

Use the **Save Volume/File System/Database Configuration** menu option to save important information, such as:

- System hardware
- Operating system
- Kernel tunables
- Database layout and control files
- Volume and file system configuration
- Packaging
- License information

This option creates a collection of files under a directory you specified. Copy these files to off-host, online storage to assist you in a disaster recovery situation.

Note Depending on the configuration and size of your system, this menu option can create sizable files. Make sure you have ample space on your system before selecting this option.

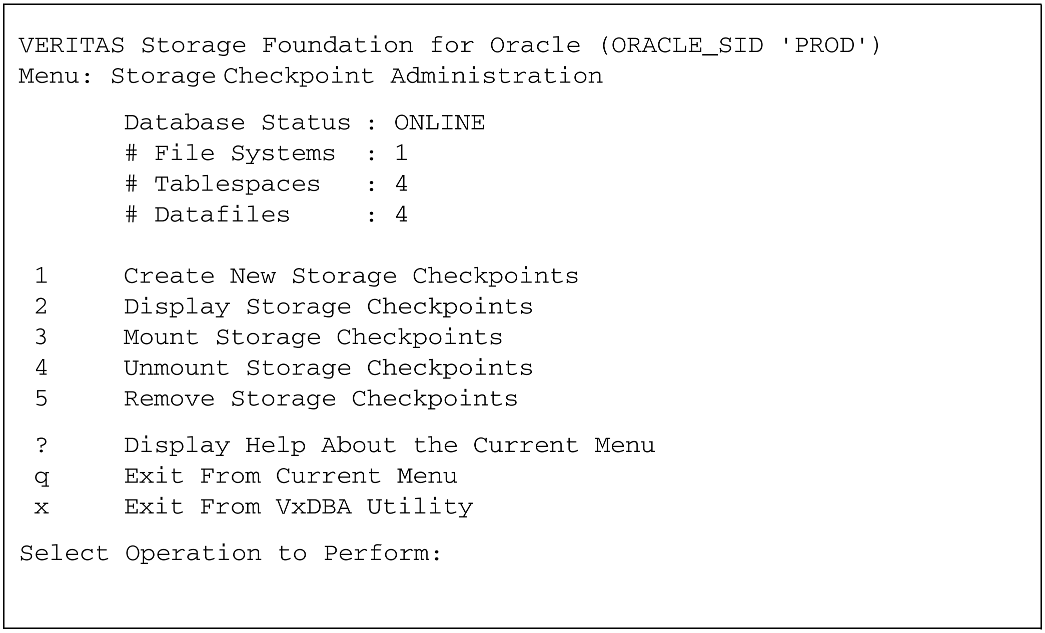
This operation displays a screen similar to the following:

```
-------------------------------------------------------------
VxDBA: Save Volume/File System/Database Configuration
-------------------------------------------------------------

Enter the path to save the configuration files: /tmp

... Saving Solaris Information ...
... Saving Oracle Information ...
... Saving Overall VERITAS Information ...
... Saving VxFS Information ...
... Saving VxVM Information ...

System configuration information saved to directory:
/tmp/vxdba.DR.6207

Press <Return> to continue...
```

Managing Storage Checkpoints

Use the **Storage Checkpoint Administration** menu option to create, display, mount, unmount, and remove Storage Checkpoints. A Storage Checkpoint can be considered an online database backup that contains a point-in-time database image. Storage Checkpoints can later be used to restore the image of a file, a tablespace, or the entire database to any earlier state recorded by the Storage Checkpoints.

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Administration

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

 1     Create New Storage Checkpoints
 2     Display Storage Checkpoints
 3     Mount Storage Checkpoints
 4     Unmount Storage Checkpoints
 5     Remove Storage Checkpoints

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

VxDBA uses the repository to determine the list of tablespaces, datafiles, and file systems for Storage Checkpoint creation and removal. Select from the following Storage Checkpoint Administration operations:

Create New Storage Checkpoints. Use this menu option to create a Storage Checkpoint as a point-in-time image of the database and to back up control file, initialization file, and log information. The database can be offline or online when you create a Storage Checkpoint.

Display Storage Checkpoints. Use this menu option to display the Storage Checkpoints created by VxDBA. After displaying the VxDBA Storage Checkpoints, VxDBA asks if you want to display any other Storage Checkpoints, such as Storage Checkpoints created by the Capacity Planning Utility or NetBackup.

Mount Storage Checkpoints. Use this menu option to mount a Storage Checkpoint into the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine, and you can access mounted Storage Checkpoints using all normal file system-based commands.

Unmount Storage Checkpoints. Use this menu option to unmount a mounted Storage Checkpoint.

Remove Storage Checkpoints. Use this menu option to remove Storage Checkpoints that you no longer need.

Creating Storage Checkpoints

This operation creates a Storage Checkpoint (point-in-time image) of the database instance. A Storage Checkpoint for a database is a collection of Storage Checkpoints created at the same time on each of the file systems where the database files reside.

It is recommended that you take a Storage Checkpoint after you convert to or from Quick I/O files.

Note Enable `ARCHIVELOG` mode before taking Storage Checkpoints. See "Creating Storage Checkpoints" on page 353 and "Using Storage Checkpoints and Storage Rollback for Backup and Restore" on page 156.

The database can be offline or online when a Storage Checkpoint is created. If the database is online when the Storage Checkpoint is created, VxDBA switches the database to online backup mode before creating the Storage Checkpoint. Once the Storage Checkpoint is created, VxDBA switches the database back to its normal operation mode.

In addition to creating a Storage Checkpoint, VxDBA also automatically backs up the associated control files, initialization file, and log information. Suppose that you made a structural change to your database, and then needed to roll back the database to a Storage Checkpoint that was created before the structural change. The Storage Rollback would only be successful if you could also reconstruct the database to the same structure that it was when the Storage Checkpoint was created. You can recreate the previous database structure using the control files, initialization file, and log information that were backed up when the Storage Checkpoint was created.

Note VxDBA does not automatically roll back the control file associated with a Storage Checkpoint. See "Rolling Back the Database to a Storage Checkpoint" on page 364 for more information on rolling back control files and "Guidelines for Oracle Recovery" on page 179 for information on Oracle recovery.

Create New Storage Checkpoints displays a screen similar to the following:

```
 ----------------------------------------------------
 VxDBA: Create New Storage Checkpoints - PROD
 ----------------------------------------------------

 NOTICE: To correctly create a Storage Checkpoint, you must
have up-to-date tablespace information stored in VxDBA's database.

 Are you certain the tablespace information is up-to-date? [y,n] y

 Do you want to create a new Storage Checkpoint?
 [y,n,q,?] (default: y) y

 Do you want to create an instant Storage Checkpoint?
 [y,n,q,?] (default: n) n

 By default, VxFS removes Storage Checkpoints when a file
 system runs out of space. While this removal allows normal file
 system operations to continue, Storage Checkpoint removal inhibits
 your ability to recover a database using Storage  Rollback. You can,
 however, direct VxFS to retain Storage  Checkpoints even in an
 out-of-space condition.

 Do you want this Storage Checkpoint to be retained if the
 file systems run out of space? [y,n,q,?] (default: n) n

 NOTICE: Oracle instance PROD is still running.

 A Storage Checkpoint can be created using online database
 backup mode (using alter tablespace ... begin backup).

 Do you want to alter tablespaces for taking the Storage
 Checkpoint? [y,n,q,?] (default: n) y

 Running 'alter tablespace' to begin online backup mode.

 Storage Checkpoint Checkpoint_971672042 created.

 Running 'alter tablespace' to end online backup mode.

 Running 'alter system switch logfile' to force a log switch.

 Backing up init.ora file for PROD in
 /etc/vx/vxdba/PROD/checkpoint_dir/Checkpoint_971672042

 Backing up control file and log information for PROD in
 /etc/vx/vxdba/PROD/checkpoint_dir/Checkpoint_971672042
```

```
Backing up include file and list of file system for VxDBA in
/etc/vx/vxdba/PROD/checkpoint_dir/Checkpoint_971672042

Do you want to display the instance's Storage
Checkpoints? [y,n,q,?] (default: y) y
```

Displaying Storage Checkpoints

This operation displays the list of Storage Checkpoints created by the VxDBA utility on the file systems used by the Oracle instance. VxDBA also asks if you want to display Storage Checkpoints created by other applications, such as Capacity Planning Utility or NetBackup.

While you see and manage only a single Storage Checkpoint, the Storage Checkpoint is a collection of Storage Checkpoints. Each file system used by the database contains a Storage Checkpoint of the same name, and it is this collection of Storage Checkpoints across file systems that you manage as a single Storage Checkpoint in VxDBA.

Storage Checkpoints are of the following types:

- ON–Online
- OF–Offline
- IN–Instant
- UN–Unknown

Storage Checkpoints may have additional status appended to show other attributes. Current additional status modifiers are:

- M–Mounted
- R–Read-only
- W–Writable
- I–Invalid for current Oracle SID
- C–Complete
- P–Partial

The status of a Storage Checkpoint can be 'C' for Complete or 'P' for partial. Complete means that the Storage Checkpoint successfully completed across all file systems used by the database, and each file system contains the same Storage Checkpoint name. Partial means that the Storage Checkpoint operation did not successfully complete or that one or more of the file systems used by the database does not contain the named Storage Checkpoint. Partial Storage Checkpoints can happen if, for example:

- You are in the middle of creating a new Storage Checkpoint and the database or system crashes. When the system or database is back online, VxDBA detects that not all of the file systems used by the database contain the named Storage Checkpoint.

  You should consider deleting partial Storage Checkpoints that are a result of a database or system crash, and create a new Storage Checkpoint.

- After taking a successful, complete Storage Checkpoint, you add a tablespace on a new file system or a file system that was not previously used by the database. Again, VxDBA detects that not all of the file systems used by the database contain the named Storage Checkpoint. In this case, you can use the partial Storage Checkpoint for Storage Rollback, but only datafiles on file systems that contain complete Storage Checkpoints can be rolled back.

  Be sure you understand the ramifications of rolling back to such a Storage Checkpoint (for example, losing the new tablespace). See the *Oracle Backup and Recovery Guide* for information and tips on restoring databases from an old backup.

- One of the file systems used by your database runs out of space and VxFS automatically removes the oldest Storage Checkpoint on that file system. VxDBA detects that the Storage Checkpoint on that file system is missing and marks the Storage Checkpoint as partial. Here again, you may be able to use the partial Storage Checkpoint for Storage Rollback, but do consider the ramifications of doing so.

  To avoid this situation, use VxDBA's Monitoring Agent to monitor file system space usage on all file systems used by the database and allow VxDBA to automatically grow the file systems when they are running out of space. See "Managing File System Space" on page 380 for more information.

- One or more of the file systems used by the database are not VxFS file systems and, therefore, do not support Storage Checkpoints. VxDBA detects that one or more file system Storage Checkpoints are missing and marks the Storage Checkpoint as partial.

  Avoid using mixed file systems in support of databases when possible.

Mounted means that the Storage Checkpoint is currently mounted. Writable means that the Storage Checkpoint has been modified, by `fsck` or by being mounted as read-write, and is not suitable for Storage Rollback operations.

Ordering of the displayed Storage Checkpoints is by default from most recently created to least recently created. This ordering may not be intuitive, especially if you want to keep the existing list items in the same order. If you need to modify this ordering, set the VXDBA_CKPT_SORT environment variable. The default ordering for sorting Storage

Checkpoint names is "-r" (most to least recent). By setting this variable to another sort option, the **Status** field identifies if the Storage Checkpoint is partial (P), complete (C), invalid (I), mounted (M), read-only (R), or writable (W).

Display Storage Checkpoints displays a screen similar to the following:

```
 ------------------------------------------------------
 VxDBA: Display Storage Checkpoints - PROD
 ------------------------------------------------------

 Storage Checkpoint                Creation Time              Status
 -------------------------------- -------------------------- ------
 Checkpoint_971672042              Sun Mar 9 13:55:53 2003      C+R
 Checkpoint_903937870              Fri Apr 18 22:51:10 2003     C+R
 Checkpoint_901426272              Mon May 5 16:17:52 2003      P+R
  REP_ckpt_incr_975265015            Wed Oct 9  16:00:00 2000   C+R
  REP_ckpt_incr_975258031            Sun Oct 8  16:00:00 2000   C+R


 Press <Return> to continue...

 Do you want to display any other for Storage Checkpoints (for
 example, Capacity Planning Utility or NetBackup)? [y,n,q,?]
(default: n) y

 Storage Checkpoint                    Creator   Location
 ------------------------------------  -------   ---------
 Planning_00001_971596803              PLAN      /db01

 Press <Return> to continue...
```

Mounting Storage Checkpoints

This operation lets you mount a Storage Checkpoint. You can mount, access, and write to Storage Checkpoints just as you can any file system. See “Using Storage Checkpoints and Storage Rollback” on page 155 for more information.

Mount Storage Checkpoints displays a screen similar to the following:

```
----------------------------------------------------
VxDBA: Mount Storage Checkpoints - PROD
----------------------------------------------------

 # Storage Checkpoint             Creation Time            Status
-- ----------------------------- ------------------------- ------
 1 Checkpoint_971672042          Sun Mar 9 13:55:53 2003    ON
 2 Checkpoint_903937870          Fri May 13 22:51:10 2003   C+R+ON
 3 Checkpoint_901426272          Wed May 11 16:17:52 2003   P+R+I
 4 REP_ckpt_incr_975265015       Mon Oct 9  16:00:00 2000   C+R
 5 REP_ckpt_full_975258031       Sun Oct 8  16:00:00 2000   C+R


Do you want to mount any of these Storage Checkpoints?
[y,n,q,?] (default: n) y

Enter a Storage Checkpoint number to mount: 1

You have the option to mount a Storage Checkpoint as writable.
When you mount a Storage Checkpoint as writable, VxFS retains
a backup, read-only version of the Storage Checkpoint and
creates a shadow Storage Checkpoint for the write operations.

Do you want to mount this Storage Checkpoint as writable?
[y,n,q,?] (default: n) y

By default, VxDBA adds a suffix (for example, _wr001) to the
writable Storage Checkpoint name to distinquish it from the
original, read-only Storage Checkpoint. You have the option to
include an additional identifying tag (up to 4 characters) to the
Storage Checkpoint name to help you identify the writable Storage
Checkpoint.

Do you want to add a tag to the writable Storage Checkpoint name?
[y,n,q,?] (default: n) y

Enter a tag for the Storage Checkpoint name: t1
```

```
You must supply a mount point for the Storage Checkpoint.
You can create a new directory for the mount point, or use
an existing directory. If you use an existing directory,
the directory must be empty.

Enter a mount point for the writable Storage Checkpoint: /wckpt

Mounting Storage Checkpoint Checkpoint_971672042 under /wckpt.

Creating Storage Checkpoint on /wckpt/db01 with name
Checkpoint_971672042_t1_wr001

Mount successful.

Press <Return> to continue...
```

Unmounting Storage Checkpoints

This operation lets you unmount a previously mounted Storage Checkpoint.

Unmount Storage Checkpoints displays a screen similar to the following:

```
----------------------------------------------------
VxDBA: Unmount Storage Checkpoints - PROD
----------------------------------------------------

 # Storage Checkpoint             Creation Time              Status
-- ------------------------------ -------------------------- ------
 1 Checkpoint_971672042_t1_wr001  Sun Mar 9 13:55:53 2003    C+W+ON
 2 Checkpoint_971672042           Fri May 13 22:51:10 2002   C+R+ON
 3 Checkpoint_901426272           Wed May 11 16:17:52 2002   P+R+I
 4 REP_ckpt_incr_975265015        Mon Oct 9  16:00:00 2000   C+R
 5 REP_ckpt_full_975258031        Sun Oct 8  16:00:00 2000   C+R


Do you want to unmount any of these Storage Checkpoints?
[y,n,q,?] (default: n) y

Enter a Storage Checkpoint number to unmount: 1

Unmounting Storage Checkpoint Checkpoint_971672042_t1_wr001
from /wckpt.

Unmount successful.

Press <Return> to continue...
```

Removing Storage Checkpoints

This operation removes Storage Checkpoints that are no longer needed. For example, you can remove a Storage Checkpoint on a file system to free up needed space.

Remove Storage Checkpoints displays a screen similar to the following:

```
-----------------------------------------------------
VxDBA: Remove Storage Checkpoints - PROD
-----------------------------------------------------

 # Storage Checkpoint              Creation Time              Status
-- ------------------------------ -------------------------- ------
 1 Checkpoint_971672042_t1_wr001  Sun Mar 9 13:55:53 2003    C+W+ON
 2 Checkpoint_971672042           Fri May 13 22:51:10 2003   C+R+ON
 3 Checkpoint_901426272           Wed May 11 16:17:52 2003   P+R+I
 4 REP_ckpt_incr_975265015        Mon Oct 9  16:00:00 2000   C+R
 5 REP_ckpt_full_975258031        Sun Oct 8  16:00:00 2000   C+R


Do you want to remove any of these Storage Checkpoints?
[y,n,q,?] (default: n) y

Enter a Storage Checkpoint number or a range of numbers to remove.
You can also enter 'all' to remove the entire list of Storage
Checkpoints.  [<number>,<number>-<number>,all,q,?] 1

Do you want to remove Checkpoint_971672042_t1_wr001? [y,n,q,?]
(default: n) y

Storage Checkpoint Checkpoint_971672042_t1_wr001
removed.

Backup control file, init.ora file, and log information removed.
Updating VxDBA database...
Done.

Press <Return> to continue...


Do you want to remove any other Storage Checkpoints
(for example, Storage Checkpoints created by the Capacity
Planning Utility or NetBackup)? [y,n,q,?]  (default: n) n
```

Note When you remove Storage Checkpoints created by NetBackup, remember to restart the backup schedule. Also remember that NetBackup needs two Storage Checkpoints to perform an incremental backup. If you remove Storage Checkpoints needed for incremental backups, NetBackup will perform a full backup instead of an incremental backup.

Managing Storage Rollback

Use the **Storage Rollback Administration** menu to roll back a database file, a list of database files, a single tablespace, or the entire database to a Storage Checkpoint.

Note You must be the Oracle Database Administrator to perform Storage Rollback operations. You must shut down the instance to perform full Storage Rollback of the database, or you can elect to leave the database up to roll back a file or tablespace. In this situation, VxDBA checks to see if the target database objects are offline before proceeding. See "Backing Up and Recovering the Database Using Storage Checkpoints" on page 169 and "Guidelines for Oracle Recovery" on page 179 for more information.

This operation displays a screen similar to the following:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Rollback Administration

       Database Status : OFFLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

 1     Roll Back the Database to a Storage Checkpoint
 2     Roll Back a Tablespace to a Storage Checkpoint
 3     Roll Back Files to a Storage Checkpoint
 4     Set Number of Storage Rollback Threads
 5     Set Buffer Size for Storage Rollback

 6     Show Backup Control File List

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Storage Checkpoints can only be used to roll back files that are damaged due to a software error or a human error (for example, accidental deletion of a table). Because Storage Checkpoints reside on the same physical disks as the primary file system, when a file is corrupted due to a media failure, the file on the Storage Checkpoints will not be available either. In this case, you need to restore files from a tape backup.

Note Some database changes, made after a Storage Checkpoint was taken, will make it impossible to run Storage Rollback successfully. For example, you cannot successfully run Storage Rollback if the control files for the database have recorded the addition or removal of datafiles To provide recovery options, a backup copy of the control file for the database is saved under the `/etc/vx/vxdba/$ORACLE_SID/checkpoint_dir/CKPT_NAME` directory just after a Storage Checkpoint is created. You can use this file to assist with database recovery, if necessary. If possible, both an ascii and binary version of the control file will be left under the `/etc/vx/vxdba/$ORACLE_SID/checkpoint_dir/CKPT_NAME` directory, with the binary version being compressed to conserve space. Use extreme caution when recovering your database using alternate control files.

After the files are rolled back, you may need to follow the recovery procedure described in the Oracle manuals to recover the database before the database can be used.

Rolling Back the Database to a Storage Checkpoint

This operation rolls back the entire database (all the datafiles used by the database, except the redo logs and control files) to a Storage Checkpoint. You must shut down the database to roll back the database.

Note While the Storage Rollback process is running, it creates a temporary file, `/`*`filesystem`*`/.VRTSstrb.lock`, in each file system. Do not remove these temporary lock files.

Roll Back the Database to a Storage Checkpoint displays a screen similar to the following:

```
--------------------------------------------------------------
VxDBA: Roll Back the Database to a Storage Checkpoint - PROD
--------------------------------------------------------------


       ORACLE_SID      : PROD
       ORACLE_HOME     : /usr1/oracle
       Database Status : ORA-01034: ORACLE not available
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

From the following list, select a Storage Checkpoint to roll
back the database to.

 # Storage Checkpoint             Creation Time            Status
-- ------------------------------ ------------------------ ------
 1 Checkpoint_903937870           Fri May 13 22:51:10 2003   C+R+ON
 2 Checkpoint_901426272           Wed May 11 16:17:52 2003   P+R+I
 3 REP_ckpt_incr_975265015        Mon Oct 9  16:00:00 2000   C+R
 4 REP_ckpt_full_975258031        Sun Oct 8  16:00:00 2000   C+R


Enter a Storage Checkpoint number [<number>,q,?] 1

The following information may be required for Oracle
recovery procedures. Times reflect the real begin and
end times for Oracle tablespaces in hot-backup
mode.
Begin hot backup time: Fri May 13 22:51:10 PDT 2003
End hot backup time  : Fri May 13 22:51:15 PDT 2003
```

```
Do you want to roll back the database to Fri May 13 22:51:10 2003?
[y,n,q,?] (default: n) y



The database has been rolled back to a Storage Checkpoint dated
Fri Apr 11 15:13:13 2003

You must follow the recovery procedure described in the Oracle
documentation before you can use the database.

Press <Return> to continue...
```

VxDBA first displays a list of Storage Checkpoints for selection. After a Storage Checkpoint is selected, VxDBA rolls back every database file in parallel. Parallel Storage Rollback significantly reduces the time required to roll back a database.

Rolling Back a Tablespace to a Storage Checkpoint

If a tablespace is corrupted or removed due to a software error or a human mistake, this operation rolls back all of the files of the tablespace to a Storage Checkpoint.

Note Rolling back a tablespace is used for complete recovery of the tablespace. It is not designed for point-in-time (incomplete) tablespace recovery, which is more complicated and requires interaction with Oracle Customer Support. The tablespace point-in-time recovery requires using a clone database. See "Cloning the Oracle Instance Using dbed_clonedb" on page 455 for more information.

▼ If the database is offline

Roll Back a Tablespace to a Storage Checkpoint displays a screen similar to the following:

```
---------------------------------------------------------------
VxDBA: Roll Back a Tablespace to a Storage Checkpoint - PROD
---------------------------------------------------------------

       ORACLE_SID      :  PROD
       ORACLE_HOME     : /usr1/oracle

       Database Status : ORA-01034: ORACLE not available
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

Select a tablespace:

Tablespace Name                  File Name
-------------------------------- ------------------------
SYSTEM                           /db01/PROD/sys1
TSTAB                            /db01/PROD/tstab
TSIDX                            /db01/PROD/tsidx
TSTMP                            /db01/PROD/tstmp

Enter a tablespace name [<name>,q,?] TSIDX

TABLESPACE: TSIDX
#  File Name
--- ------------------------------------
1 /db01/PROD/tsidx

Do you want to roll back the files used by TSIDX to a Storage
Checkpoint? [y,n,q,?] (default: n) y
```

VxDBA prompts you for a tablespace name, and then displays a list of Storage Checkpoints:

```
From the following list, select the Storage Checkpoint, used
by the tablespace, to roll back the files to. Use the 'Roll
Back Files to a Storage Checkpoint' option if you only plan to
roll back a single file or a set of files.

 # Storage Checkpoint              Creation Time              Status
-- ------------------------------ -------------------------- ------
 1 Checkpoint_903937870           Fri May 13 22:51:10 2003   C+R+ON
 2 Checkpoint_901426272           Wed May 11 16:17:52 2003   P+R+I


Enter a Storage Checkpoint number [<number>,q,?] 1

The following information may be required for Oracle
recovery procedures. Times reflect the real begin and
end times for Oracle tablespaces being in hot-backup
mode.
Begin hot backup time: Fri May 13 22:51:10 PDT 2003
End hot backup time  : Fri May 13 22:51:15 PDT 2003

Do you want to roll back tablespace TSIDX to
Fri May 13 22:51:10 2003? [y,n,q,?] (default: n) y

Rolling back tablespace TSIDX using Checkpoint_903937870...

The tablespace TSIDX has been rolled back to
Fri May 13 22:51:10 2003

You must follow the recovery procedure described in the Oracle
documentation before you can use the database.

Press <Return> to continue...
```

After you select a Storage Checkpoint, VxDBA rolls back the files of the selected tablespace in parallel. Parallel Storage Rollback significantly reduces the time required to roll back a tablespace.

▼ If the database is online

Roll Back a Tablespace to a Storage Checkpoint displays a screen similar to the following:

```
---------------------------------------------------------------
VxDBA: Roll Back a Tablespace to a Storage Checkpoint - PROD
---------------------------------------------------------------

       ORACLE_SID      :  PROD
       ORACLE_HOME     : /usr1/oracle

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

WARNING: Oracle instance PROD is still running.

Proceed with ONLINE rollback? [y,n,q,?] (default: n) y

Select a tablespace:

Tablespace Name                   File Name
--------------------------------- ------------------------
SYSTEM                            /db01/PROD/sys1
TSTAB                             /db01/PROD/tstab
TSIDX                             /db01/PROD/tsidx
TSTMP                             /db01/PROD/tstmp

Enter a tablespace name [<name>,q,?] TSIDX

TABLESPACE: TSIDX

#  File Name
--- -------------------------------------
1 /db01/PROD/tsidx

Do you want to roll back the files used by TSIDX to a Storage
Checkpoint? [y,n,q,?] (default: n) y
```

VxDBA prompts you for a tablespace name, and then displays a list of Storage Checkpoints:

```
From the following list, select the Storage Checkpoint, used
by the tablespace, to roll back the files to. Use the 'Roll
Back Files to a Storage Checkpoint' option if you only plan to
roll back a single file or a set of files.

 # Storage Checkpoint             Creation Time              Status
-- ------------------------------ -------------------------- ------
 1 Checkpoint_903937870           Fri May 13 22:51:10 2003   C+R+ON
 2 Checkpoint_901426272           Wed May 11 16:17:52 2003   P+R+I


Enter a Storage Checkpoint number [<number>,q,?] 1

The following information may be required for Oracle
recovery procedures. Times reflect the real begin and
end times for Oracle tablespaces being in hot-backup
mode.
Begin hot backup time: Fri May 13 22:51:10 PDT 2003
End hot backup time  : Fri May 13 22:51:15 PDT 2003

Do you want to roll back tablespace TSIDX to
Fri May 13 22:51:10 2003? [y,n,q,?] (default: n) y
NOTICE: Tablespace TSIDX is currently online.

Do you want to offline tablespace TSIDX now?
[y,n,q,?] (default: y) y
Running script to offline tablespace TSIDX.

Rolling back tablespace TSIDX using Checkpoint_903937870...

The tablespace TSIDX has been rolled back to
Fri May 13 22:51:10 2003
You must follow the recovery procedure described in the Oracle
documentation before you can use the database.

Press <Return> to continue...
```

After you select a Storage Checkpoint, VxDBA rolls back the files of the selected tablespace in parallel. Parallel Storage Rollback significantly reduces the time required to roll back a tablespace.

Rolling Back Datafiles to a Storage Checkpoint

This operation rolls back database files to a Storage Checkpoint. You can also use this operation to roll back more than one tablespace. Specify the list of files for Storage Rollback in a list file, or enter the list of files one by one. If the database is offline **Roll Back Files to a Storage Checkpoint** displays a screen similar to the following:

```
---------------------------------------------------------
VxDBA: Roll Back Files to a Storage Checkpoint - PROD
---------------------------------------------------------

       ORACLE_SID      :  PROD
       ORACLE_HOME     : /usr1/oracle

       Database Status : ORA-01034: ORACLE not available
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

To roll back files to a Storage Checkpoint, enter the name of
a list file that contains the set of files for Storage Rollback or
press <Return> to enter a list of file names one by one.

Enter the list file name [<name>,<Return>,q,?] <Return>

Enter a list of file names for Storage Rollback. To end the
list, enter <Return>. <Return>

Enter file name [<name>,<Return>,q,?] /db01/PROD/tsidx

Enter file name [<name>,<Return>,q,?] <Return>

#     File Name
---   -------------------------------------
 1    /db01/PROD/tsidx

Do you want to roll back these files to a Storage Checkpoint?
[y,n,q,?] (default: n) y
```

```
From the following list, select the Storage Checkpoint to roll back
the files to.

 # Storage Checkpoint             Creation Time             Status
-- ------------------------------ ------------------------- ------
 1 Checkpoint_903937870           Fri May 13 22:51:10 2003   C+R
 2 Checkpoint_901426272           Wed May 11 16:17:52 2003   P+R


Enter a Storage Checkpoint number [<number>,q,?] 1

The following information may be required for Oracle
recovery procedures. Times reflect the real begin and
end times for Oracle tablespaces being in hot-backup
mode.
Begin hot backup time: Fri May 13 22:51:10 PDT 2003
End hot backup time  : Fri May 13 22:51:15 PDT 2003

Do you want to roll back files to
Fri May 13 22:51:10 2003? [y,n,q,?] (default: n) y

Rolling back files to Checkpoint_903937870...


The files have been rolled back to Fri May 13 22:51:10 2003.

You must follow the recovery procedure described in the
Oracle documentation before you can use the database.

Press <Return> to continue...
```

After you select a Storage Checkpoint, VxDBA rolls back the files to the selected Storage Checkpoint in parallel. Parallel Storage Rollback significantly reduces the time required to roll back files. When all the files are rolled back, follow the Oracle recovery procedure to recover the database before the database is used.

▼ If the database is online

Roll Back Files to a Storage Checkpoint displays a screen similar to the following:

```
---------------------------------------------------------
VxDBA: Roll Back Files to a Storage Checkpoint - PROD
---------------------------------------------------------

       ORACLE_SID      :  PROD
       ORACLE_HOME     : /usr1/oracle

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

WARNING: Oracle instance PRD is still running.

Proceed with ONLINE rollback? [y,n,q,?] (default: n) y

To roll back files to a Storage Checkpoint, enter the name of
a list file that contains the set of files for Storage Rollback or
press <Return> to enter a list of file names one by one.

Enter the list file name [<name>,<Return>,q,?] <Return>

Enter a list of file names for Storage Rollback. To end the
list, enter <Return>. <Return>

Enter file name [<name>,<Return>,q,?] /db01/PROD/tsidx

Enter file name [<name>,<Return>,q,?] <Return>

#    File Name
---  -------------------------------------
 1   /db01/PROD/tsidx

Do you want to roll back these files to a Storage Checkpoint?
[y,n,q,?] (default: n) y

From the following list, select the Storage Checkpoint to roll back
the files to.

 # Storage Checkpoint              Creation Time              Status
-- ------------------------------ -------------------------- ------
 1 Checkpoint_903937870           Fri May 13 22:51:10 2003   C+R+ON
 2 Checkpoint_901426272           Wed May 11 16:17:52 2003   P+R+I
 3 REP_ckpt_incr_975265015        Mon Oct 9  16:00:00 2000   C+R
 4 REP_ckpt_full_975258031        Sun Oct 8  16:00:00 2000   C+R
```

```
Enter a Storage Checkpoint number [<number>,q,?] 1

The following information may be required for Oracle
recovery procedures. Times reflect the real begin and
end times for Oracle tablespaces being in hot-backup
mode.
Begin hot backup time: Fri May 13 22:51:10 PDT 2003
End hot backup time  : Fri May 13 22:51:15 PDT 2003

Do you want to roll back files to
Fri May 13 22:51:10 2003? [y,n,q,?] (default: n) y

From the following list, select the Storage Checkpoint to roll back
the files to.

NOTICE: Datafile/db01/PROD/tsidx is currently online
and must be taken offline for online rollback.
Do you want to offline datafile (s) for rollback? [y,n,q,?]
(default: y) y
Running script to offline datafile (s).

Rolling back files to Checkpoint_903937870...

The files have been rolled back to a Storage Checkpoint dated
Fri May 13 22:51:10 2003.

You must follow the recovery procedure described in the
Oracle documentation before you can use the database.

Press <Return> to continue...
```

After you select a Storage Checkpoint, VxDBA rolls back the files to the selected Storage Checkpoint in parallel. Parallel Storage Rollback significantly reduces the time required to roll back files. When all the files are rolled back, follow the Oracle recovery procedure to recover the database before the database is used.

Setting the Number of Storage Rollback Threads

This operation lets you configure the number of threads used when rolling back an Oracle datafile. Performance is a critical factor when rolling a file back to a Storage Checkpoint. By default, 8 lightweight threads are used to partition up and recover a datafile. Depending on the number of CPUs available on your system and the type of volume on which the file system is located, this default setting may specify too few or too many threads.

Set Number of Storage Rollback Threads displays a screen similar to the following:

```
 ----------------------------------------------------------
 VxDBA: Set Number of Storage Rollback Threads for PROD
 ----------------------------------------------------------


 There are currently 8 threads set for
 Storage Rollbacks associated with database PROD.

 Do you want to change the number of parallel threads?
[y,n,q,?](default: y)

 New number of threads: 16
 Settings saved.


 Press <Return> to continue...
```

This menu option lets you experiment with various settings to achieve optimal performance for your system. You do not need to change the default number of threads if the Storage Rollback performance on your system is satisfactory. The maximum number of threads you can set is 63, and the minimum number of threads is 1.

Setting the Buffer Size for Storage Rollback

This operation lets you configure the buffer size used for Storage Rollback. As with setting the number of Storage Rollback threads, the buffer size configured for reads and writes when rolling back an Oracle datafile can also affect performance. By default, a 128K read/write buffer is used.

Set Buffer Size for Storage Rollback displays a screen similar to the following:

```
---------------------------------------------------------
VxDBA: Set Buffer Size for Storage Rollback for PROD
---------------------------------------------------------

There is currently a buffer of 131072 bytes
for Storage Rollbacks associated with database PROD.

Do you want to change the buffer size? [y,n,q,?](default: y)
New size of buffer: 1024000

There is currently a buffer of 1024000 bytes
for Storage Rollbacks associated with database PROD.

Settings saved.

Press <Return> to continue...
```

This menu option lets you experiment with various settings to gain optimal performance for your system. You do not need to change the default buffer size if the Storage Rollback performance on your system is satisfactory. Set the buffer size in bytes—the minimum buffer size is 1K (1024 bytes), and the maximum buffer size is 1 MB (1,048,576 bytes).

Showing the Backup Control File List

This operation displays the list of control files that VxDBA has backed up each time you create a Storage Checkpoint using VxDBA.

Show Backup Control File List displays a screen similar to the following:

```
-----------------------------------------------------
VxDBA: Show Backup Control File List - PROD
-----------------------------------------------------


                                                  DB         INIT&
Backup Control File                               UP?  CKPT?  CNTL?
------------------------------------------------- --  -----  -----
/etc/vx/vxdba/PROD/c*dir/Checkpoint_903937870      Y     Y       N
/etc/vx/vxdba/PROD/c*dir/Checkpoint_901426272      Y     N       Y
/etc/vx/vxdba/PROD/c*dir/REP_ckpt_incr_975265015   N     Y       Y
/etc/vx/vxdba/PROD/c*dir/REP_ckpt_full_975258031   N     Y       Y

Press <Return> to continue...
```

Following the location of the backup control file, **Show Backup Control File List** displays the following information:

DB UP? Shows the status of the Oracle instance (`Y` for `ONLINE` or `N` for `OFFLINE`) when the Storage Checkpoint was taken. When a Storage Checkpoint is created while the Oracle instance is up, the control files, initialization file, and the log information are backed up at the same time. However, if a Storage Checkpoint is created while the Oracle instance is down, the log information will not be available. If this backed up information is removed from the Storage Checkpoint directory, the status of the `INIT&CNTL?` field becomes `N` (for null).

CKPT? Shows whether the same Storage Checkpoint still exists across all file systems used by the database (see the discussion of partial Storage Checkpoints in "Displaying Storage Checkpoints" on page 355 for more information). When one or more of the file systems used by the database does not contain the named Storage Checkpoint, the status of `CKPT?` becomes `n` (for null).

INIT& CNTL. Shows whether the Oracle instance initialization file and the backup control file still exist in the Storage Checkpoint directory. If this backed up information is removed from the Storage Checkpoint directory, the status of the `INIT& CNTL?` field becomes `N` (for null).

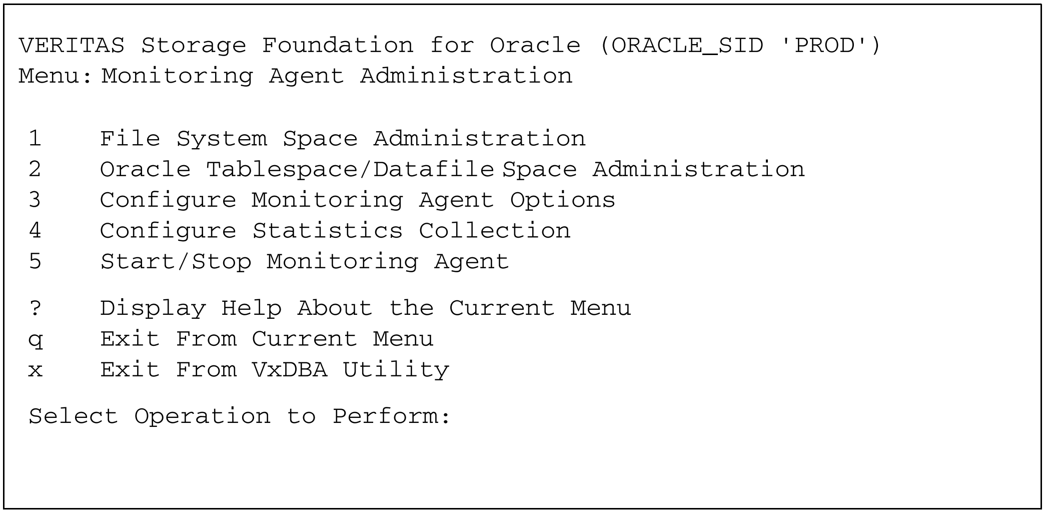
Note If you need to recover the database from structural changes, these backup control files may be required to effect the recovery. Storage Rollback does not automatically determine whether this is the case, as usage depends on the Oracle recovery that the administrator intends to perform. If it is determined that a backup control file is required for recovery, it can be copied from the directory location shown in the **Backup Control File** field of the above display.

Managing Space Usage and the VxDBA Monitoring Agent

Use the **Monitoring Agent Administration** menu to:

- Manage and monitor VxFS file system, Oracle tablespace, and datafile space usage
- Configure the Monitoring Agent options and statistics collection
- Start and stop the Monitoring Agent

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Monitoring Agent Administration

 1    File System Space Administration
 2    Oracle Tablespace/Datafile Space Administration
 3    Configure Monitoring Agent Options
 4    Configure Statistics Collection
 5    Start/Stop Monitoring Agent

 ?    Display Help About the Current Menu
 q    Exit From Current Menu
 x    Exit From VxDBA Utility

 Select Operation to Perform:
```

Managing File System Space

The VxDBA Monitoring Agent monitors the file system space, and when the space usage reaches a configured threshold value, a predefined action script grows the file system automatically. The agent can be enabled or disabled to start at boot-time. Each file system monitored has three settings that the Monitoring Agent needs to know about:

- *Warning Threshold* is a percent value (% of file system utilized) that determines when the agent begins warning the administrator of space shortage
- *Grow Threshold* is a percent value (% of file system utilized) that determines when the agent is to attempt to grow the file system (when space usage is at a critical level)
- *Amount* is either a percentage or a value in megabytes by which to grow file systems when the Grow Threshold is reached or exceeded

The VxDBA Monitoring Agent operations are driven from the following files:

- `/opt/VRTSdbed/lib/dbed_mon_config.base`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_config.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_fslist.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_oralist.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/include`

The `/opt/VRTSdbed/lib/dbed_mon_config.base` file contains the site-level configuration settings for monitoring all file systems and databases recognized. This configuration file specifies how often to check for file system and database configuration changes, how often to check the file space usage, where space usage information gets logged, and the thresholds for warning and automatically growing the file system. By default, the monitoring agent log file is located under the `/var/log/dbed_mon` directory.

During file system and database configuration, the `dbed_mon_config.base` file gets copied into each database-specific directory as the file `/etc/vx/vxdba/ORACLE_SID dbed_mon_config.$ORACLE_SID`. For example, if you are monitoring a database named `PROD`, the database-specific file would be `/etc/vx/vxdba/PROD/dbed_mon_config.PROD`. This is the first file opened when the agent is started and contains the default settings for monitoring file systems at the database level. The VxDBA Monitoring Agent cannot start without this file. Modify this configuration file if you want to change the preconfigured settings carried over from the `dbed_mon_config.base` file to maintain a different set of settings at the database level.

The following two files:

- `/etc/vx/vxdba/ORACLE_SID/dbed_mon_fslist.$ORACLE_SID`
- `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_oralist.$ORACLE_SID`

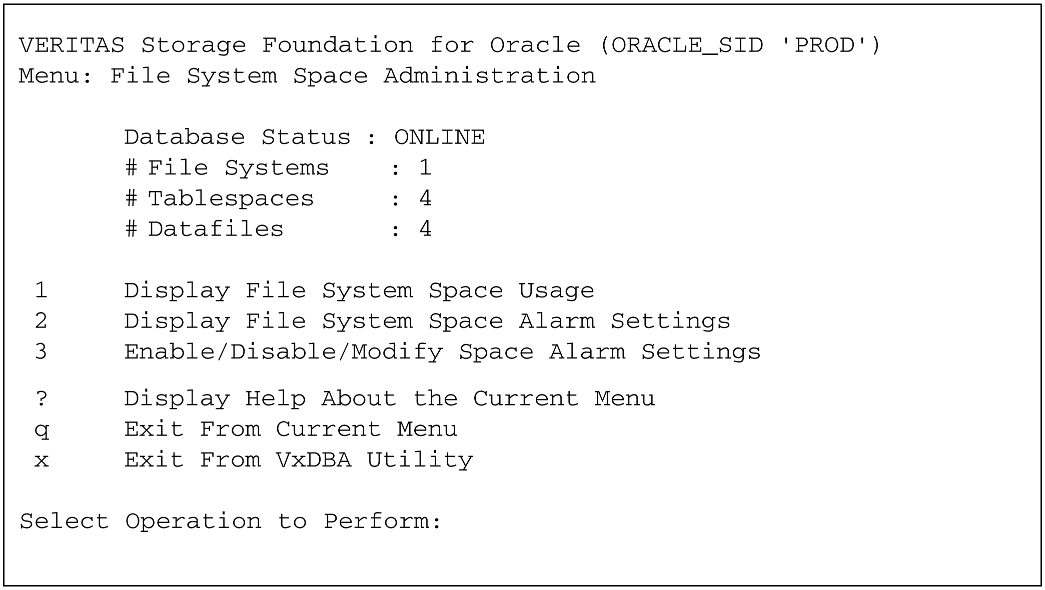
are created by the VxDBA utility and are used for restarting the VxDBA Monitoring Agent. These files specify the status of the database. The files also specify the space monitoring and alarm information for each file system, tablespace, and datafile. You can edit these files manually to change settings, and then use VxDBA to restart the Monitoring Agent.

The VxDBA Monitoring Agent uses the `/etc/vx/vxdba/$ORACLE_SID/include` file to check that all files are up-to-date and are being monitored. This file is created by the VxDBA utility and should not be edited.

Occasionally, Monitoring Agents ignore Storage Checkpoints. This happens when a Storage Checkpoint is not owned by the current Oracle instance. These Storage Checkpoints will not be used to calculate thresholds and potential removal candidates. Storage Checkpoints that are not considered part of the current Oracle database instance's data set are logged as such in the file `/var/log/dbed_mon/dbed_mon.prune_ckpt_log.$ORACLE_SID` when the Monitoring Agent is looking for potential removal candidates. A Storage Checkpoint must have an entry in the `/etc/vx/vxdba/$ORACLE_SID/checkpoint_dir` directory before it is considered owned by the database. This is done automatically by the provided VxDBA(1M) and `dbed_ckpt create`(1M) utilities and ensures that, if multiple databases share the same file system(s), the policy for one database does not affect another.

Use the **File System Space Administration** menu to monitor the space usage of the file system and each Storage Checkpoint. You can also use the menu to enable or disable the VxDBA Monitoring Agent.

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: File System Space Administration

       Database Status : ONLINE
       # File Systems    : 1
       # Tablespaces     : 4
       # Datafiles       : 4

 1     Display File System Space Usage
 2     Display File System Space Alarm Settings
 3     Enable/Disable/Modify Space Alarm Settings

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Select from the following File System Space Alarm Administration operations:

Display File System Space Usage

Use this menu option to display file system space usage information and the estimated space used for Storage Checkpoints created by VxDBA.

Display File System Space Alarm Settings

Use this menu option to display the space alarm information on the file systems used by the Oracle instance. The menu operation provides the boot-time status and current status of the VxDBA Monitoring Agent, as well as the list of file systems with their associated space alarm settings and status (ENABLED or DISABLED).

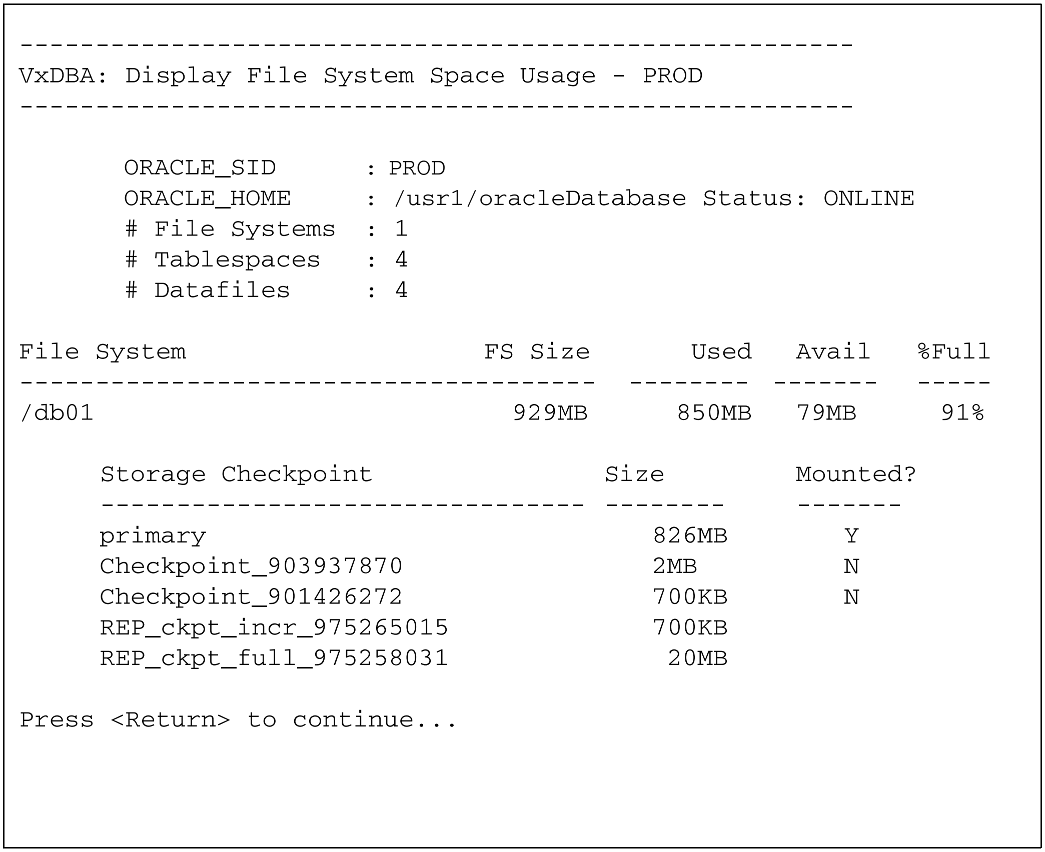
Enable/Disable/Modify Space Alarm Settings

Use this menu option to control the Monitoring Agent activity and modify the current space alarm settings on the file systems used by the Oracle instance.

Displaying File System Space Usage

This operation displays the space usage of the file system and the Storage Checkpoints used by the Oracle instance. For example, you can use this operation to monitor the daily database change history and use this data for capacity planning to forecast the additional disk space needed for Storage Checkpoints.

Display File System Usage displays a screen similar to the following:



```
-------------------------------------------------------
VxDBA: Display File System Space Usage - PROD
-------------------------------------------------------


       ORACLE_SID      : PROD
       ORACLE_HOME     : /usr1/oracleDatabase Status: ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

File System                   FS Size      Used   Avail   %Full
-------------------------------------  --------  -------   -----
/db01                           929MB     850MB   79MB      91%

     Storage Checkpoint               Size        Mounted?
     -------------------------------- --------    -------
     primary                             826MB       Y
     Checkpoint_903937870                2MB         N
     Checkpoint_901426272                700KB       N
     REP_ckpt_incr_975265015             700KB
     REP_ckpt_full_975258031              20MB

Press <Return> to continue...
```

In this example, the space used by `Checkpoint_901426272` is less than 1 MB, which means the Storage Checkpoint does not contain many data blocks. This means that the database has not modified many distinct data blocks since this Storage Checkpoint was created. Another Storage Checkpoint may have been created after this one, with all subsequent changes going to the new Storage Checkpoint.

Displaying File System Space Alarm Settings

This operation displays the information about the space alarm settings defined for the file systems used by the Oracle instance.

The space alarm relies on the VxDBA Monitoring Agent. The agent daemon processes must be running first. If the agent daemons are not running, a message is displayed asking you to start the agent daemons.

```
-------------------------------------------------------------
VxDBA: Display File System Space Alarm Settings - PROD
-------------------------------------------------------------



       ORACLE_SID      : PROD
       ORACLE_HOME     : /usr1/oracle
       Database Status :ONLINE

       # Tablespaces   : 4
       # Datafiles     : 4

Monitoring Agent is DISABLED at system boot time.

Monitoring Agent is not running.

The Monitoring Agent daemon must be running for
the file system space alarm to work. You can start
the Monitoring Agent using the VxDBA utility. From
the File System Space Administration menu, select
menu item 3 Enable/Disable/Modify Space Alarm
Settings to configure and start the Monitoring
Agent daemon. You can also start the Monitoring
Agent daemon automatically at system boot time
using this menu item.

Press <Return> to continue...
```

After you start the VxDBA Monitoring agent, **Display File System Space Alarm Settings** displays the list of file systems and the space alarm status:

```
-------------------------------------------------------------
VxDBA: Display File System Space Alarm Settings - PROD
-------------------------------------------------------------



        ORACLE_SID       : PROD
        ORACLE_HOME      : /usr1/oracle


                Database Status  : ONLINE
        # Tablespaces    : 4
        # Datafiles      : 4

Monitoring Agent is DISABLED at system boot time.

Monitoring Agent is running as pid 6991.

Press <Return> to continue...

-------------------------------------------------------------
VxDBA: Display File System Space Alarm Settings - PROD
-------------------------------------------------------------

  File System                         Status   Thresholds Grow By
                                               Warn Grow
  ----------------------------------- -------- ---- ----  ------
  /db01                               enabled  70     90      5%


Press <Return> to continue...
```

Enabling, Disabling, or Modifying Space Alarm Settings

When the file system runs out of space, VxFS automatically removes Storage Checkpoints to free up space. This could happen when Oracle is processing update transactions such that original data blocks are saved in the Storage Checkpoints. Enabling the space alarm allows VxDBA to monitor space usage and grow the file systems automatically, so that Storage Checkpoints are not unnecessarily removed.

Note Only users with superuser (`root`) privileges can perform this operation.

The **Enable/Disable/Modify Space Alarm Settings** operation first checks to see if you are logged in as `root`. If you are not logged in as `root`, VxDBA prompts you for the `root` password:

```
---------------------------------------------------------------
VxDBA: Enable/Disable/Modify Space Alarm Settings - PROD
---------------------------------------------------------------

You must be root to access the space alarm.

If you can enter the root password, you can continue.

Continue? [y,n,q,?] (default: y) y
Password:
```

After you enter the root password, **Enable/Disable/Modify Space Alarm Settings** displays a screen similar to the following:

```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Enable/Disable/Modify Space Alarm Settings

       Database Status : ORA-01031: insufficient privileges
       # File Systems  : (1)
       # Tablespaces   : (4)
       # Datafiles     : (4)


 1     Enable or Disable Boot-Time Start of Monitoring Agent
 2     Set Monitoring/Expansion Policy for File Systems
 3     Set Monitoring/Expansion Policy Per File System
 4     Re-Read Configuration File for Monitoring Agent
 5     Start/Stop Monitoring Agent

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Note When you run VxDBA operations as root, VxDBA cannot connect to and obtain information directly from the database, so the submenu **Database Status** header reports a permission error, and the number of tablespaces and datafiles are enclosed in parentheses.

Select from the following file system space alarm operations:

Enable or Disable Boot-Time Start of Monitoring Agent. Use this menu option to change the boot-time start activity of the VxDBA Monitoring Agent. You are provided with the current setting (ENABLED or DISABLED), and then prompted for changes.

While this is not recommended, you can disable the boot-time start activity of the VxDBA Monitoring Agent that monitors the space usage of the file systems used by the Oracle instance.

Note If the space alarm is disabled and a file system that contains Storage Checkpoints runs out of space, VxFS removes Storage Checkpoints to free up the space.

Set Monitoring/Expansion Policy for All File Systems. Use this menu option to configure the monitoring and expansion policy for all file systems. You are prompted for the Warning Threshold for space usage, the Grow Threshold for space usage, and the Amount as a percentage or a value in megabytes by which to grow the file system. These three policy values are then used for all file systems unless a per file system policy is set.

Set Monitoring/Expansion Policy Per File System. Use this menu option to configure the monitoring and expansion policy for a particular file system. After displaying the current policy per file system, you are asked if you want to change the policies and are then prompted for the new values for the Warning Threshold for space usage, the Grow Threshold for space usage, and the Amount as a percentage or a value in megabytes by which to grow the file system. If you want to disable monitoring of a particular file system, set the Grow Threshold and the amount to Grow By values to zero. By default, the expansion of file systems is disabled and must be enabled by the user.

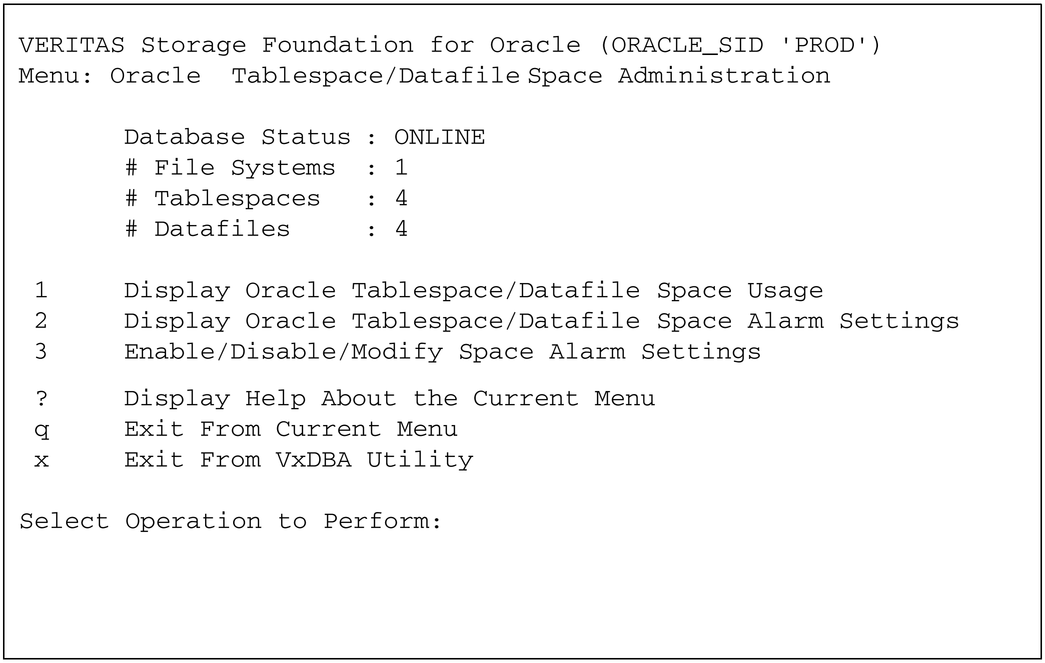
Re-Read Configuration File for Monitoring Agent. Use this menu option to re-read the `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_fslist.$ORACLE_SID` file. Select this operation if you manually edit this file.

Start/Stop Monitoring Agent. Use this menu option to start or stop the monitoring agent. After displaying the current status of the Monitoring Agent (running or not running), you can either start or stop the Monitoring Agent.

Managing Oracle Tablespace and Datafile Space

Use the **Oracle Tablespace/DatafileSpace Administration** menu to monitor the space usage of Oracle tablespaces and datafiles, and to display or modify the VxDBA Monitoring Agent's Oracle space alarm settings. You can also use the menu to enable or disable the VxDBA Monitoring Agent.

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Oracle  Tablespace/Datafile Space Administration

       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

 1     Display Oracle Tablespace/Datafile Space Usage
 2     Display Oracle Tablespace/Datafile Space Alarm Settings
 3     Enable/Disable/Modify Space Alarm Settings

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Select from the following File System Space Alarm Administration operations:

Display Oracle Tablespace/Datafile Space Usage. Use this menu option to display datafile space usage information. Total size of Oracle objects, free space available, and Oracle blocks are displayed. Oracle objects and free space available are given in MB.

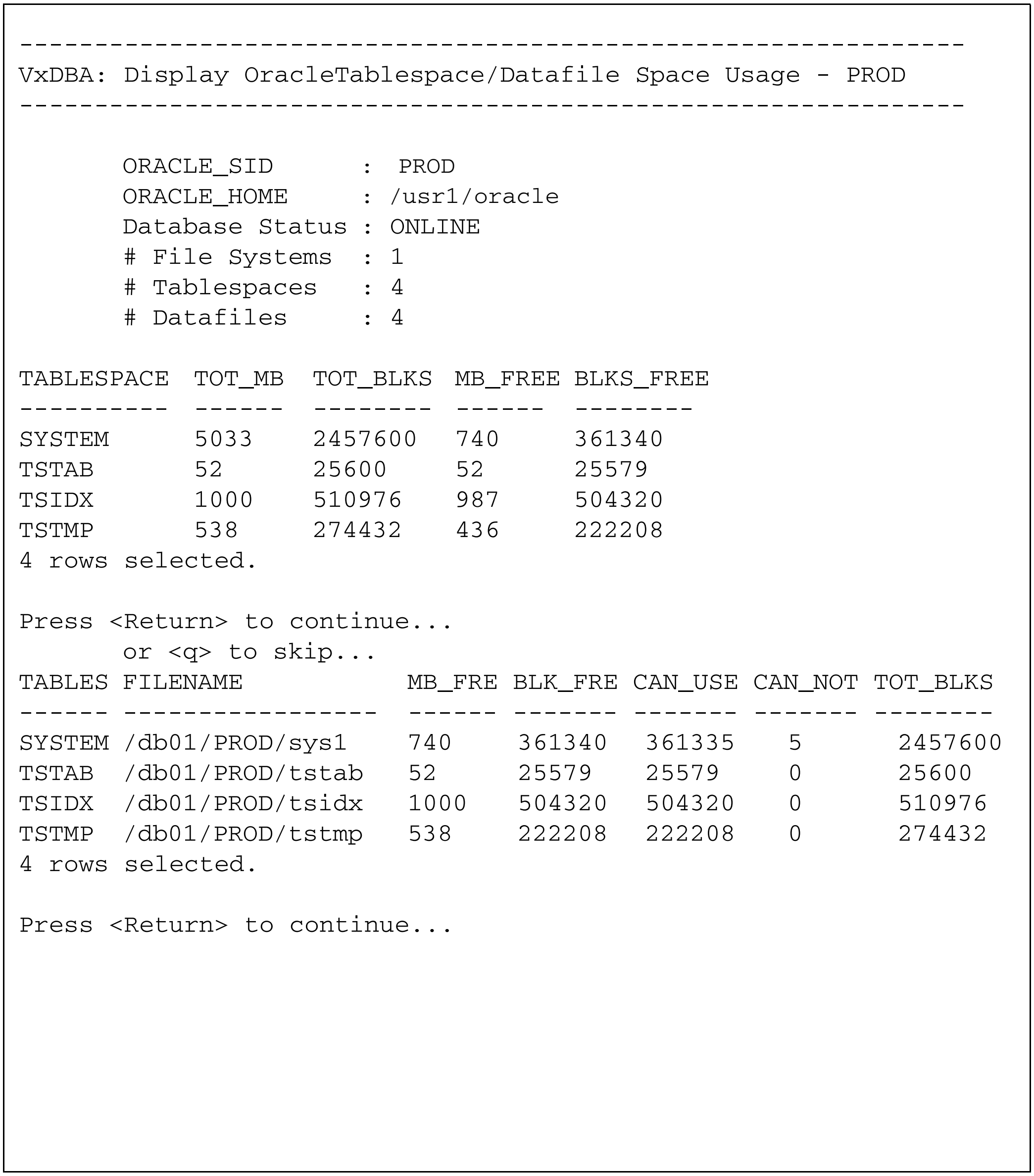
Display Oracle Tablespace/Datafile Space Alarm Settings. Use this menu operation to display the space alarm information on the Oracle tablespaces. The menu display provides the boot-time and current status of the VxDBA Monitoring Agent and the list of Oracle tablespaces with their associated space alarm settings and status (ENABLED or DISABLED).

Enable/Disable/Modify Space Alarm Settings. Use this menu option to change the boot-time start activity of the VxDBA Monitoring Agent. You are provided with the current setting (ENABLED or DISABLED), and then prompted for changes.

Displaying Oracle Tablespace/Datafile Space Usage

This operation displays the space usage of Oracle tablespaces, datafiles, and the Storage Checkpoints used by the Oracle instance.

Display Oracle Tablespace/Datafile Space Usage displays a screen similar to the following:



```
----------------------------------------------------------------
VxDBA: Display OracleTablespace/Datafile Space Usage - PROD
----------------------------------------------------------------

       ORACLE_SID      :  PROD
       ORACLE_HOME     : /usr1/oracle
       Database Status : ONLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

TABLESPACE  TOT_MB  TOT_BLKS  MB_FREE BLKS_FREE
----------  ------  --------  ------  --------
SYSTEM      5033    2457600   740     361340
TSTAB       52      25600     52      25579
TSIDX       1000    510976    987     504320
TSTMP       538     274432    436     222208
4 rows selected.

Press <Return> to continue...
       or <q> to skip...
TABLES FILENAME           MB_FRE BLK_FRE CAN_USE CAN_NOT TOT_BLKS
------ -----------------  ------ ------- ------- ------- --------
SYSTEM /db01/PROD/sys1    740     361340  361335   5      2457600
TSTAB  /db01/PROD/tstab   52      25579   25579    0      25600
TSIDX  /db01/PROD/tsidx   1000    504320  504320   0      510976
TSTMP  /db01/PROD/tstmp   538     222208  222208   0      274432
4 rows selected.

Press <Return> to continue...
```

Displaying Oracle Tablespace/Datafile Space Alarm Settings

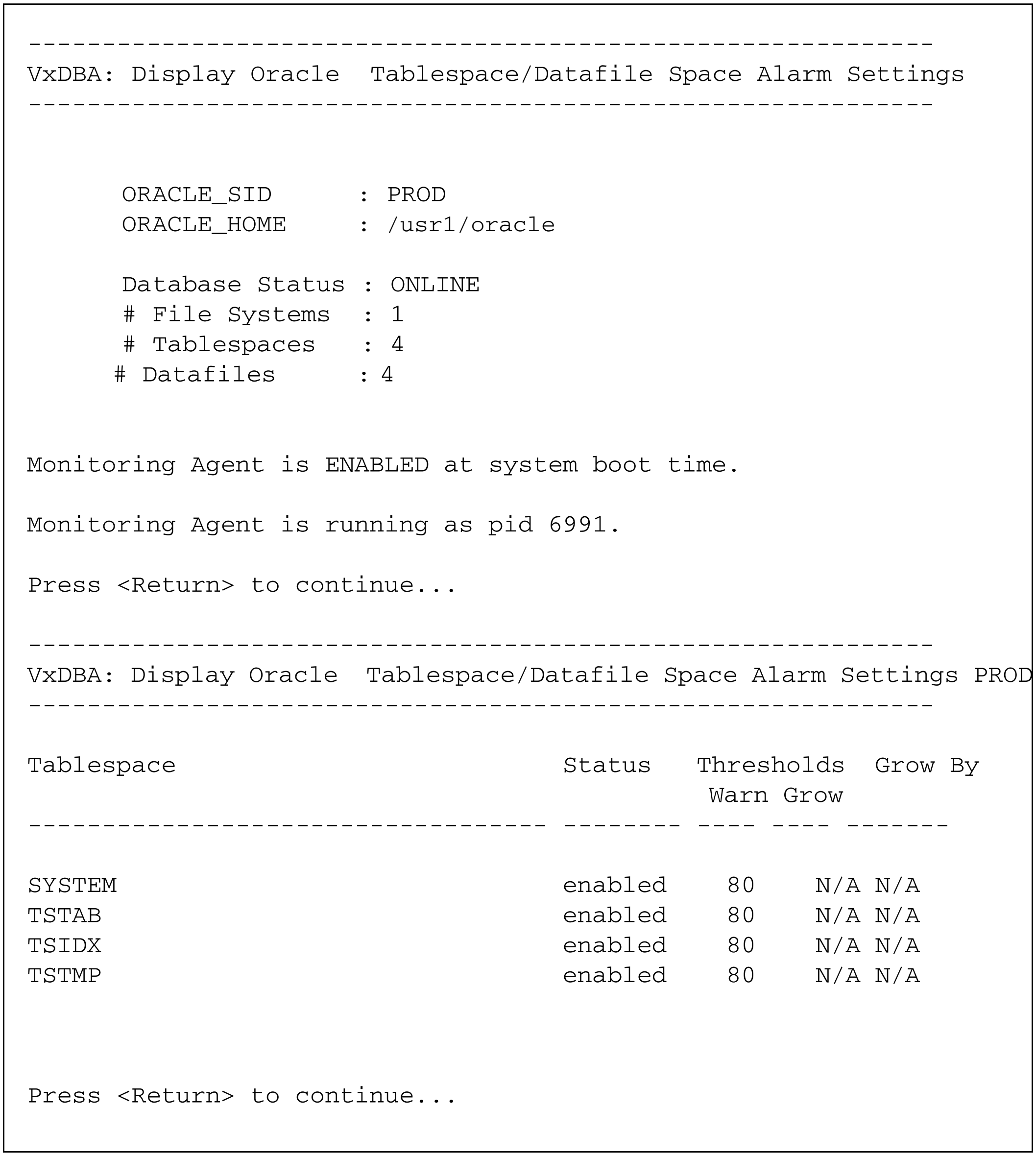
This operation displays the information about the space alarm settings defined for the Oracle tablespaces and datafiles.

The space alarm relies on the VxDBA Monitoring Agent. The agent daemon processes must be running first. If the agent daemons are not running, a message is displayed asking you to start the agent daemons.

The expansion of Oracle Tablespaces is not currently supported. The display for settings on the grow threshold and the grow amount always shows N/A.

```
-------------------------------------------------------------
VxDBA: Display Oracle Tablespace/Datafile Space Alarm Settings -
PROD
-------------------------------------------------------------


       ORACLE_SID      : PROD
       ORACLE_HOME     : /usr1/oracle

       Database Status : OFFLINE
       # File Systems  : 1
       # Tablespaces   : 4
       # Datafiles     : 4

Monitoring Agent is DISABLED at system boot time.

Monitoring Agent is not running.

The Monitoring Agent daemon must be running for
the oracle tablespace/datafile space alarm to work.
You can start the Monitoring Agent using the VxDBA
utility. From the Oracle Tablespace/Datafile Space
Administration menu, select menu item 3 Enable/Disable
/Modify Space Alarm Settings to configure and start
the Monitoring Agent daemon. You can also start the
Monitoring Agent daemon automatically at system boot
time using this menu item.

Press <Return> to continue...
```

Once you start the VxDBA Monitoring agent, **Display Oracle Tablespace/Datafile / Space Alarm Settings** displays the list of tablespaces and datafiles and the space alarm status:



```
-------------------------------------------------------------
VxDBA: Display Oracle  Tablespace/Datafile Space Alarm Settings
-------------------------------------------------------------



      ORACLE_SID      : PROD
      ORACLE_HOME     : /usr1/oracle

      Database Status : ONLINE
      # File Systems  : 1
      # Tablespaces   : 4
      # Datafiles     : 4


Monitoring Agent is ENABLED at system boot time.

Monitoring Agent is running as pid 6991.

Press <Return> to continue...

-------------------------------------------------------------
VxDBA: Display Oracle  Tablespace/Datafile Space Alarm Settings PROD
-------------------------------------------------------------

Tablespace                          Status   Thresholds  Grow By
                                              Warn Grow
----------------------------------- -------- ---- ---- -------

SYSTEM                              enabled    80    N/A N/A
TSTAB                               enabled    80    N/A N/A
TSIDX                               enabled    80    N/A N/A
TSTMP                               enabled    80    N/A N/A



Press <Return> to continue...
```

Enabling, Disabling, or Modifying Oracle Space Alarm Settings

Enabling the Oracle space alarm allows VxDBA to monitor tablespace and datafile space usage. The warning is sent to the log file

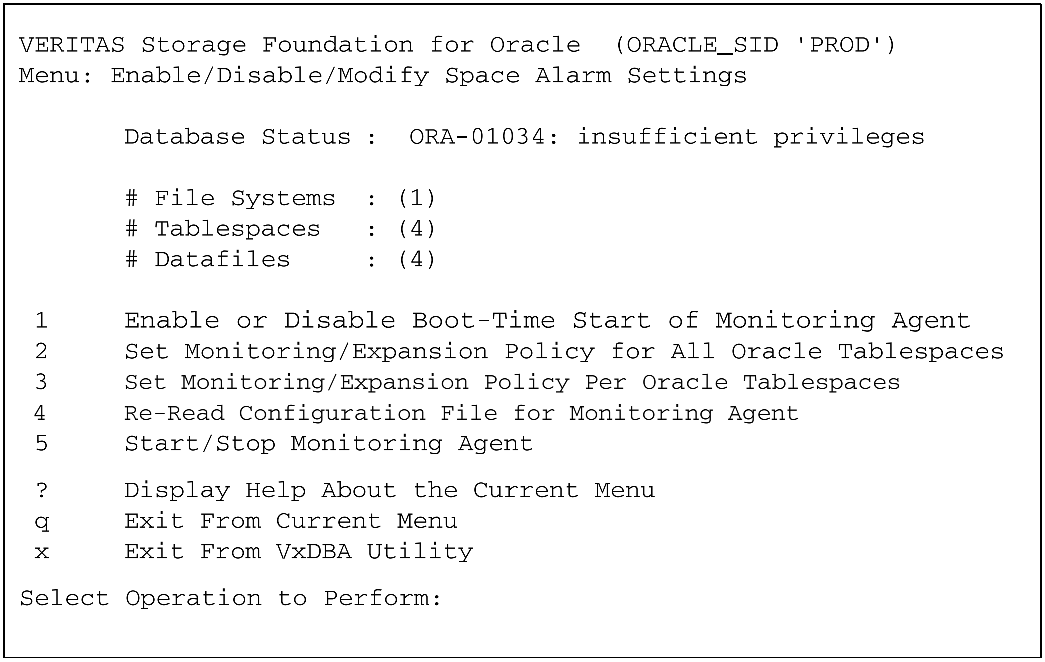
```
/var/log/dbed_mon/dbed_mon.logfile.$ORACLE_SID
```

Note Only users with superuser (`root`) privileges can perform this operation.

The **Enable/Disable/Modify Oracle Space Alarm Settings** operation first checks to see if you are logged in as `root`. If you are not logged in as `root`, VxDBA prompts you for the `root` password:

```
---------------------------------------------------------------
VxDBA: Enable/Disable/Modify Space Alarm Settings - PROD
---------------------------------------------------------------

You must be root to access the space alarm.

If you can enter the root password, you can continue.

Continue? [y,n,q,?] (default: y) y
Password:
```

After you enter the root password, **Enable/Disable/Modify Space Alarm Settings** displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle  (ORACLE_SID 'PROD')
Menu: Enable/Disable/Modify Space Alarm Settings

       Database Status :  ORA-01034: insufficient privileges

       # File Systems  : (1)
       # Tablespaces   : (4)
       # Datafiles     : (4)

 1     Enable or Disable Boot-Time Start of Monitoring Agent
 2     Set Monitoring/Expansion Policy for All Oracle Tablespaces
 3     Set Monitoring/Expansion Policy Per Oracle Tablespaces
 4     Re-Read Configuration File for Monitoring Agent
 5     Start/Stop Monitoring Agent

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Note When you run VxDBA operations as root, VxDBA cannot connect to and obtain information directly from the database, so the submenu **Database Status** header reports a permission error, and the number of tablespaces and datafiles are enclosed in parentheses.

Select from the following Oracle space alarm operations:

Enable or Disable Boot-Time Start of Monitoring Agent. Use this menu option to change the boot-time start activity of the monitoring agent. The message provides you with the current setting and prompts you for changes.

Set Monitoring/Expansion Policy for All Oracle Tablespaces. Use this menu option to configure the monitoring policy for all Oracle tablespaces. The menu prompts you for the Warning Threshold for space usage.

Note The expansion of Oracle Tablespaces is not currently supported. No option to modify settings for the grow threshold or grow amount is available.

Set Monitoring Policy Per Oracle Tablespaces. Use this menu option to configure the monitoring policy for a particular Oracle tablespace. After displaying the current policy per Oracle tablespace, the program asks if you want to change the policies and then prompts you for the new values for the Warning Threshold.

Note The expansion of Oracle Tablespaces is not currently supported. No option to modify settings for the grow threshold or grow amount is available.

Re-Read Configuration File for Monitoring Agent. Use this menu option to re-read the `/etc/vx/vxdba/$ORACLE_SID/dbed_mon_oralist.$ORACLE_SID` file. Select this operation if you manually edited this file.

Start/Stop Monitoring Agent. Use this menu option to start or stop the monitoring agent.

Configuring Monitoring Agent Options

Use this menu operation to modify current default settings for the Monitoring Agent. The agent configuration is saved under the following:

```
/etc/vx/vxdba/$ORACLE_SID/dbed_mon_config.$ORACLE_SID
```

When the configuration file is modified, restart the Monitoring Agent for the changes to take effect. This operation displays a screen similar to the following:

```
-------------------------------------------------------------
VxDBA: Configure Monitoring Agent Options - PROD
-------------------------------------------------------------
You will now be prompted to modify current VxDBA
Monitoring Agent settings.
For detailed information on these settings, see
the dbed_mon(1M) manual page and the Monitoring
Agent configuration file in the following location:
/etc/vx/vxdba/PROD/dbed_mon_config.PROD
Continue [y,n,q,?] (default: y)
NOTICE: Default setting for CHECK is 300.
Do you want to change the setting for CHECK [y,n,q,?] (default: n) y
Enter new setting for CHECK [current 300] : 60
Changing variable CHECK from 300 to 60
NOTICE: Default setting for LOGFREQ is 0.
Do you want to change the setting for LOGFREQ [y,n,q,?] (default: n)
n
NOTICE: Default setting for DEF_WARN is 85.
Do you want to change the setting for DEF_WARN [y,n,q,?] (default: n)
n
```

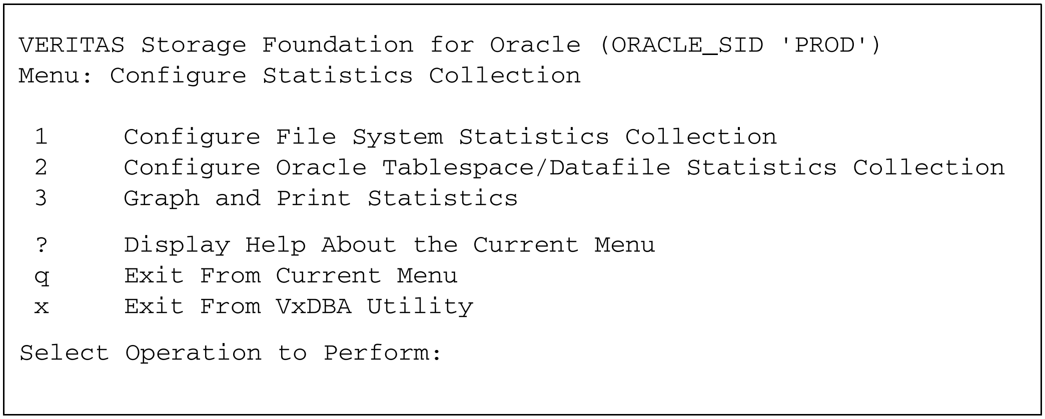
```
NOTICE: Default setting for DEF_GROW is 90.
  Do you want to change the setting for DEF_GROW [y,n,q,?] (default:
n)
  NOTICE: Default setting for DEF_GROWBY is 0%.
  Do you want to change the setting for DEF_GROWBY [y,n,q,?] (default:
n)
  n
  NOTICE: Default setting for DEF_ORA_WARN is 85.
  Do you want to change the setting for DEF_ORA_WARN [y,n,q,?]
(default:
  n) n
  WARNING: No default setting for LOGPATH variable.
  Do you want to change the setting for LOGPATH [y,n,q,?] (default: n)
n
  WARNING: No default setting for LOG_EMAIL variable.
  Do you want to change the setting for LOG_EMAIL [y,n,q,?] (default:
n) n
  WARNING: No default setting for SYSLOG_FACILITY variable.
  Do you want to change the setting for SYSLOG_FACILITY [y,n,q,?]
(default: n)
  WARNING: No default setting for SYSLOG_PRIORITY variable.
  Do you want to change the setting for SYSLOG_PRIORITY [y,n,q,?]
  (default: n) n
  NOTICE: Default setting for STATS is 0.
  Do you want to change the setting for STATS [y,n,q,?] (default: n) y
  Changing variable STATS from 0 to 1800
  Changing variable STATS from 0 to 1800
Enter new setting for STATS [current 0] : 1800
```

```
 WARNING: No default setting for FSSTATSPATH variable.

 Do you want to change the setting for FSSTATSPATH [y,n,q,?]
(default: n)
 n

 NOTICE: Default setting for ORASTATS is 0.

 Do you want to change the setting for ORASTATS [y,n,q,?] (default:
n)
 Enter new setting for ORASTATS [current 0] : 1800
 Changing variable ORASTATS from 0 to 1800

 WARNING: No default setting for ORA_STATSPATH variable.

 Do you want to change the setting for ORASTATSPATH [y,n,q,?]
 (default:n) n

 The following changes are to be made to the Monitoring
 Agent configuration file.

 CHECK 60
 ORASTATS 1800


 Do you want to commit the changes [y,n,q,?] (default: n)y
 Changes committed.

 You need to restart the Monitoring Agent for these
 configuration changes to take effect.

 Do you want to restart the Monitoring Agent [y,n,q,?] (default: y)
```

Configuring Statistics Collection

Use the **Configure Statistics Collection** menu to collect file system and Oracle tablespace and datafile space usage statistics.

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Configure Statistics Collection

 1     Configure File System Statistics Collection
 2     Configure Oracle Tablespace/Datafile Statistics Collection
 3     Graph and Print Statistics

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Select from the following **Configure Statistics Collection** operations:

Configure File System Statistics Collection. Use this menu option to modify agent settings to collect file system space statistics. Variables relevant to statistics are as follows:.

`STATS`: Interval between samples. Default is 0 (disabled).

`FSSTATSPATH`: Path to a file where file system space statistics will be recorded. Default file is `/var/log/dbed_mon/dbed_mon.fs_stats.$ORACLE_SID`.

Ensure that there is adequate space in the specified locations to hold the statistics. The amount of space required will vary depending on the number of file systems being monitored and the sampling interval for the statistics.

Configure Oracle Tablespace/Datafile Statistics Collection. Use this menu option to modify agent settings to collect Oracle space statistics. Variables relevant to statistics are as follows:

`ORA_STATS` : Interval between samples. Default is 0 (disabled).

`ORASTATSPATH` : Path to a file where Oracle space statistics will be recorded. Default file is `/var/log/dbed_mon/dbed_mon.ora_stats.$ORACLE_SID`.

Ensure that there is adequate space in the specified locations to hold the statistics. The amount of space required will vary depending on the number of tablespaces and datafiles being monitored and the sampling interval for the statistics.

See "Configuring Monitoring Agent Options" on page 395 for the setting's output.

Graph and Print Statistics. Use this menu operation to display or print the collected statistics. If you try to print and the PRINTER environment variable is not set, you will have to supply a printer name. If you try to create a graph and the DISPLAY environment variable is not set, you will have to supply a machine display name. Graphing statistics depend on access to an X-windows system.

This operation displays a screen similar to the following:

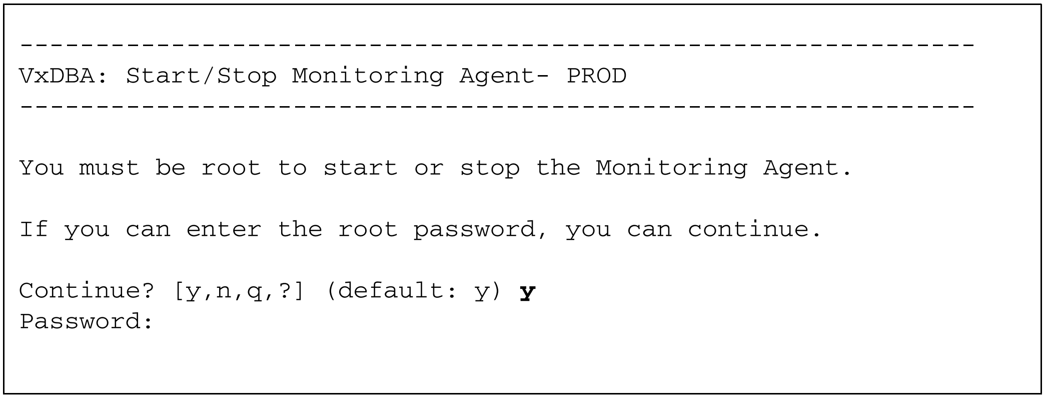
```
-------------------------------------------------------------
VxDBA: Graph and Print Statistics - PROD
-------------------------------------------------------------

Log file containing file system statistics is available.

Log file containing Oracle statistics is available.

Do you want to display or print any statistics? [y,n,q,?]
(default: n) y

VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Graph and Print Statistics

 1      Use File System space statistics
 2      Use Oracle space statistics

 ?      Display Help About the Current Menu
 q      Exit From Current Menu
 x      Exit From VxDBA Utility

Select Operation to Perform: 1

Would you like to print or display statistics?[print,display,q,?]
display

DISPLAY environment variable is not set.
```

```
Enter a display name for graph display [<name>,q,?] abbey:0.0
DISPLAY set to abbey:0.0.

Found a statistics file at
/var/log/dbed_mon/dbed_mon.fs_stats.PROD



Do you want to supply another file name? [<name>,n,q,?]  n

Validating file format of
/var/log/dbed_mon/dbed_mon.fs_stats.PROD

Setting temp directory to /tmp/.stats
Saving all output files for later use.
/db01
To graph to an X window, use:
         /opt/VRTSdbed/stats/gnuplot
        'load "/tmp/.stats/gr.commands._db01.10209"'
Press <Return> to continue...
  or <q> to skip...
```

Starting and Stopping the Monitoring Agent

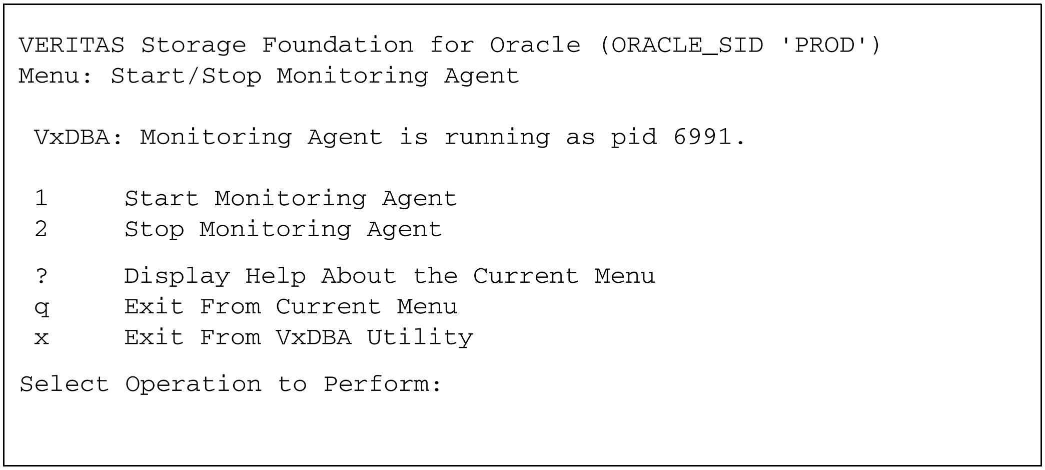
Note Only users with superuser (`root`) privileges can perform this operation.

The **Start/Stop Monitoring Agent** operation first checks to see if you are logged in as `root`. If you are not logged in as `root`, VxDBA prompts you for the `root` password:



```
---------------------------------------------------------------
VxDBA: Start/Stop Monitoring Agent- PROD
---------------------------------------------------------------

You must be root to start or stop the Monitoring Agent.

If you can enter the root password, you can continue.

Continue? [y,n,q,?] (default: y) y
Password:
```

After you enter the `root` password, **Start/Stop Monitoring Agent** displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Start/Stop Monitoring Agent

 VxDBA: Monitoring Agent is running as pid 6991.

 1      Start Monitoring Agent
 2      Stop Monitoring Agent

 ?      Display Help About the Current Menu
 q      Exit From Current Menu
 x      Exit From VxDBA Utility

Select Operation to Perform:
```

Select from the following menu operations:

Start Monitoring Agent. Use this menu option to start the monitoring activity of the VxDBA Monitoring Agent.

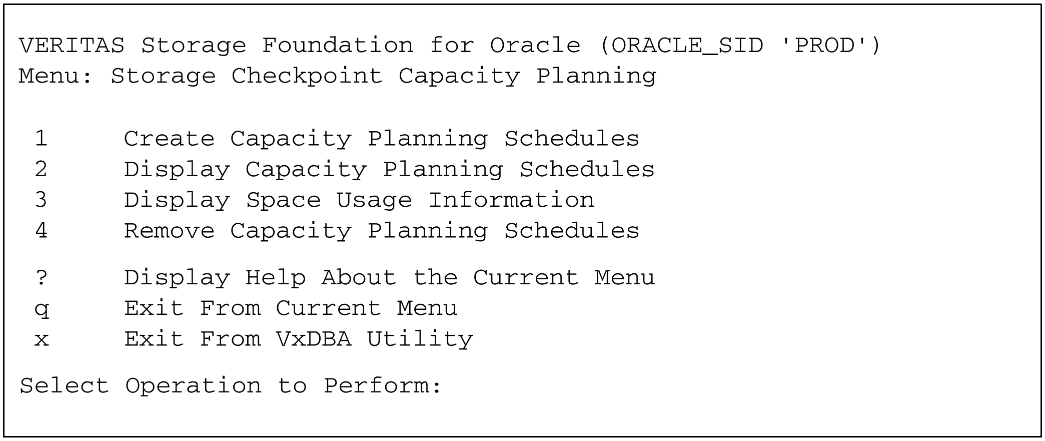
Stop Monitoring Agent. Use this menu option to stop the monitoring activity of the VxDBA Monitoring Agent.

Note VxDBA keeps a record of the Monitoring Agent process ID. To avoid any inconsistent Monitoring Agent status (running or not running), do not stop the Monitoring Agent outside of VxDBA (for example, using the `kill`(1M) command).

Planning File System Space for Storage Checkpoints

Use the **Storage Checkpoint Capacity Planning** menu option to plan adequate space for Storage Checkpoints.

This operation displays a screen similar to the following:



```
VERITAS Storage Foundation for Oracle (ORACLE_SID 'PROD')
Menu: Storage Checkpoint Capacity Planning

 1     Create Capacity Planning Schedules
 2     Display Capacity Planning Schedules
 3     Display Space Usage Information
 4     Remove Capacity Planning Schedules

 ?     Display Help About the Current Menu
 q     Exit From Current Menu
 x     Exit From VxDBA Utility

Select Operation to Perform:
```

Setting Up VxDBA in an HA Environment

VxDBA puts its repository and lock files in the directory `/etc/vx/vxdba`. The repository for a particular database instance (noted by `$ORACLE_SID`) is under directory `/etc/vx/vxdba/$ORACLE_SID`. If you are configuring your database in a high availability or cluster environment, you must configure this directory for failover.

In a VERITAS Cluster Server (VCS) environment, you should make the directory `/etc/vx/vxdba/$ORACLE_SID` the mount point of a file system that mounts on a shared disk device. Also, set the group and owner to the Oracle DBA user and set up the proper configuration for the mount agent. See the *VERITAS Cluster Server Installation Guide* for more information.

Tuning for Performance 14

This chapter provides tuning tips that you can use to improve database performance.

Topics covered in this chapter include:

- "Tuning VxVM" on page 408
- "Tuning VxFS" on page 410
- "Tuning Oracle Databases" on page 419
- "Tuning Solaris" on page 422

Use the tuning tips and information provided in this chapter in conjunction with other more in-depth publications, such as:

- *Oracle Performance Tuning Tips & Techniques* (Osborne McGraw-Hill)
- *Oracle8i Installation Guide* (Oracle Corporation)
- *Oracle9i Installation Guide* (Oracle Corporation)
- *Oracle Performance Tuning* (O'Reilly & Associates)
- *Oracle Performance Tuning and Optimization* (Sams Publishing)
- *VERITAS Volume Manager Administrator's Guide*, chapter on "VxVM Performance Monitoring"

Tuning VxVM

VERITAS Volume Manager (VxVM) is tuned for most configurations ranging from small systems to larger servers. On smaller systems with less than a hundred drives, tuning should not be necessary and VERITAS Volume Manager should be capable of adopting reasonable defaults for all configuration parameters. On very large systems, however, there may be configurations that require additional tuning of these parameters, both for capacity and performance reasons. For information on tuning VERITAS Volume Manager, refer to the "Tuning VxVM" section of the "Performance Monitoring and Tuning" chapter in the *VERITAS Volume Manager Administrator's Guide*.

Obtaining Volume I/O Statistics

If your database is created on a single file system that is on a single volume, there is typically no need to monitor the volume I/O statistics. If your database is created on multiple file systems on multiple volumes, or the volume configurations have changed over time, it may be necessary to monitor the volume I/O statistics for the databases.

Use the `vxstat` command to access information about activity on volumes, plexes, subdisks, and disks under VxVM control, and to print summary statistics to the standard output. These statistics represent VxVM activity from the time the system initially booted or from the last time the counters were reset to zero. If no VxVM object name is specified, statistics from all volumes in the configuration database are reported. Use the `-g` option to specify the database disk group to report statistics for objects in that database disk group.

VxVM records the following I/O statistics:

- count of operations
- number of blocks transferred (one operation can involve more than one block)
- average operation time (which reflects the total time through the VxVM interface and is not suitable for comparison against other statistics programs)

VxVM records the preceding three pieces of information for logical I/Os, including reads, writes, atomic copies, verified reads, verified writes, plex reads, and plex writes for each volume. VxVM also maintains other statistical data such as read failures, write failures, corrected read failures, corrected write failures, and so on. In addition to displaying volume statistics, the `vxstat` command is capable of displaying more detailed statistics on the components that form the volume. For detailed information on available options, refer to the `vxstat`(1M) manual page.

To reset the statistics information to zero, use the `-r` option. You can reset the statistics information for all objects or for only those objects that are specified. Resetting just prior to an operation makes it possible to measure the impact of that particular operation.

The following is an example of output produced using the `vxstat` command:

```
 OPERATIONS          BLOCKS       AVG TIME(ms)
TYP  NAME       READ   WRITE     READ     WRITE    READ     WRITE
vol  blop          0       0        0         0     0.0       0.0
vol  foobarvol     0       0        0         0     0.0       0.0
vol  rootvol   73017  181735   718528   1114227    26.8      27.9
vol  swapvol   13197   20252   105569    162009    25.8     397.0
vol  testvol       0       0        0         0     0.0       0.0
```

The “Performance Monitoring” section of the “Performance Monitoring and Tuning” chapter in the *VERITAS Volume Manager Administrator’s Guide* provides detailed information on how to use the `vxstat` output to identify volumes that have excessive activity and how to reorganize, change to a different layout, or move these volumes.

Additional volume statistics are available for RAID-5 configurations. Refer to the `vxstat`(1M) manual page for more information.

Tuning VxFS

VERITAS File System provides a set of tuning options to optimize file system performance for different application workloads. VxFS provides a set of tunable I/O parameters that control some of its behavior. These I/O parameters help the file system adjust to striped or RAID-5 volumes that could yield performance far superior to a single disk. Typically, data streaming applications that access large files see the largest benefit from tuning the file system.

Most of these tuning options have little or no impact on database performance when using Quick I/O. However, you can gather file system performance data when using Quick I/O, and use this information to adjust the system configuration to make the most efficient use of system resources.

Monitoring Free Space

In general, VxFS works best if the percentage of free space in the file system is greater than 10 percent. This is because file systems with 10 percent or more of free space have less fragmentation and better extent allocation. Regular use of the `df` command to monitor free space is desirable. Full file systems may have an adverse effect on file system performance. Full file systems should therefore have some files removed or should be expanded. See the `fsadm_vxfs`(1M) manual page for a description of online file system expansion.

Monitoring Fragmentation

Fragmentation reduces performance and availability. Regular use of `fsadm`'s fragmentation reporting and reorganization facilities is therefore advisable.

The easiest way to ensure that fragmentation does not become a problem is to schedule regular defragmentation runs using the `cron` command.

Defragmentation scheduling should range from weekly (for frequently used file systems) to monthly (for infrequently used file systems). Extent fragmentation should be monitored with `fsadm` or the `df -o s` commands. There are three factors that can be used to determine the degree of fragmentation:

- Percentage of free space in extents that are less than eight blocks in length
- Percentage of free space in extents that are less than 64 blocks in length
- Percentage of free space in extents that are 64 or more blocks in length

An unfragmented file system will have the following characteristics:

- Less than 1 percent of free space in extents that are less than eight blocks in length
- Less than 5 percent of free space in extents that are less than 64 blocks in length

- More than 5 percent of the total file system size available as free extents that are 64 or more blocks in length

A badly fragmented file system will have one or more of the following characteristics:

- More than 5 percent of free space in extents that are less than 8 blocks in length
- More than 50 percent of free space in extents that are less than 64 blocks in length
- Less than 5 percent of the total file system size available as free extents that are 64 or more blocks in length

The optimal period for scheduling extent reorganization runs can be determined by choosing a reasonable interval, scheduling `fsadm` runs at the initial interval, and running the extent fragmentation report feature of `fsadm` before and after the reorganization.

The "before" result is the degree of fragmentation prior to the reorganization. If the degree of fragmentation approaches the percentages for bad fragmentation, reduce the interval between `fsadm`. If the degree of fragmentation is low, increase the interval between `fsadm` runs.

Tuning VxFS I/O Parameters

VxFS provides a set of tunable I/O parameters that control some of its behavior. These I/O parameters are useful to help the file system adjust to striped or RAID-5 volumes that could yield performance far superior to a single disk. Typically, data streaming applications that access large files see the biggest benefit from tuning the file system.

If VxFS is being used with VERITAS Volume Manager, the file system queries VxVM to determine the geometry of the underlying volume and automatically sets the I/O parameters. VxVM is queried by `mkfs` when the file system is created to automatically align the file system to the volume geometry. If the default alignment from `mkfs` is not acceptable, the `-o align=n` option can be used to override alignment information obtained from VxVM. The `mount` command also queries VxVM when the file system is mounted and downloads the I/O parameters.

If the default parameters are not acceptable or the file system is being used without VxVM, then the `/etc/vx/tunefstab` file can be used to set values for I/O parameters. The `mount` command reads the `/etc/vx/tunefstab` file and downloads any parameters specified for a file system. The `tunefstab` file overrides any values obtained from VxVM. While the file system is mounted, any I/O parameters can be changed using the `vxtunefs` command, which can have tunables specified on the command line or can read them from the `/etc/vx/tunefstab` file. For more details, see the `vxtunefs`(1M) and `tunefstab`(4) manual pages. The `vxtunefs` command can be used to print the current values of the I/O parameters.

Tunable VxFS I/O Parameters

read_pref_io

The preferred read request size. The file system uses this parameter in conjunction with the read_nstream value to determine how much data to read ahead. The default value is 64K.

write_pref_io

The preferred write request size. The file system uses this parameter in conjunction with the write_nstream value to determine how to do flush behind on writes. The default value is 64K.

read_nstream

The number of parallel read requests of size read_pref_io that you can have outstanding at one time. The file system uses the product of read_nstream multiplied by read_pref_io to determine its read ahead size. The default value for read_nstream is 1.

write_nstream

The number of parallel write requests of size write_pref_io that you can have outstanding at one time. The file system uses the product of write_nstream multiplied by write_pref_io to determine when to do flush behind on writes. The default value for write_nstream is 1.

default_indir_size

On VxFS, files can have up to ten variably sized direct extents stored in the inode. After these extents are used, the file must use indirect extents that are a fixed size. The size is set when the file first uses indirect extents. These indirect extents are 8K by default. The file system does not use larger indirect extents because it must fail a write and return ENOSPC if there are no extents available that are the indirect extent size. For file systems with many large files, the 8K indirect extent size is too small. Large files that require indirect extents use many smaller extents instead of a few larger ones. By using this parameter, the default indirect extent size can be increased so that large files in indirects use fewer large extents.

Be careful using this tunable. If it is too large, then writes fail when they are unable to allocate extents of the indirect extent size to a file. In general, the fewer and the larger the files on a file system, the larger the default_indir_size parameter can be. The value of this parameter is generally a multiple of the read_pref_io parameter.

This tunable is not applicable on Version 4 disk layouts.

`discovered_direct_iosz`
Any file I/O requests larger than the `discovered_direct_iosz` are handled as discovered direct I/O. A discovered direct I/O is unbuffered similar to direct I/O, but does not require a synchronous commit of the inode when the file is extended or blocks are allocated. For larger I/O requests, the CPU time for copying the data into the page cache and the cost of using memory to buffer the I/O data becomes more expensive than the cost of doing the disk I/O. For these I/O requests, using discovered direct I/O is more efficient than regular I/O. The default value of this parameter is 256K.

`initial_extent_size`
Changes the default initial extent size. VxFS determines the size of the first extent to be allocated to the file based on the first write to a new file. Normally, the first extent is the smallest power of 2 that is larger than the size of the first write. If that power of 2 is less than 8K, the first extent allocated is 8K. After the initial extent, the file system increases the size of subsequent extents (see `max_seqio_extent_size`) with each allocation. Since most applications write to files using a buffer size of 8K or less, the increasing extents start doubling from a small initial extent. `initial_extent_size` can change the default initial extent size to be larger, so the doubling policy will start from a much larger initial size and the file system will not allocate a set of small extents at the start of file. Use this parameter only on file systems that will have a very large average file size. On these file systems, it will result in fewer extents per file and less fragmentation. `initial_extent_size` is measured in file system blocks.

`max_direct_iosz`
The maximum size of a direct I/O request that will be issued by the file system. If a larger I/O request comes in, then it is broken up into `max_direct_iosz` chunks. This parameter defines how much memory an I/O request can lock at once, so it should not be set to more than 20 percent of memory.

`max_diskq`
Limits the maximum disk queue generated by a single file. When the file system is flushing data for a file and the number of pages being flushed exceeds `max_diskq`, processes will block until the amount of data being flushed decreases. Although this doesn't limit the actual disk queue, it prevents flushing processes from making the system unresponsive. The default value is 1MB.

`max_seqio_extent_size`

Increases or decreases the maximum size of an extent. When the file system is following its default allocation policy for sequential writes to a file, it allocates an initial extent that is large enough for the first write to the file. When additional extents are allocated, they are progressively larger (the algorithm tries to double the size of the file with each new extent) so each extent can hold several writes' worth of data. This is done to reduce the total number of extents in anticipation of continued sequential writes. When the file stops being written, any unused space is freed for other files to use. Normally, this allocation stops increasing the size of extents at 2048 blocks, which prevents one file from holding too much unused space. `max_seqio_extent_size` is measured in file system blocks.

`qio_cache_enable`

Enables or disables caching on Quick I/O files. The default behavior is to disable caching. To enable caching, set `qio_cache_enable` to 1. On systems with large memories, the database cannot always use all of the memory as a cache. By enabling file system caching as a second level cache, performance may be improved. If the database is performing sequential scans of tables, the scans may run faster by enabling file system caching so the file system will perform aggressive read-ahead on the files.

`write_throttle`

The `write_throttle` parameter is useful in special situations where a computer system has a combination of a lot of memory and slow storage devices. In this configuration, sync operations (such as `fsync()`) may take so long to complete that the system appears to hang. This behavior occurs because the file system is creating *dirty pages* (in-memory updates) faster than they can be asynchronously flushed to disk without slowing system performance.

Lowering the value of `write_throttle` limits the number of dirty pages per file that a file system will generate before flushing the pages to disk. After the number of dirty pages for a file reaches the `write_throttle` threshold, the file system starts flushing pages to disk even if free memory is still available. The default value of `write_throttle` typically generates a lot of dirty pages, but maintains fast user writes. Depending on the speed of the storage device, if you lower `write_throttle`, user write performance may suffer, but the number of dirty pages is limited, so sync operations will complete much faster.

Because lowering `write_throttle` can delay write requests (for example, lowering `write_throttle` may increase the file disk queue to the `max_diskq` value, delaying user writes until the disk queue decreases), it is recommended that you avoid changing the value of `write_throttle` unless your system has a a large amount of physical memory and slow storage devices.

If the file system is being used with VxVM, it is recommended that you set the VxFS I/O parameters to default values based on the volume geometry.

If the file system is being used with a hardware disk array or volume manager other than VxVM, align the parameters to match the geometry of the logical disk. With striping or RAID-5, it is common to set `read_pref_io` to the stripe unit size and `read_nstream` to the number of columns in the stripe. For striping arrays, use the same values for `write_pref_io` and `write_nstream`, but for RAID-5 arrays, set `write_pref_io` to the full stripe size and `write_nstream` to 1.

For an application to do efficient disk I/O, it should issue read requests that are equal to the product of `read_nstream` multiplied by `read_pref_io`. Generally, any multiple or factor of `read_nstream` multiplied by `read_pref_io` should be a good size for performance. For writing, the same rule of thumb applies to the `write_pref_io` and `write_nstream` parameters. When tuning a file system, the best thing to do is try out the tuning parameters under a real-life workload.

If an application is doing sequential I/O to large files, it should issue requests larger than the discovered_direct_iosz. This causes the I/O requests to be performed as discovered direct I/O requests, which are unbuffered like direct I/O but do not require synchronous inode updates when extending the file. If the file is too large to fit in the cache, then using unbuffered I/O avoids throwing useful data out of the cache and lessons CPU overhead.

Obtaining File I/O Statistics using the Quick I/O Interface

The qiostat command provides access to activity information on Quick I/O files on VxFS file systems. The command reports statistics on the activity levels of files from the time the files are first opened using their Quick I/O interface. The accumulated qiostat statistics are reset once the last open reference to the Quick I/O file is closed.

The qiostat command displays the following I/O statistics:

- Number of read and write operations
- Number of data blocks (sectors) transferred
- Average time spent on read and write operations

When Cached Quick I/O is used, qiostat also displays the caching statistics when the -l (the long format) option is selected.

The following is an example of qiostat output:

```
                 OPERATIONS        FILE BLOCKS       AVG TIME(ms)
FILENAME       READ     WRITE     READ     WRITE      READ     WRITE
/db01/file1       0         0        0         0       0.0       0.0
/db01/file2       0         0        0         0       0.0       0.0
/db01/file3   73017    181735   718528   1114227      26.8      27.9
/db01/file4   13197     20252   105569    162009      25.8     397.0
/db01/file5       0         0        0         0       0.0       0.0
```

For detailed information on available options, see the qiostat(1M) manual page.

Using I/O Statistics Data

Once you gather the file I/O performance data, you can use it to adjust the system configuration to make the most efficient use of system resources. There are three primary statistics to consider:

- file I/O activity
- volume I/O activity

- raw disk I/O activity

If your database is using one file system on a striped volume, you may only need to pay attention to the file I/O activity statistics. If you have more than one file system, you may need to monitor volume I/O activity as well.

First, use the `qiostat -r` command to clear all existing statistics. After clearing the statistics, let the database run for a while during a typical database workload period. For example, if you are monitoring a database with many users, let the statistics accumulate for a few hours during prime working time before displaying the accumulated I/O statistics.

To display active file I/O statistics, use the `qiostat` command and specify an interval (using `-i`) for displaying the statistics for a period of time. This command displays a list of statistics such as:

```
                OPERATIONS         FILE BLOCKS        AVG TIME(ms)
FILENAME      READ    WRITE      READ    WRITE      READ    WRITE
/db01/cust1    218       36       872      144      22.8     55.6
/db01/hist1      0        1         0        4       0.0     10.0
/db01/nord1     10       14        40       56      21.0     75.0
/db01/ord1      19       16        76       64      17.4     56.2
/db01/ordl1    189       41       756      164      21.1     50.0
/db01/roll1      0       50         0      200       0.0     49.0
/db01/stk1    1614      238      6456      952      19.3     46.5
/db01/sys1       0        0         0        0       0.0      0.0
/db01/temp1      0        0         0        0       0.0      0.0
/db01/ware1      3       14        12       56      23.3     44.3
/logs/log1       0        0         0        0       0.0      0.0
/logs/log2       0      217         0     2255       0.0      6.8
```

File I/O statistics help identify files with an unusually large number of operations or excessive read or write times. When this happens, try moving the “hot” files or busy file systems to different disks or changing the layout to balance the I/O load.

Obtaining File I/O Statistics using VERITAS Extension for Oracle Disk Manager

The `odmstat` command provides access to activity information on Oracle Disk Manager files on VxFS systems. Refer to the `odmstat`(1M) manual page for more information. The command reports statistics on the activity from the time that the files were opened by the Oracle Disk Manager interface. The command has an option for zeroing the statistics. When the file is closed, the statistics are discarded.

The odmstat command displays the following I/O statistics:

- Number of read and write operations
- Number of data blocks read and written
- Average time spent on read and write operations

The following is an example of odmstat output:

```
# /opt/VRTS/bin/odmstat -i 5 /mnt/odmfile*
   OPERATIONS          FILE BLOCKS    AVG TIME(ms)
FILE NAME             READ     WRITE      READ     WRITE READ  WRITE

Mon May 11 16:21:10 2015
/db/cust.dbf            0         0         0    0         0.0   0.0
/db/system.dbf          0         0         0    0         0.0   0.0
Mon May 11 16:21:15 2015
/db/cust.dbf          371        0      371      0         0.2   0.0
/db/system.dbf          0      371         0    371        0.0   5.7

Mon May 11 16:21:20 2015
/db/cust.dbf          813        0      813      0         0.3   0.0
/db/system.dbf          0      813       0      813        0.0   5.5

Mon May 11 16:21:25 2015
/db/cust.dbf          816         0     816      0         0.3   0.0
/db/system.dbf          0       816         0   816        0.0   5.3

Mon May 11 16:21:30 2015
/db/cust.dbf            0         0         0    0         0.0   0.0
/db/system.dbf          0         0         0    0         0.0   0.0
```

Interpreting I/O Statistics

When running your database through the file system, the read-write lock on each file allows only one active write per file. When you look at the disk statistics using iostat, the disk reports queueing time and service time. The service time is the time that I/O spends on the disk, and the queueing time is how long it waits for all of the other I/Os ahead of it. At the volume level or the file system level, there is no queueing, so vxstat and qiostat do not show queueing time.

For example, if you send 100 I/Os at the same time and each takes 10 milliseconds, the disk reports an average of 10 milliseconds of service and 490 milliseconds of queueing time. The vxstat, odmstat, and qiostat report an average of 500 milliseconds service time.

Tuning Oracle Databases

To achieve optimal performance on your Oracle database, the database needs to be tuned to work with VxFS. This section describes some of the Oracle parameters that you can tune to improve your Oracle database performance when using Quick I/O.

Sequential Table Scans

Quick I/O in its default mode performs all I/O as direct I/O. In the case of single-threaded sequential scans (common in decision support system (DSS) workloads), using buffered reads can yield better performance. Because the file system detects these sequential reads and performs read-aheads, the next few blocks that Oracle requests are readily available in the system buffer cache and are simply copied to the Oracle system global area (SGA). Because access from memory is inherently faster than access from disk, this achieves a significant reduction in response time.

To handle large sequential scans when using Quick I/O, one of two methods is available to improve performance:

- Use the Oracle Parallel Query Option to break the single large scan into multiple smaller scans.

Note Consult the Oracle documentation for your system and version of Oracle, and use the settings recommended for these parameters when provided.

- The second method is to enable Cached Quick I/O for the files that would be read by the Oracle sequential scan process. Cached Quick I/O enables buffered reads, and the automatic file system read-ahead helps lower response times by pre-loading data.

Note Do not use this option if you are using a 64-bit version of Oracle.

Asynchronous I/O

Quick I/O and Oracle Disk Manager support kernel asynchronous I/O, which reduces CPU utilization and improves transaction throughput. Enabling the following parameters lets Oracle take advantage of asynchronous I/O and avoids having to configure multiple `DBWR` slaves:

- If you are using Quick I/O datafiles with Oracle8*i*, set `DISK_ASYNCH_IO` to `TRUE` in `init.ora`.
- If you are using ODM on Oracle9*i*, you do not need to change any `init.ora` parameters.

Your Oracle Installation Guide provides detailed instructions on implementing asynchronous I/O on your system.

Tuning Buffer Cache

The UNIX buffer cache plays an important role in performance when using UFS in buffered I/O mode. When using Quick I/O, however, the database cache must be tuned as if raw devices are being used. You can allocate more memory to the database buffer cache because Quick I/O bypasses the file system cache to improve database performance. Memory pages normally allocated to the file system cache can be allocated to the database buffer cache (SGA). With Oracle9i, you can adjust the SGA size without shutting down the database.

Setting Oracle Block Reads During Sequential Scans

The `DB_FILE_MULTIBLOCK_READ_COUNT` parameter specifies the maximum number of blocks Oracle reads in one I/O operation during a sequential scan. The `/etc/system` tunable parameter `maxphys` establishes the maximum physical I/O transfer size at the operating system level. To take advantage of the maximum transfer size, the Oracle `init.ora` parameter `DB_FILE_MULTIBLOCK_READ_COUNT` should be set to `maxphys/DB_BLOCK_SIZE`.

Example

If `maxphys` is set to 1048576 and `DB_BLOCK_SIZE` is set to 4096, then `DB_FILE_MULTIBLOCK_READ_COUNT` should be set to 256.

Determining I/O Buffer Size (Oracle8 and Oracle8i only)

The `DB_FILE_DIRECT_IO_COUNT` parameter specifies the number of blocks used for I/O operations during backup, restore, or direct path reads and writes. The I/O buffer size is `DB_FILE_DIRECT_IO_COUNT` * `DB_BLOCK_SIZE`.

The range of the I/O buffer size is operating system-dependent and cannot exceed `max_io_size` for your platform. Increasing the `DB_FILE_DIRECT_IO_COUNT` parameter increases PGA or SGA memory use.   Check your current setting for this parameter in the Oracle `V$PARAMETER` table.

Setting Slave Parameters

Quick I/O and ODM (Oracle9*i* only) provide support for kernel asynchronous I/O, eliminating the need for multiple logwriter slaves or database writer slaves. This parameter is set to `0` by default.

It is not necessary to set the DBWR_IO_SLAVES settings if you are using Quick I/O. The number of `DBWR` writer processes is set within DB_WRITER_PROCESSES, which performs asynchronous I/O.

Configuring Memory Allocation

Never configure Oracle to make use of more memory than is physically available on the system. Oracle may have to compete with other processes for system memory resources, and all of these potential processes must be considered when sizing and allocating memory. In the ideal configuration, a system that is dedicated to Oracle simplifies the tuning and monitoring issues and ensures best performance.

Tuning Solaris

To achieve optimal performance using VERITAS Storage Foundation *for Oracle*, certain Solaris parameters need to be tuned. Changing these parameters requires modifying the Solaris kernel settings (specified in the `/etc/system` file) and rebooting the system.

The rest of this section describes the important tuning parameters that Oracle depends on for optimal performance. You can add or change these tuning parameters in the `/etc/system` file using a text editor. The following example shows the contents of an `/etc/system` file:

```
* start Oracle *
set shmsys:shminfo_shmmax=0xffffffff
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=200
*
set semsys:seminfo_semmap=100
set semsys:seminfo_semmni=1000
set semsys:seminfo_semmns=4000
set semsys:seminfo_semmnu=800
set semsys:seminfo_semmsl=512
* end Oracle *
```

Note The settings for all tunable parameters depend on such factors as the size of your system and database, the database load, and the number of users. In some cases, we make suggestions for setting the parameters; however, you should always consult the Oracle Installation Guide for your system and version, and use the settings recommended by Oracle when provided.

maxuprc

This parameter sets the maximum number of processes that can be run concurrently by any one user. If you anticipate having a large number of users accessing Oracle concurrently, you may need to increase this parameter.

1. Check the current setting for `maxuprc` as follows:

   ```
   # echo “maxuprc/D” | adb -k
   ```

2. Modify or add the `maxuprc` setting in the `/etc/system` file as follows:

   ```
   # set maxuprc=some_integer
   ```

shmmax

This parameter sets the maximum size (in bytes) of a single shared memory segment. See your Oracle documentation for the recommended value.

shmmin

This parameter sets the minimum size (in bytes) of a single shared memory segment. See your Oracle documentation for the recommended value.

shmmni

This parameter sets the number of shared memory identifiers. See your Oracle documentation for the recommended value.

shmseg

This parameter sets the maximum number of shared memory segments that can be attached by a process. See your Oracle documentation for the recommended value.

semmap

This parameter sets the number of entries in semaphore map. The memory space given to the creation of semaphores is taken from `semmap`, which is initialized with a fixed number of map entries based on the value of `semmap`. The value of `semmap` should never be larger than `semmni`. See your Oracle documentation for the recommended value.

semmni

This parameter sets the number of semaphore set identifiers in the system. The `semmni` parameter determines the number of semaphore sets that can be created at any one time, and may need to be set higher for a large database. See your Oracle documentation for the recommended value.

semmns

This parameter sets the maximum number of semaphores in the system. The `semmns` parameter may need to be set higher for a large database. See your Oracle documentation for the recommended value.

semmnu

This parameter sets the system-wide maximum number of undo structures. Setting this parameter value equal to `semmni` provides for an undo structure for every semaphore set. Semaphore operations performed using `semop`(2) can be undone if the process terminates, but an undo structure is required to guarantee it. See your Oracle documentation for the recommended value of `semmnu`.

semmsl

This parameter sets the maximum number of semaphores that can be in one semaphore set. The `semmsl` parameter should be equal to the maximum number of Oracle processes. See your Oracle documentation for the recommended value.

VERITAS Storage Foundation for Oracle Command Line Interface

A

VERITAS Storage Foundation *for Oracle* provides a command line interface (CLI) to many key operations also supplied from within the VxDBA utility menus and VERITAS Storage Foundation GUI application. The command line interface lets you incorporate command operations into scripts and other administrative processes.

VERITAS Storage Foundation *for Oracle* also provides commands specific to the Database FlashSnap feature. These commands can be executed using the CLI or the VERITAS Storage Foundation *for Oracle* GUI. At this time, there are no VxDBA menu equivalents for Database FlashSnap operations.

Note The VERITAS Storage Foundation *for Oracle* command line interface depends on certain tablespace and datafile information that is collected and stored in VxDBA's repository. Some CLI commands update the repository by default. It is also important to regularly ensure the VxDBA repository is up-to-date by using **Display/Update Tablespace Information** from either the **Database Administration** or **Display Database/VxDBA Information** VxDBA submenus, the update repository option from the GUI (right click on the database and select **Update Repository**), or the `dbed_update` command.

Topics include:

- "Overview of Commands" on page 426
- "Examples of Using the Command Line Interface" on page 431

Overview of Commands

All VERITAS Storage Foundation *for Oracle* commands supported in the command line interface are located in the `/opt/VRTS/bin` directory. Exceptions are the Quick I/O commands, `vxstorage_stats`, and `edgetmsg2`, which are located in `/opt/VRTSdbed/bin.` Online manual pages are located in the `/opt/VRTS/man` directory. Follow the installation instructions provided in the *VERITAS Database Edition for Oracle Installation Guide* to ensure you can use these commands and view the online manual pages.

The following table summarizes the commands available to you from the command line:

VERITAS Storage Foundation for Oracle Commands

Command	Description
`dbed_update`	Creates or updates the VxDBA repository in VERITAS Storage Foundation *for Oracle*. Performs the same operation from the command line, as **Display/Update Tablespace Information** from the **Display Database/VxDBA Information** in the VxDBA utility menus. Performs the same operation from the command line, as right clicking on a database and selecting **Resync Repository** in the GUI.
`dbed_checkconfig`	Checks the configuration of an Oracle database in a VERITAS Storage Foundation *for Oracle* environment. Performs the same operation from the command line, as **Display Database/VxDBA Information** > **Examine Volume/File System/Database Configuration** in the VxDBA utility menus. Performs the same operation from the command line, as right clicking on a database and selecting **Check System Configuration** in the GUI.
`dbed_saveconfig`	Saves the configuration of an Oracle database in a VERITAS Storage Foundation *for Oracle* environment. Performs the same operation from the command line, as **Display Database/VxDBA Information** > **Save Volume/File System/Database Configuration** in the VxDBA utility menus. Performs the same operation from the command line, as right clicking on a database and selecting **Save System Configuration** in the GUI.

VERITAS Storage Foundation for Oracle Commands

Command	Description
`dbed_ckptcreate`	Creates a Storage Checkpoint for an Oracle database. Performs the same operation from the command line, as **Storage Checkpoint Administration** > **Create New Storage Checkpoint** in the VxDBA utility menus. Performs the same operation from the command line, as **Storage Checkpoints** > **Create Storage Checkpoint** in the GUI.
`dbed_ckptdisplay`	Displays Storage Checkpoints for an Oracle database. Performs the same operation from the command line, as **Storage Checkpoint Administration** > **Display Storage Checkpoints** in the VxDBA utility menus. Performs the same operation from the command line, as **Storage Checkpoint** > **Properties** in the GUI. (Clicking on **Storage Checkpoint** alone lists all Storage Checkpoints.)
`dbed_ckptmount`	Mounts a Storage Checkpoint for an Oracle database. Performs the same operation from the command line, as **Storage Checkpoint Administration** > **Mount Storage Checkpoint** in the VxDBA utility menus. Performs the same operation from the command line, as **Storage Checkpoint** > **Mount Storage Checkpoint** in the GUI.
`dbed_ckptplan`	Used to obtain estimates of space usage for Storage Checkpoints. `dbed_ckptplan` manages scheduled Storage Checkpoint creation and summarizes statistics from these Storage Checkpoints. Performs the same operation from the command line, as **Storage Checkpoint Capacity Planning** > **Create Capacity Planning Schedules** in the VxDBA utility menu. Performs the same operation from the command line, as **Oracle** > **Storage Checkpoint Capacity Planning** in the GUI.
`dbed_ckptpolicy`	Creates and administers Storage Checkpoint allocation policies for a Multi-Volume File System (MVS). You can display, create, update, assign, and remove Storage Checkpoint allocation policies using this command. This option is not available through the VxDBA utility menu or the GUI.
`dbed_ckptquota`	Administers quotas for Storage Checkpoints. This option is not available through the VxDBA utility menu or the GUI.

VERITAS Storage Foundation for Oracle Commands

Command	Description
`dbed_ckptremove`	Removes a Storage Checkpoint for an Oracle database. Performs the same operation from the command line, as **Storage Checkpoint Administration** > **Remove Storage Checkpoint** in the VxDBA utility menus. Performs the same operation from the command line, as **Storage Checkpoint** > **Remove Storage Checkpoint** in the GUI.
`dbed_ckptrollback`	Rolls back an Oracle database to a Storage Checkpoint point-in-time image. Performs the same Storage Rollback operation from the command line, as the database Storage Rollback operation available in the VxDBA utility **Storage Rollback Administration** menu. Performs the same operation from the command line, as **Storage Checkpoint** > **Rollback Storage Checkpoint** in the GUI.
`dbed_ckptumount`	Unmounts a Storage Checkpoint for an Oracle database. Performs the same operation from the command line, as **Storage Checkpoint Administration** > **Unmount Storage Checkpoint** in the VxDBA utility menus. Performs the same operation from the command line, as **Storage Checkpoint** > **Unmount Storage Checkpoint** in the GUI.
`dbed_clonedb`	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. Performs the same operation from the command line, as **Oracle** > **Create Clone Database** in the GUI.
`qio_convertdbfiles`	Converts VxFS files to Quick I/O files. Performs the same operation from the command line, as **Container** > **Conversion** in the GUI.
`qio_getdbfiles`	Extracts information on files used by the database and stores the names of these files in `mkqio.dat`. The `mkqio.dat` file is used by the `qio_convertdbfiles` command. This command is not available through the GUI.

VERITAS Storage Foundation for Oracle Commands

Command	Description
`qio_recreate`	Automatically recreates Quick I/O files when the database is recovered. The command expects to find a mkqio.dat file in the directory where the `qio_recreate` command is run. This command is not available through the GUI.
`dbed_vmchecksnap`	Creates and validates a snapplan that the `dbed_vmsnap` command uses to create a volume snapshot of an Oracle database. The snapplan specifies snapshot scenarios (such as online, offline, or instant). The command can also be used to validate a snapplan. This option is not available through the VxDBA utility menu. Performs the same operation from the command line, as the options within **Snapplans > Create Snapplan...** and **Snapplan > Modify/Validate Snapplan** in the GUI.
`dbed_vmsnap`	Creates a snapshot image of an Oracle database by splitting the mirror volumes used by the database. You can also use this command to resynchronize the snapshot image back to the current database. The command also allows you to reverse resynchronize a snapshot image of an Oracle database. This option is not available through the VxDBA utility menu. Performs the same operation from the command line, as the options within **Snapplan > Create Snapshot** in the GUI.
`dbed_vmclonedb`	Mounts the file systems on the snapshot volumes and starts a clone database from snapshot volumes. You can also use this command to shut down or restart the clone database, unmount the file systems, or deport the clone database's volumes. This option is not available through the VxDBA utility menu. Performs the same operation from the command line, as the options within **Oracle > Start Up Snapshot Database** in the GUI.
`dbed_analyzer`	Maps tablespaces to physical disks and retrieves the information, including the percentage of disk space used by a tablespace. This option is not available through the VxDBA utility menu or through the GUI.

VERITAS Storage Foundation for Oracle Commands

Command	Description
`edgetmsg2`	Manages message log files. You can use this utility to display and list message log files. You can also use this utility to write a message to a log file or to the console, or read the log file and print to the console. This option is not available through the VxDBA utility menu or through the GUI.
`vxstorage_stats`	Displays storage object I/O statistics. This option is not available through the VxDBA utility menu. Performs the same operation from the command line, as the options within **Datafiles > Topology Statistics...** in the GUI.

Examples of Using the Command Line Interface

This section provides examples for using the VERITAS Storage Foundation *for Oracle* 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.

Creating or Updating VxDBA's Repository Using dbed_update

You can use the VERITAS Storage Foundation *for Oracle* `dbed_update` command to create or update the repository for VxDBA.

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`.

Prerequisites

- 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 VERITAS Storage Foundation *for Oracle* 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 VxDBA repository

Use the `dbed_update` command as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/9i
```

▼ To view the status of the VxDBA repository

Use the `dbed_update` command with the `-n` option as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/9i -n
VxDBA repository is up to date.
```

Checking Oracle Configuration Environment Using dbed_checkconfig

You can use the VERITAS Storage Foundation *for Oracle* dbed_checkconfig command to verify and report on an Oracle environment from the command line.

Prerequisites

- You must be logged on as the database administrator (typically, the user ID oracle).

Usage Notes

- The dbed_checkconfig command is used to verify various elements of the Oracle database environment. The utility attempts to use certain basic rules on Oracle settings, file system and volume parameters and layout options to verify how resilient and well configured a configuration is. The information provided is valid for the supplied Oracle database (SID).

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

▼ To check the Oracle configuration environment

Use the dbed_checkconfig command as follows:

```
$ /opt/VRTS/bin/dbed_checkconfig -S PROD -H /oracle/product/9i
Examining file system attributes.

All file systems are VxFS.
All file systems are VxFS Version 4 or higher layout.
All file systems have the same version layout (version 6).

Examining Quick I/O settings.

All datafiles are Quick I/O files.

Examining Cached Quick I/O settings.

No file systems have Cached Quick I/O enabled.

Examining datafiles fragmentation.

5 files are fragmented.

Examining File System tunable settings.

Parameters for all VxFS file systems used by PROD.
```

```
Filesystem i/o parameters for /prod_db
read_pref_io = 65536
read_nstream = 1
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 1
write_unit_io = 65536
pref_strength = 10
buf_breakup_size = 1048576
discovered_direct_iosz = 262144
max_direct_iosz = 1048576
default_indir_size = 8192
qio_cache_enable = 0
write_throttle = 0
max_diskq = 1048576
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 = 65075200
fcl_keeptime = 0
fcl_winterval = 3600

Examining Oracle volume and file system layout.

Data for database PROD is contained in one volume group.

Examining Oracle internal information.

Oracle Version is 9.2.0.4.0.


Control file /prod_db/control1 is on file system /prod_db.


Control file /prod_db/control2 is on file system /prod_db.

Total of 2 control files over 1 file systems.

SFORA dbed_checkconfig WARNING V-81-3122: Control files are not
spread over multiple file
systems. Spread control files over multiple file systems
for better redundancy.
```

```
Examining Oracle automatic extension of datafiles.

Total of 0 datafiles are configured to auto extend.
Total of 9 datafiles are defined to the database.

Examining Oracle log modes.

The database is running in archivelog mode.

The database is running in automatic log archiving mode.
```

▼ To check the Oracle configuration environment and not update the VxDBA repository

Use the `dbed_checkconfig` command with the `-v` and `-n` options as follows:

```
$ /opt/VRTS/bin/dbed_checkconfig -S PROD -H /oracle/product/9i -v -n
```

where `-v` lists all files and `-n` stops the VxDBA repository from being updated.

```
Examining file system attributes.

All file systems are VxFS.
All file systems are VxFS Version 4 or higher layout.
All file systems have the same version layout (version 6).

Examining Quick I/O settings.

All datafiles are Quick I/O files.

Examining Cached Quick I/O settings.

No file systems have Cached Quick I/O enabled.

Examining datafiles fragmentation.

5 files are fragmented.
/prod_db/index02.dbf is fragmented.
/prod_db/rolbak1.dbf is highly fragmented.
/prod_db/system.dbf is highly fragmented.
/prod_db/data01.dbf is highly fragmented.
/prod_db/data02.dbf is highly fragmented.

Examining File System tunable settings.

Parameters for all VxFS file systems used by PROD.
Filesystem i/o parameters for /prod_db
read_pref_io = 65536
```

```
read_nstream = 1
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 1
write_unit_io = 65536
pref_strength = 10
buf_breakup_size = 1048576
discovered_direct_iosz = 262144
max_direct_iosz = 1048576
default_indir_size = 8192
qio_cache_enable = 0
write_throttle = 0
max_diskq = 1048576
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 = 65075200
fcl_keeptime = 0
fcl_winterval = 3600

Examining Oracle volume and file system layout.

Data for database PROD is contained in one volume group.

Examining Oracle internal information.

Oracle Version is 9.2.0.4.0.


Control file /prod_db/control1 is on file system /prod_db.


Control file /prod_db/control2 is on file system /prod_db.

Total of 2 control files over 1 file systems.

SFORA dbed_checkconfig WARNING V-81-3122: Control files are not
spread over multiple file
systems. Spread control files over multiple file systems
for better redundancy.

Examining Oracle automatic extension of datafiles.
```

```
Total of 0 datafiles are configured to auto extend.
The following datafiles are not configured to autoextend:
/prod_db/deflt.dbf
/prod_db/temp.dbf
/prod_db/index02.dbf
/prod_db/index01.dbf
/prod_db/data1.dbf
/prod_db/rolbak1.dbf
/prod_db/system.dbf
/prod_db/data01.dbf
/prod_db/data02.dbf

Total of 9 datafiles are defined to the database.

Examining Oracle log modes.

The database is running in archivelog mode.

The database is running in automatic log archiving mode.
```

Saving the Oracle Configuration Environment Using dbed_saveconfig

You can use the VERITAS Storage Foundation *for Oracle* dbed_saveconfig command to save configuration information on Oracle, VERITAS products, and system hardware from the command line.

Prerequisites

- You must be logged on as the database administrator (typically, the user ID oracle).

Usage Notes

- The dbed_saveconfig command is used to collect and record configuration information on Oracle, VERITAS products, and system hardware. Information is gathered in the context of a specified Oracle database. The utility attempts to gather enough information to allow an administrator to reconstruct a system and database from scratch, in the case of a complete system failure.

  Information collected is in the form of many system configuration files and the results of querying the system hardware, VERITAS products, and Oracle. The location where configuration information has been saved is displayed as output from the dbed_saveconfig command. Alternatively, you can use the -l option to designate this location.
- See the dbed_saveconfig(1M) manual page for more information.

▼ To save the Oracle configuration environment

Use the dbed_saveconfig command as follows:

```
$ /opt/VRTS/bin/dbed_saveconfig -S PROD -H /oracle/product/9i
Saving System and Oracle Information, please wait ...

System configuration information saved to directory:
/tmp/vxdba.DR.1148
```

▼ To save the Oracle configuration environment without updating the VxDBA repository

Use the dbed_saveconfig command with the -n option as follows:

```
$ dbed_saveconfig -S PROD -H /oracle/product/9i -n
Saving System and Oracle Information, please wait ...

System configuration information saved to directory:
/tmp/vxdba/DR.1149
```

Creating Storage Checkpoints Using dbed_ckptcreate

You can use the VERITAS Storage Foundation *for Oracle* `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.

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 under the following directory:

  ```
  /etc/vx/vxdba/$ORACLE_SID/checkpoint_dir
  ```

- 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/9i -o online
Creating online Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_971672042 created.
```

▼ To create Storage Checkpoints without updating the VxDBA repository while the database is online

Use the `dbed_ckptcreate` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD \
  -H /oracle/product/9i -o online -n
Creating online Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_971672043 created.
```

▼ To create Storage Checkpoints while the database is offline

1. Use the dbed_ckptcreate command as follows:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD \
-H /oracle/product/9i -o offline
Creating offline Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_971672044 created.
```

Note The default option is online.

▼ To create an 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/9i -o instant
Creating instant Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_971672045 created.
```

▼ 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/9i -o online -p ckpt_data,ckpt_metadata
Creating online Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_971672044 created.
```

Displaying Storage Checkpoints Using dbed_ckptdisplay

You can use the VERITAS Storage Foundation *for Oracle* 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.

Prerequisites

- You may be logged in as either the database administrator or root.

Usage Notes

- In addition to displaying the Storage Checkpoints created by VxDBA, dbed_ckptdisplay also displays other Storage Checkpoints (for example, Storage Checkpoints created by the Capacity Planning Utility and NetBackup).
- See the dbed_ckptdisplay(1M) manual page for more information.
- 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. For details, see "Mounting the Snapshot Volumes and Backing Up" on page 242.

▼ To display Storage Checkpoints created by VERITAS Storage Foundation for Oracle

Use the dbed_ckptdisplay command as follows to display information for Storage Checkpoints created by VERITAS Storage Foundation for *Oracle*:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
  -H /oracle/product/9i
Checkpoint_975876659          Sun Apr 3 12:50:59 2004    P+R+IN
Checkpoint_974424522_wr001    Thu May 16 17:28:42 2004   C+R+ON
Checkpoint_974424522          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/9i -o other
NetBackup_incr_PROD_955133480    NBU      /db01
NetBackup_full_PROD_955132952    NBU      /db01
```

▼ To display other Storage Checkpoints without updating the VxDBA repository

Use the dbed_ckptdisplay command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
  -H /oracle/product/9i -o other -n
NetBackup_incr_PROD_955133480    NBU       /db01
NetBackup_full_PROD_955132952    NBU       /db01
```

▼ To display all Storage Checkpoints

Use the dbed_ckptdisplay command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
  -H /oracle/product/9i -o all
Checkpoint_971672042             Sun May 15 13:55:53 2004   C+R+IN
Checkpoint_903937870             Fri May 13 22:51:10 2004   C+R+ON
Checkpoint_901426272             Wed May 11 16:17:52 2004   P+R+ON
NetBackup_incr_PROD_955133480    NBU       /db01
NetBackup_full_PROD_955132952    NBU       /db01
```

▼ To display all Storage Checkpoints without updating the VxDBA repository

Use the dbed_ckptdisplay command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
  -H /oracle/product/9i -o all -n
Checkpoint_971672042             Sun May 15 13:55:53 2004   C+R+IN
Checkpoint_903937870             Fri May 13 22:51:10 2004   C+R+ON
Checkpoint_901426272             Wed May 11 16:17:52 2004   P+R+ON
NetBackup_incr_PROD_955133480    NBU       /db01
NetBackup_full_PROD_955132952    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 2003      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 VERITAS Storage Foundation *for Oracle* dbed_ckptcreate command to schedule Storage Checkpoint creation in a cron job or other administrative script.

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

- 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/9i -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/9i -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/9i -o offline
```

Mounting Storage Checkpoints Using dbed_ckptmount

You can use the VERITAS Storage Foundation *for Oracle* dbed_ckptmount command to mount a Storage Checkpoint for an Oracle database from the command line.

Prerequisites

- You may be logged in as either the database administrator or root.

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, _wr*xxx*, 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 DBA user 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. For details, refer to "Mounting the Snapshot Volumes and Backing Up" on page 242.
- 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 VERITAS Storage Foundation *for Oracle* `dbed_ckptumount` command to unmount a Storage Checkpoint for an Oracle database from the command line.

Prerequisites

- You may be logged in as either the database administrator or `root`.

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_wr001
```

Creating and Working with Storage Checkpoint Allocation Policies Using dbed_ckptpolicy

You can use the VERITAS Storage Foundation *for Oracle* `dbed_ckptpolicy` command to create and administer Storage Checkpoint allocation policies for Multi-Volume File Systems (MVSs). Storage Checkpoint allocation policies specify a list of volumes and the order in which to allocate data to them.

Prerequisites

- You must be logged on as the database administrator (typically, the user ID `oracle`).

Usage Notes

- All volumes must be MVSs.
- The `dbed_ckptpolicy` command can be used only on file systems using disk layout Version 6.
- The VxVM volume set and VxFS Multi-Volume File System features must be enabled to use Storage Checkpoint allocation policies.
- The status of a Storage Checkpoint allocation policy is either `partial` or `complete`. A *partial policy* is one that does not exist on all file systems used by the database. A *complete policy* is one that exists on all file systems.
- After an allocation policy is assigned to a Storage Checkpoint, the allocation mechanism attempts to satisfy requests from each device in the order specified in the allocation policy. If the request cannot be satisfied from any of the devices in the allocation policy, the request will fail, even if other devices that have space exist in the file system. Only devices listed in the policy can be allocated.
- See the `dbed_ckptpolicy`(1M) manual page for more information.

▼ To create a Storage Checkpoint allocation policy

Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
  -o create -p ckpt_policy
```

Output similar to the following is displayed. This example assumes the following:

- Two MVS file systems `/mvsfs/v1` and `/mvsfs/v2` are used for datafiles.
- File system `/mvsfs/v1` is created on volume set mvsvset1.
- File system `/mvsfs/v2` is created on volume set mvsvset2.
- Volume set `mvsvset1` contains volumes `mvsv1`, `mvsv2`, and `mvsv3`.

- Volume set `mvsvset2` contains volumes `mvsv4` and `mvsv5`.

```
File System: /mvsfs/v2 (MVS volumes: mvsv4,mvsv5)
Assigned Data Policy: NONE (MVS Volumes: N/A)
Assigned Meta Data Policy: NONE (MVS Volumes: N/A)
Please enter the volume name(s), sperated by space, for the policy
ckpt_policy [skip,quit]: mvsv4

File System: /mvsfs/v1 (MVS volumes: mvsv1,mvsv2,mvsv3)
Assigned Data Policy: NONE (MVS Volumes: N/A)
Assigned Meta Data Policy: NONE (MVS Volumes: N/A)
Please enter the volume name(s), separated by space, for the policy
ckpt_policy [skip,quit]: mvsv2

The following information will be used to create policy ckpt_sample
ckpt_sample            /mvsfs/v2                       mvsv4
ckpt_sample            /mvsfs/v1                       mvsv2
```

▼ To display Storage Checkpoint allocation policy within the database

Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
  -n -o display [-c storage_ckpt | -p ckpt_policy]
```

If `-p` *`ckpt_policy`* and `-c` *`storage_ckpt`* options are not specified, output similar to the following is displayed:

```
Policy Name          File System Coverage
-------------------- --------------------
 ckpt                Complete
 ckpt_data           Complete
 ckpt_metadata       Complete
 new_ckpt            Partial
 ckpt_sample         Complete
```

If `-p` *`ckpt_policy`* option is specified, output similar to the following is displayed:

```
Policy Name      File System  MVS volumes
---------------- ------------ -----------
ckpt_sample      /mvsfs/v2    mvsv4
ckpt_sample      /mvsfs/v1    mvsv2
```

If the `-c` *`storage_ckpt`* option is specified, output similar to the following is displayed:

```
Storage Checkpoint    File System  Data Policy  Meta Data Policy
--------------------- ------------ -----------  ----------------
Checkpoint_1095125037 /mvsfs/v2    ckpt_data    ckpt_metadata
Checkpoint_1095125037 /mvsfs/v1    ckpt_data    ckpt_metadata
```

▼ To update a Storage Checkpoint allocation policy

Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
  -n -o update -p ckpt_policy
```

Output similar to the following is displayed:

```
File System: /mvsfs/v2 (MVS volumes: mvsv4,mvsv5)
Policy: ckpt_sample (MVS volumes: mvsv4)
Please enter the volume name(s), separated by space, for the policy
ckpt_sample [skip,quit]: mvsv5

File System: /mvsfs/v1 (MVS volumes: mvsv1,mvsv2,mvsv3)
Policy: ckpt_sample (MVS volumes: mvsv2)
Please enter the volume name(s), separated by space, for the policy
ckpt_sample [skip,quit]: mvsv2,mvsv3

The following information will be used to create policy ckpt_sample
ckpt_sample            /mvsfs/v2                             mvsv5
ckpt_sample           /mvsfs/v1                             mvsv2,mvsv3
```

▼ To assign a Storage Checkpoint allocation policy

Use the `dbed_ckptpolicy` command as follows to assign an allocation policy to a specified Storage Checkpoint:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
  -n -o assign -c ckpt_name -p ckpt_policy[,ckpt_metadata_policy]
```

▼ To remove a Storage Checkpoint allocation policy

Use the `dbed_ckptpolicy` command as follows to remove an allocation policy from a specified Storage Checkpoint:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
  -n -o remove -p ckpt_policy
```

Administering Storage Checkpoint Quotas dbed_ckptquota

You can use the VERITAS Storage Foundation *for Oracle* `dbed_ckptquota` command to administer file system quotas for Storage Checkpoint for an Oracle database from the command line.

Prerequisites

- You must be logged on as the database administrator (typically, the user ID `oracle`).
- The VxDBA 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
```

Note `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 VERITAS Storage Foundation *for Oracle* `dbed_ckptrollback` command to rollback an Oracle database to a Storage Checkpoint.

Prerequisites

- You may be logged in as either the database administrator or `root`.

Usage Notes

- The `dbed_ckptrollback` 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 4.1 *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/9i \
  -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/9i \
  -T DATA01 -c Checkpoint_903937870
```

Note 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.

Note 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/9i \
-F /share/oradata1/data01.dbf /share/oradata2/index01.dbf \
-c Checkpoint_903937870
```

Note 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.

Note In the case of an instant Storage Checkpoint, rolling back datafiles does not apply.

Removing Storage Checkpoints Using dbed_ckptremove

You can use the VERITAS Storage Foundation *for Oracle* `dbed_ckptremove` command to remove a Storage Checkpoint for an Oracle database at the command line.

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 VxDBA(1M) menu-driven utility, the VxDBA GUI, or the `dbed_ckptcreate`(1M) command.

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

◆ You must unmount the Storage Checkpoint before you can remove it.

▼ To remove Storage Checkpoints

Use the `dbed_ckptremove` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptremove -S PROD \
  -c Checkpoint_971672042_wr001
```

Managing Capacity Planning Utility Using dbed_ckptplan

You can use the VERITAS Storage Foundation *for Oracle* dbed_ckptplan command to manage the Storage Checkpoint Capacity Planning Utility at the command line.

Usage Notes

- The dbed_ckptplan command is used to obtain estimates on space usage for Storage Checkpoints. It obtains estimates by managing scheduled Storage Checkpoint creation and summarizing statistics from these Storage Checkpoints. You can only use the Storage Checkpoint Capacity Planning Utility in an environment that contains no Storage Checkpoints created by other tools or products.
- See the dbed_ckptplan(1M), cron(1M), and crontab(1M) manual pages for more information.

▼ To create Capacity Planning Utility schedules

Use the dbed_ckptplan command as follows:

```
$ /opt/VRTSdbed/bin/dbed_ckptplan -s
```

▼ To display Capacity Planning Utility schedules

Use the dbed_ckptplan command as follows:

```
$ /opt/VRTSdbed/bin/dbed_ckptplan -l
```

▼ To display Storage Checkpoint space usage on a VxFS file system

Use the dbed_ckptplan command as follows:

```
$ /opt/VRTSdbed/bin/dbed_ckptplan -p
```

▼ To remove Capacity Planning Utility schedules

Use the dbed_ckptplan command as follows:

```
$ /opt/VRTSdbed/bin/dbed_ckptplan -r
```

Cloning the Oracle Instance Using dbed_clonedb

You can use the VERITAS Storage Foundation *for Oracle* 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 Oracle 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 dbed_clonedb command, recovers the entire database and replays all of the archive logs. You will not be prompted for any archive log names.

Prerequisites

- 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.
- You must first create a Storage Checkpoint. (See "Creating Storage Checkpoints Using dbed_ckptcreate" on page 438.)

Usage Notes

- The dbed_clonedb command is used to create a copy of an Oracle 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 Oracle recovery process.
- See the dbed_clonedb(1M) manual page for more information.

Options

`-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.

`-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.

▼ To clone an Oracle instance with manual Oracle recovery

Use the `dbed_clonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -m /local/oracle9/1 \
  -c Checkpoint_988813047 -i
Primary Oracle SID is TEST9i
New Oracle SID is NEW9
Checkpoint_988813047 not mounted at /local/oracle9/1
Mounting Checkpoint_988813047 at /local/oracle9/1
Using environment-specified parameter file
        /local/oracle9/links/dbs/initTEST9i.ora
Default Oracle parameter file found:
        /local/oracle9/links/dbs/initTEST9i.ora
Copying /local/oracle9/links/dbs/initTEST9i.ora
        to /local/oracle9/1/testvol
Control file 'ora_control2'
        path not explicitly specified in init file; assuming
        ORACLE_HOME/dbs
```

```
All redo-log files found
Copying initTEST9i.ora to initNEW9.ora
        in /local/oracle9/1/testvol
Altering db_name in initNEW9.ora
Altering control file locations in initNEW9.ora
Creating new link for clone database init file
Creating archive log directory

About to start up new database and begin reconfiguration

Database NEW9 is being reconfigured
Altering clone database archive log directory
Updating log_archive_dest in clone database init file
Found archive log destination at /testvol

The latest archive log(s) must now be applied. To apply
        the logs, open a new window and perform the following steps:

1. copy required archive log(s) from primary to clone:
   primary archive logs in /testvol
   clone archive logs expected in /local/oracle9/1/testvol
2. ORACLE_SID=NEW9; export ORACLE_SID # sh and ksh, OR
   setenv ORACLE_SID NEW9 #csh
3. /local/oracle9/links/bin/sqlplus /nolog
4. CONNECT / AS SYSDBA
5. RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
6. enter the archive log(s) you wish to apply
7. EXIT

Press <Return> after you have completed the above steps.
<Return>

Resetting logs on new database NEW9
Database instance NEW9 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 NEW9 -m /local/oracle9/1 \
  -c Checkpoint_988813047
Primary Oracle SID is TEST9i
New Oracle SID is NEW9
Checkpoint_988813047 not mounted at /local/oracle9/1
Mounting Checkpoint_988813047 at /local/oracle9/1
Using environment-specified parameter file
        /local/oracle9/links/dbs/initTEST9i.ora
```

```
Default Oracle parameter file found:
        /local/oracle9/links/dbs/initTEST9i.ora
Copying /local/oracle9/links/dbs/initTEST9i.ora
        to /local/oracle9/1/testvol
Control file 'ora_control2'
        path not explicitly specified in init file; assuming
        ORACLE_HOME/dbs

All redo-log files found
Copying initTEST9i.ora to initNEW9.ora
        in /local/oracle9/1/testvol
Altering db_name in initNEW9.ora
Altering control file locations in initNEW9.ora
Creating new link for clone database init file
Creating archive log directory

About to start up new database and begin reconfiguration
Database NEW9 is being reconfigured
Starting automatic (full) 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 /testvol
Mounting clone database
Resetting logs on new database NEW9
Database instance NEW9 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 an Oracle 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. For more information on snapplans, see "Validating a Snapplan (dbed_vmchecksnap)" on page 230.

Note You must have the Enterprise Edition of VERITAS Storage Foundation *for Oracle* to use this command.

Snapplan Parameters

When using `dbed_vmchecksnap -o setdefaults` option to create the snapplan, the following parameters are set:

Parameter	Value
`SNAPSHOT_VERSION`	Specifies the snapshot version for this release of VERITAS Storage Foundation *for Oracle*
`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.
`PRIMARY_DG`	Specifies the name of the Volume Manager disk group used by the primary database.
`SNAPSHOT_DG`	Specifies the name of the disk group containing the snapshot volumes. The snapshot volumes will be put into this disk group on the primary host and deported if the primary and secondary hosts are different. The secondary host will import this disk group to start a clone database.
`ORACLE_SID`	The name of the Oracle database.

Parameter	Value
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**.
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**.
SNAPSHOT_MODE	**online** or **offline** or **instant** Specifies whether the database snapshot should 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. 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.

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.
SNAPSHOT_MIRROR	Specifies the number of plexes to be snapshot. The default value is 1.

Creating a Snapplan

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 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

- 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.

Options

`-S`	Specifies the `ORACLE_SID`, which is the name of the Oracle database, for which a snapshot image will be created.
`-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. See "Summary of Database Snapshot Steps" on page 217 for descriptions of the snapplan parameters.
`-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 and your local directory.
`-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. If it is not specified, `dbed_flashsnap` is used as the default plex tag.

▼ To create a snapplan and set the default values for a single host

Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/9i \
-f snap1 -o setdefaults -t host1
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_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/9i -f snap2 -o setdefaults -t host2
Snapplan snap2 for PROD.
=====================================================
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_VOL_PREFIX=SNAP_
```

```
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Validating a Snapplan

You can use the 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 on the primary host

Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/9i \
-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.
```

▼ 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/9i -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/arch.

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

▼ 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         init            yes
snap2           init_full         init            yes
snap3           init_full         init            yes
```

Note 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 explanations of the various statuses that may appear for SNAP_STATUS and DB_STATUS, refer to "VERITAS Database FlashSnap Status Information" on page 495.

To view a snapplan

Use the dbed_vmchecksnap command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o list
SNAPSHOT_VERSION=4.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODdg
SNAPSHOT_DG=SNAP_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_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
DB_STATUS=init
LOCAL_SNAPPLAN=/export/snap_dir/snap1
```

Copying or Removing a Snapplan Using dbed_vmchecksnap

▼ To copy a snapplan from the VxDBA repository to your local directory

To copy a snapplan from the VxDBA repository to your local directory, the snapplan must not already be 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 VxDBA repository.
```

Creating, Resynchronizing, or Reverse Resynchronizing a Snapshot Database Using dbed_vmsnap

You can use the VERITAS Storage Foundation *for Oracle* `dbed_vmsnap` command to create a snapshot image of an Oracle 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.

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.
- 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

`-S` *`ORACLE_SID`*	Specifies the `ORACLE_SID`, which is the name of the Oracle database, for which a snapshot image will be created.
`-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.

`-o reverse_resync_commit` Commits the reverse resynchronization changes after you have verified that they are acceptable.

`-o reverse_resync_abort` Aborts reverse resynchronization and mounts the original volumes back with the file systems that are configured to use the volume.

▼ To create a snapshot image on the primary host

Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot
dbed_vmsnap started at 2004-04-02 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 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 2004-04-02 16:19:05
The option resync has been completed.
dbed_vmsnap ended at 2004-04-02 16:19:26
```

▼ To resynchronize your database to a snapshot

Note To run this command successfully, the mount point for the primary database must be created by and owned by the Oracle DBA user before mounting the VxFS file system.

Note 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.

Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 \
-o reverse_resync_begin
dbed_vmsnap started at 2004-04-02 15:53:45
Database PROD (SID=PROD) is running.
dbed_vmsnap ended at 2004-04-02 15:54:29
```

▼ To abort resynchronizing your database to a snapshot

This option is only allowed when `reverse_resync_begin` has been run. It is not allowed if `reverse_resync_commit` has been executed.

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
```

▼ To commit reverse resynchronization changes

This option is only allowed after `reverse_resync_begin` has been run.

Caution 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. The `dbed_vmsnap -o reverse_resync_commit` command cannot be undone and should be used with extreme caution.

Use the `dbed_vmsnap` command as follows:

```
  $ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o
reverse_resync_commit
  dbed_vmsnap started at 2004-04-02 16:16:44
  The option reverse_resync_commit has been completed.
  dbed_vmsnap ended at 2004-04-02 16:18:56
```

Creating or Shutting Down a Clone Database Using dbed_vmclonedb

You can use the VERITAS Storage Foundation *for Oracle* `dbed_vmclonedb` command to create or shutdown a clone database on either the primary or secondary host using snapshot volumes from the primary 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 complete the steps in "Summary of Database Snapshot Steps" on page 217, "Validating a Snapplan (dbed_vmchecksnap)" on page 230, and "Creating a Snapshot (dbed_vmsnap)" on page 236.
- 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.
- In a single-host configuration, the primary and secondary hosts can be the same.
- In a single-host configuration, `-r` *`relocate_path`* is required.
- In a two-host configuration, the `vxdbavol=`*`vol_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. For details, see "Mounting the Snapshot Volumes and Backing Up" on page 242.
- See the `dbed_vmclonedb`(1M) manual page for more information.

Options

`-S` *`ORACLE_SID`*

Specifies the `ORACLE_SID`, which is the name of the Oracle database, for which a snapshot image will be created.

`-g` *`snap_dg`*

Specifies the name of the disk group that contains all snapshot volumes.

`-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.

`-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 VxDBA repository. This option is required only after the database has been manually recovered (`-o mountdb`).

`-o umount`

Shuts down the clone database and unmounts all snapshot files.

`new_sid=`

Specifies the new `ORACLE_SID` for the snapshot image.

This is a required argument.

`vxdbavol=`*`vol_name`*

Specifies the volume that contains snapplan data. This name is not determined by the user. It is provided after you run `vmsnap -o snapshot`.

This parameter is required when creating the clone in a two-host configuration.

`-f SNAPPLAN`

Indicates the name of the snapplan that you are using.

`-H`	Specifies the Oracle home directory that corresponds to the `ORACLE_SID`. This is an optional argument.
`-r relocate_path`	Specifies the initial mount point for the snapshot image. If you are creating a clone in a single-host configuration, -r is required. Otherwise, it is an optional argument. If `-r relocate_path` is used when cloning a database, it will also be required for running `-o update_status` and `-o umount`. **Note** Do not use `-r relocate_path` if the `SNAPSHOT_MODE` parameter is set to instant or offline.

▼ To clone a database automatically

In a single-host configuration:

To create a clone of the primary database on the same host as the primary database, use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o recoverdb,new_SID=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2003-04-02 14:43:05
```

In a two-host configuration:

To create a clone of the primary database on a secondary host, use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o recoverdb,new_SID=NEWPROD,vxdbavol=SNAP_arch -f snap2
dbed_vmclonedb started at 2004-04-09 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.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2004-04-09 23:04:50
```

▼ To clone a database manually

In a single-host configuration:

To mount file systems and create a clone of the primary database on the same host as the primary database *without recovering the clone database,* use the `dbed_vmclonedb` command as follows. The clone database must be manually recovered before it can be used:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o mountdb,new_SID=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2003-04-02 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.
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 2004-04-02 15:34:59
```

Recover the database manually.

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 -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
```

In a two-host configuration:

To mount file systems and create a clone of the primary database on a secondary host *without recovering the clone database,* use the `dbed_vmclonedb` command as follows. The clone database must be manually recovered before it can be used:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
 -o mountdb,new_sid=NEWPROD,vxdbavol=SNAP_arch -f snap2
dbed_vmclonedb started at 2003-04-09 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.
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 2004-04-02 15:34:59
```

Recover the database manually.

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 \
-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

In a single-host configuration:

To shut down the clone database and unmount file systems for a clone on the same host as the primary database, use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD \
-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
```

In a two-host configuration:

To shut down the clone database, unmount file systems, and deport the snapshot disk group for a clone on a secondary host, use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD \
-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
```

▼ To restart a clone database

In a single-host configuration:

To start the clone database on the same host as the primary database, use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o restartdb,new_sid=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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 2004-04-02 15:15:19
```

In a two-host configuration:

To start the clone database on the secondary host, use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o restartdb,new_sid=NEWPROD,vxdbavol=SNAP_arch -f snap2
dbed_vmclonedb started at 2003-04-09 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 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.

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/vxdba/logs/sfua_`*`database`*`.log`

  where *`database`* is the *`ORACLE_SID`*.

- By default, only messages with a severity greater than `ERROR` will be logged.
- The `-s` *`set_num`* option specifies the message catalogue set number. The default is 1.
- The `-M` *`msgid`*`[:`*`severity`*`]` option specifies the message ID and severity to be printed.
- The `-f` *`msg_catalog | logfile | log_directory`* option specifies the message catalogue path, log file, or log directory.
- The `-v` *`severity | severity`* option is used to overwrite the minimum log severity or to create a severity filter. The severity values are either `0-8` or `100-108`.
- The `-p` option 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 indicates that no line feed is to be added.
- The `-m` *`value`* option is for internal use only. This option overwrites certain internal settings for the `edgetmsg2` utility.
- The `-o` *`list`*`[,`*`suppress_time`*`]` option displays the content of a log file. You can specify `,`*`suppress_time`* to exclude time information in the utility output.
- The `-o` *`report`*`[,`*`no_archive`*`]` displays the available log files. You can specify `,`*`no_archive`* to exclude log files from the utility output.
- The `-t` *`from_time`*`[,`*`to_time`*`]` reduced 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`*.
- The `-S` *`ORACLE_SID`* option specifies the *`ORACLE_SID`* for an Oracle database.
- "*`default format string`*" option specifies the C language `printf()` format string.
- [*`args`*] specifies arguments for the format string conversion characters.
- See the `edgetmsg2`(1M) manual page for more information.

▼ To print a message

Use the `edgetmsg2` command as follows:

```
$ /opt/VRTSdbed/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/VRTSdbed/bin/edgetmsg2 -o list[,suppress_time] \
-S ORACLE_SID | [-f logfile] \
[-v severity] [-t from_time,to_time]
```

Output similar to the following is displayed:

```
$ edgetmsg2 -o list -S t10 -t "2004-10-14 13:44:45,2004-10-14 14:46:31"
2004-10-14 13:44:45 SFORA vxsnapadm ERROR V-81-5542 Snapback datavol failed.
2004-10-14 13:44:45 SFORA vxsnapadm ERROR V-81-5542 Snapback archvol failed.
2004-10-14 13:44:45 SFORA dbed_vmsnap ERROR V-81-5617 snapback failed.
2004-10-14 13:56:46 SFORA dbed_vmsnap ERROR V-81-5255 sp2 does not exist.
2004-10-14 14:46:16 SFORA dbed_vmclonedb ERROR V-81-4833 There already appears to
be a database T1 running.
2004-10-14 14:46:31 SFORA dbed_vmclonedb ERROR V-81-5713 The database name exceeds
the maximum length of 8 characters.
```

▼ To list available log files

Use the `edgetmsg2` command as follows:

```
$ /opt/VRTSdbed/bin/edgetmsg2 -o report[,no_archive] \
[-f log_directory]
```

Output similar to the following is displayed:

```
dbid     begin               end                 size(K) file
----     -----               ---                 ----    ----
default  2004-10-25 09:50:17 2004-10-25 11:04:00 0       /etc/vx/vxdba/logs/sfua_default.log
t10      2004-09-09 08:54:59 2004-10-21 14:19:35 10      /etc/vx/vxdba/logs/sfua_t10.log
clone2   2004-10-11 09:22:35 2004-10-11 09:22:35 0       /etc/vx/vxdba/logs/sfua_clone2.log
T1       2004-10-14 14:51:33 2004-10-14 14:51:33 0       /etc/vx/vxdba/logs/sfua_T1.log
Tiff1    2004-10-21 14:24:27 2004-10-21 14:24:27 0       /etc/vx/vxdba/logs/sfua_Tiff1.log
clone    2004-08-05 09:23:19 2004-10-21 17:45:19 11      /etc/vx/vxdba/logs/sfua_clone.log
```

Displaying I/O Mapping and Statistics Using vxstorage_stats

You can use the VERITAS Storage Foundation *for Oracle* `vxstorage_stats` command to display I/O mapping and statistics about VERITAS File System files one file at a time. The statistics are recorded only for VxFS files and VxVM volumes. These statistics show I/O activity.

Prerequisites

- You may be logged in as either the database administrator or `root`.

Usage Notes

- The `-m` option displays the I/O topology for the specified file.
- The `-s` option displays the file statistics for the specified file.
- The `-c` *`count`* option specifies the number of times to display statistics.
- The `-i` *`interval`* option specifies the interval frequency for displaying updated I/O statistics.
- The `-f` *`filename`* option specifies the file to display I/O mapping and statistics for.

Command usage for `vxstorage_stats` is as follows:

```
$ /opt/VRTS/bin/vxstorage_stats [-m] [-s] \
[-i interval -c count ] -f file_name
```

▼ To display I/O mapping information

Use the `vxstorage_stats` command with the `-m` option as follows:

```
$ /opt/VRTS/bin/vxstorage_stats -m -f \
/oradata/system01.dbf
TY NAME                   NSUB  DESCRIPTION  SIZE(sectors)    OFFSET(sectors)
PROPERTIES
fi /oradata/system01.dbf  1     FILE         2621442048(B)    4718592(B)
Extents: 3     Sparse Extents:0
v  myindex                1     MIRROR          16777216        0
pl vxvm:mydb/myindex-01          3     STRIPE        16779264       0
Stripe_size:2048
rd /dev/vx/rdmp/c3t1d3s3         1     PARTITION      5593088        0
sd /dev/rdsk/c3t1d3s3            1     PARTITION     17674560       960
sd c3t1d3                        2     MIRROR        17677440       0
da EMC000184502242:02:0c:02      0     DISK          143113019      0
da EMC000184502242:31:0c:02     0     DISK          143113019      0
rd /dev/vx/rdmp/c3t1d15s4       1     PARTITION      5593088       0
sd /dev/rdsk/c3t1d15s4         1     PARTITION     17669760       5760
sd c3t1d15                    2     MIRROR        17677440       0
da EMC000184502242:01:0c:02     0     DISK          143113019      0
da EMC000184502242:32:0c:02     0     DISK          143113019      0
rd /dev/vx/rdmp/c3t1d2s4        1     PARTITION      5593088       0
sd /dev/rdsk/c3t1d2s4          1     PARTITION     17671680       3840
sd c3t1d2                     2     MIRROR        17677440       0
da EMC000184502242:16:0c:02     0     DISK          143113019      0
da EMC000184502242:17:0c:02       0     DISK            143113019       0
```

Note For file type (`fi`), the `SIZE` column is number of bytes, and for volume (`v`), plex (`pl`), sub-disk (`sd`), and physical disk (`da`), the `SIZE` column is in 512-byte blocks. Stripe sizes are given in sectors.

▼ To display the entire I/O mapping and statistics for each I/O stack

Use the `vxstorage_stats` command with the `-m` and `-s` options as follows:

```
$ /opt/VRTS/bin/vxstorage_stats -m -s -f /data/system01.dbf
```

```
TY NAME                   NSUB  DESCRIPTION     SIZE            OFFSET
PROPERTIES
fi /data/system01.dbf     1     FILE            262146048       1172128
Extents: 6      Sparse Extents:0
v  data_vol               2     MIRROR          8388608         0
pl vxvm:mapdg/data_vol-01 1     CONCAT_VOLUME   8388608         0
rd /dev/vx/rdmp/c1t10d0s2 1     PARTITION       8388608         0
sd /dev/rdsk/c1t10d0s2    1     PARTITION       35368272        0
da c1t10d0                0     DISK            35368272        0
pl vxvm:mapdg/data_vol-03 1     CONCAT_VOLUME   8388608         0
rd /dev/vx/rdmp/c1t13d0s2 1     PARTITION       8388608         0
sd /dev/rdsk/c1t13d0s2    1     PARTITION       71127180        0
da c1t13d0                0     DISK            71127180        0
                            OPERATIONS     FILE BLOCKS(512 byte)  AVG TIME(ms)

OBJECT                      READ   WRITE     B_READ   B_WRITE    AVG_RD  AVG_WR
/data/system01.dbf          615    19        20752    152        3.53    24.74
/dev/vx/rdsk/mapdg/data_vol 19444  33318     895903   1376825    9.26    16.14
vxvm:mapdg/data_vol-01      19444  33318     895903   1376825    9.24    14.00
/dev/rdsk/c1t10d0s2         19444  33318     895903   1376825    9.24    14.00

c1t10d0                     19444  33318     895903   1376825    9.24    14.00
vxvm:mapdg/data_vol-03      0      33318     0        1376825    0.00    14.18
/dev/rdsk/c1t13d0s2         0      33318     0        1376825    0.00    14.18
c1t13d0                     0      33318     0        1376825    0.00    14.18
```

Example

To display statistics two times with a time interval of two seconds:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -i2 -c2 \
-f /data/system01.dbf
```

```
                          OPERATIONS     FILE BLOCKS(512 byte)     AVG TIME(ms)

OBJECT                      READ   WRITE     B_READ B_WRITE    AVG_RD    AVG_WR
/data/system01.dbf          615    19        20752  152        3.53      24.74
/dev/vx/rdsk/mapdg/data_vol 19386  33227     895692 1376438    9.27      16.18
vxvm:mapdg/data_vol-01      19386  33227     895692 1376438    9.26      14.03
/dev/rdsk/c1t10d0s2         19386  33227     895692 1376438    9.26      14.03
c1t10d0                     19386  33227     895692 1376438    9.26      14.03
```

```
vxvm:mapdg/data_vol-03      0       33227     0       1376438    0.00      14.21
/dev/rdsk/c1t13d0s2         0       33227     0       1376438    0.00      14.21
c1t13d0                     0       33227     0       1376438    0.00      14.21
                         OPERATIONS     FILE BLOCKS(512 byte)     AVG TIME(ms)

OBJECT                      READ    WRITE     B_READ  B_WRITE    AVG_RD    AVG_WR
/data/system01.dbf          0       0         0       0          0.00      0.00
/dev/vx/rdsk/mapdg/data_vol 0       1         0       2          0.00      0.00
vxvm:mapdg/data_vol-01      0       1         0       2          0.00      0.00
/dev/rdsk/c1t10d0s2         0       1         0       2          0.00      0.00
c1t10d0                     0       1         0       2          0.00      0.00
vxvm:mapdg/data_vol-03      0       1         0       2          0.00      0.00
/dev/rdsk/c1t13d0s2         0       1         0       2          0.00      0.00
c1t13d0                     0       1         0       2          0.00      0.00
```

Mapping Tablespaces to Disks Using dbed_analyzer

The `dbed_analyzer` command provides tablespace-to-physical disk mapping information for *all the datafiles in a specified database*. In addition, `dbed_analyzer` provides information about the percentage of disk space being used by a tablespace.

Because the `dbed_analyzer` command output can be long, it is written to a file for easier viewing. The file name is `dbed_analyzer_$ORACLE_SID.log` and it is located `/tmp`.

Prerequisites

- You must log in as the database administrator (typically, the user ID `oracle`).

Usage Notes

- The `-o sort=tbs` option provides the layout of the specified tablespaces on the physical disk as well as the percentage of disk space they are using.
- The `-o sort=disk` option provides the name of the disks containing the specified tablespaces as well as the percentage of disk space the tablespaces are using.
- The `-f filename` option specifies the name of a file containing a list of the tablespaces for which to obtain mapping information.
- The `-t tablespace` option specifies the name of a tablespace for which to obtain mapping information.
- For more information, see the `dbed_analyzer`(1M) online manual page.

▼ To obtain storage mapping information sorted by tablespace

Use the `dbed_analyzer` command with the `-f filename` and `-o sort=tbs` options:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S $ORACLE_SID -H $ORACLE_HOME
-o sort=tbs -f filename
```

Output similar to the following is displayed in the file *filename*:

```
TBSNAME    DATAFILE                                  DEVICE
SIZE(sectors)
SYSTEM     /usr1/oracle/rw/DATA/PROD.dbf        c3t21000020379DBD5Fd0  819216
TEMP       /usr1/oracle/rw/DATA/temp_20000      c3t21000020379DBD5Fd0  1021968
TEMP       /usr1/oracle/rw/DATA/temp_20001      c3t21000020379DBD5Fd0  2048016
SYSAUX     /usr1/oracle/rw/DATA/sysaux.dbf      c3t21000020379DBD5Fd0  819216
ITEM       /usr1/oracle/rw/DATA/item_1000       c3t21000020379DBD5Fd0  1021968
ITM_IDX    /usr1/oracle/rw/DATA/itm_idx_2000    c3t21000020379DBD5Fd0  1021968
PRODID_IDX /usr1/oracle/rw/DATA/prodid_idx_3000 c3t21000020379DBD5Fd0  1021968
QTY_IDX    /usr1/oracle/rw/DATA/qty_idx_7000    c3t21000020379DBD5Fd0  1021968
ROLL_1     /usr1/oracle/rw/DATA/roll_1_5000     c3t21000020379DBD5Fd0  1021968
ROLL_2     /usr1/oracle/rw/DATA/roll_2_6000     c3t21000020379DBD5Fd0  1021968
ORDERS     /usr1/oracle/rw/DATA/orders_4000     c3t21000020379DBD5Fd0  1021968
ORD_IDX    /usr1/oracle/rw/DATA/ord_idx_10000   c3t21000020379DBD5Fd0  1021968
```

```
QTY_IDX     /usr1/oracle/rw/DATA/qty_idx_7001     c3t21000020379DBD5Fd0  1024016
ITM_IDX     /usr1/oracle/rw/DATA/itm_idx_2001     c3t21000020379DBD5Fd0  1024016
ROLL_1      /usr1/oracle/rw/DATA/roll_1_5001      c3t21000020379DBD5Fd0  1024016
QTY_IDX     /usr1/oracle/rw/DATA/qty_idx_7002     c3t21000020379DBD5Fd0  1024016
ROLL_2      /usr1/oracle/rw/DATA/roll_2_6001      c3t21000020379DBD5Fd0  1024016
ITEM        /usr1/oracle/rw/DATA/item_1001        c3t21000020379DBD5Fd0  4096016
```

▼ To obtain storage mapping information sorted by disk

Use the `dbed_analyzer` command with the `-f` *`filename`* and `-o sort=disk` options:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S $ORACLE_SID -H $ORACLE_HOME
-o sort=disk -f filename
```

Output similar to the following is displayed in the file *`filename`*:

```
DEVICE                 TBSNAME   DATAFILE                           SIZE(sectors)
c3t21000020379DBD5Fd0  SYSTEM    /usr1/oracle/rw/DATA/PROD.dbf            819216
c3t21000020379DBD5Fd0  TEMP      /usr1/oracle/rw/DATA/temp_20000          1021968
c3t21000020379DBD5Fd0  TEMP      /usr1/oracle/rw/DATA/temp_20001          2048016
c3t21000020379DBD5Fd0  SYSAUX    /usr1/oracle/rw/DATA/sysaux.dbf          819216
c3t21000020379DBD5Fd0  ITEM      /usr1/oracle/rw/DATA/item_1000           1021968
c3t21000020379DBD5Fd0  ITM_IDX   /usr1/oracle/rw/DATA/itm_idx_2000        1021968
c3t21000020379DBD5Fd0  PRODID_IDX  /usr1/oracle/rw/DATA/prodid_idx_3000  1021968
c3t21000020379DBD5Fd0  QTY_IDX   /usr1/oracle/rw/DATA/qty_idx_7000        1021968
c3t21000020379DBD5Fd0  ROLL_1    /usr1/oracle/rw/DATA/roll_1_5000         1021968
c3t21000020379DBD5Fd0  ROLL_2    /usr1/oracle/rw/DATA/roll_2_6000         1021968
c3t21000020379DBD5Fd0  ORDERS    /usr1/oracle/rw/DATA/orders_4000         1021968
c3t21000020379DBD5Fd0  ORD_IDX   /usr1/oracle/rw/DATA/ord_idx_10000       1021968
c3t21000020379DBD5Fd0  QTY_IDX   /usr1/oracle/rw/DATA/qty_idx_7001        1024016
c3t21000020379DBD5Fd0  ITM_IDX   /usr1/oracle/rw/DATA/itm_idx_2001        1024016
c3t21000020379DBD5Fd0  ROLL_1    /usr1/oracle/rw/DATA/roll_1_5001         1024016
c3t21000020379DBD5Fd0  QTY_IDX   /usr1/oracle/rw/DATA/qty_idx_7002        1024016
c3t21000020379DBD5Fd0  ROLL_2    /usr1/oracle/rw/DATA/roll_2_6001         1024016
c3t21000020379DBD5Fd0  ITEM      /usr1/oracle/rw/DATA/item_1001           4096016
```

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.

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.

▼ To identify the VxFS files to convert to Quick I/O

1. Use the qio_getdbfiles command as follows:

```
$ /opt/VRTSdbed/bin/qio_getdbfiles [-T ora|syb|db2]
```

where -T forces behavior for a specific database type. Use this option in environments with more than one type of database.

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.

Prerequisites

- To use this command for Oracle, the ORACLE_SID environment variable must be set.
- You must be logged in as the Database Administrator to use this command.

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 qio_convertdbfiles command expects all files to be owned by the database administrator.
- The -a option 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.
- The -f option 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.
- 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).
- The -h option displays a help message.
- The -i option creates extra links for all database files and log files in the /dev directory to support the SAP brbackup command.
- The -u option changes Quick I/O files back to regular files.
- Ensure that the database is shut down before running qio_convertdbfiles.
- See the qio_convertdbfiles(1M) manual page for more information.

▼ To convert VxFS files to Quick I/O files

1. After running the qio_getdbfiles command, shut down the database:

Caution 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/VRTSdbed/bin/qio_convertdbfiles
```

Note 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

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

```
$ /opt/VRTSdbed/bin/qio_convertdbfiles -u
.dbfile::cdev:vxfs: --> dbfile
```

Note 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.

Example

In this example a regular VxFS file named dbfile is converted to Quick I/O and then converted back to a regular VxFS file:

Get information about the file:

```
$ /opt/VRTSdbed/bin/qio_getdbfiles
$ cat mkqio.dat
dbfile 104800000
```

Shut down the database and convert the file to Quick I/O:

```
$ /opt/VRTSdbed/bin/qio_convertdbfiles
$ ls -alL d* .d*
-rw-r--r-- 1 oracle dba 104857600 May 2 13:42 .dbfile
crw-r--r-- 1 oracle dba 45,1 May 3 12:18 dbfile

$ ls -al d* .d*
-rw-r--r-- 1 oracle dba 104857600 May 2 14:42 .dbfile
lrwxrwxrwx 1 oracle dba 19 May 3 12:18 dbfile ->
.dbfile::cdev:vxfs:
```

The qio_convertdbfiles command renames the file dbfile to .dbfile and creates a symbolic link to .dbfile with the Quick I/O extension. By default, the symbolic link uses a relative path name.

Start up the database.

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

```
$ /opt/VRTSdbed/bin/qio_convertdbfiles -u
.dbfile::cdev:vxfs: --> dbfile
```

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.

Prerequisites

- To use this command for Oracle, the ORACLE_SID environment variable must be set.
- You must be logged in as the Database Administrator to use this command.

Usage Notes

- The qio_recreate command expects to find a file named mkqio.dat 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 mkqio.dat file is not in the directory, you will be prompted to create it using qio_getdbfiles. For more information, see "Identifying VxFS Files to Convert to Quick I/O Using qio_getdbfiles" on page 484.
- 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.
- 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.
- See the `qio_recreate`(1M) manual page for more information.

▼ To automatically recreate Quick I/O files when the database is recovered

Use the `qio_recreate` command as follows:

```
$ /opt/VRTSdbed/bin/qio_recreate
```

Using Third-Party Software to Back Up Files B

Using third-party software to back up VERITAS Quick I/O files or Oracle Disk Manager files requires special consideration and handling. This appendix discusses these issues. For information about backing up files using VERITAS NetBackup, refer to "Using VERITAS NetBackup for Database Backup" on page 261.

Topics covered in this chapter include:

Using Oracle RMAN to Back Up and Restore Quick I/O Files

Quick I/O files are treated as raw devices by Oracle and Recovery Manager (RMAN) and must be backed up and restored the same way as raw devices. A Quick I/O file consists of two components: a regular file with space allocated to it and a link pointing to the Quick I/O interface for the file.

When a Quick I/O file is created with the `qiomkfile` command, the regular file with the preallocated space is a hidden file. For example, `dbfile` points to `.dbfile::cdev:vxfs:` and `.dbfile` is the hidden file with the space allocated. (These names are used in the examples throughout this section.)

For backup, RMAN reads the Oracle datafile using the Quick I/O interface, but does not process or store the special link between the hidden file with the allocated space (`.dbfile`) and the link to its Quick I/O interface (`dbfile`, which points to `.dbfile::cdev:vxfs:`). This has implications for the restore operation, as described in the rest of this section.

Because Quick I/O files are treated as raw devices, the Quick I/O file must exist and have the necessary space preallocated to it before the file is restored using RMAN. Space can be preallocated to the file when the file is created using the `qiomkfile` command. In this case, the file can be restored using RMAN with no other special handling, and the file can be accessed after the restore as a Quick I/O file:

- If both the Quick I/O link name and the hidden file are missing, use `qiomkfile` to preallocate and set up the Quick I/O file.
- If either the Quick I/O link name or the hidden file alone exist, delete these files and recreate the Quick I/O file of the required size using `qiomkfile`.
- If both the Quick I/O link name and the hidden file are intact, you may proceed to restoring the file.
- If you attempt to restore a Quick I/O file and the original is smaller than the required size, the restore will fail with an Oracle error ORA-27069 (I/O attempt beyond the range of the file). The failure results because Quick I/O does not allow extending writes (in other words, attempts to increase the size of a file by writing beyond the end of the file). This same behavior is encountered when attempting to restore Oracle datafiles built on raw devices. If the restore fails with the above error, delete the Quick I/O link and its hidden file, then recreate or extend the file using `qiomkfile`.

Note The space needed for the Oracle datafile is the size of the datafile plus one Oracle block (as specified by the `init.ora` parameter, `db_block_size`).

Using Oracle RMAN to Back Up and Restore Oracle Disk Manager Files

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. Refer to the `odmmkfile(1)` manual page for more information.

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
```

Using Legato NetWorker to Back Up and Restore Quick I/O Files

When Quick I/O files are created using the command `qiomkfile`, a hidden file with storage space allocated to it and a link are created. The link points to the Quick I/O interface of the hidden file. Using `qiomkfile` ensures that the space for the file is allocated in a contiguous manner, which typically improves performance.

Legato NetWorker does not follow symbolic links during backups because doing so would result in the data getting backed up twice: once using the symbolic link and once as the file itself. As a result, Quick I/O files must be backed up as two separate files and restored accordingly.

Because Legato NetWorker deletes and recreates files before they are restored, the restored files lose their contiguous allocation and could be restored as fragmented files with indirect extents. While this does not impact the integrity of the data being restored, it can degrade performance. Creating the file using `qiomkfile` before doing the backup does not resolve this problem because NetWorker deletes and recreates the file.

To avoid this potential performance degradation, Quick I/O files must be backed up and restored using the same methods used to back up and restore raw devices. This method involves using the NetWorker `rawasm` command to back up or `save` directories containing Quick I/O files. Because of the way the `rawasm` command works, NetWorker follows the Quick I/O symbolic link to back up the actual data in the hidden file. Skip the hidden file to avoid backing up the data twice. During restore, NetWorker looks at the attributes of the saved file and restores it using `rawasm`, bypassing the file deletion and recreation steps. For example:

```
$ ls -al /db01
total 2192
drwxr-xr-x   2 root    root         96    Oct 20 17:39 .
drwxr-xr-x   9 root    root       8192    Oct 20 17:39 ..
-rw-r--r--   1 oracle  dba     1048576    Oct 20 17:39 .dbfile
lrwxrwxrwx   1 oracle  dba          22    Oct 20 17:39 dbfile ->\
                                                   .dbfile::cdev:vxfs:
```

The command for backing up the `/db01` directory using `rawasm` would look like:

```
<< /db01 >>
rawasm: dbfile
skip: .dbfile
```

To restore the file, preallocate the Quick I/O file using the `qiomkfile` command and enter:

```
$ cd /db01
$ recover -a /db01/dbfile
```

Using Backup Software Other than VERITAS NetBackup to Back Up and Restore ODM Files

If you are using 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.

VERITAS Database FlashSnap Status Information C

The VERITAS Database FlashSnap functionality provides both snapshot status information and snapshot database status information for various stages of snapplan and snapshot procedures. You can view the status information through the CLI and through the GUI.

For more information about Database FlashSnap command line functionality, see “Using Database FlashSnap for Backup and Off-Host Processing” on page 197.

For more information about Database FlashSnap GUI functionality, see “Managing Snapshots with Database FlashSnap” on page 310.

Topics covered in this appendix are:

- “Obtaining Database FlashSnap Snapshot Status and Database Status Using the CLI” on page 496
- “Obtaining Database FlashSnap Snapshot Status and Database Status from the GUI” on page 504

Obtaining Database FlashSnap Snapshot Status and Database Status Using the CLI

You can obtain both the snapshot status and the database status from the command line using the `dbed_vmchecksnap` command with the `-o list` option. The snapshot status and database status information may also appear in error messages. The tables in this section provide detailed information about the various status values.

Snapshot Status Details

To view snapshot 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 status information is displayed in the command output under the column heading `SNAP_STATUS`.

For information about the various status values, see the following table:

SNAP_STATUS	Completed Operations	Allowed Operations
`init_full`	◆ `dbed_vmchecksnap -o validate` (successful) ◆ `dbed_vmsnap -o resync` (successful) ◆ `dbed_vmsnap -o reverse_resync_commit` (successful)	◆ `dbed_vmsnap -o snapshot`
`init_db`	◆ `dbed_vmchecksnap -o validate -f snapplan` (failed)	◆ `dbed_vmchecksnap -o validate | list -f` *`snapplan`* ◆ Ensure that your storage configuration has been set up correctly. See "Preparing Hosts and Storage for Database FlashSnap" on page 204.
`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.

SNAP_STATUS	Completed Operations	Allowed Operations
snapshot_end	◆ dbed_vmsnap -o snapshot (successful) ◆ dbed_vmsnap -o reverse_resync_abort (successful)	◆ dbed_vmsnap -o resync ◆ dbed_vmsnap -o reverse_resync_begin ◆ dbed_vmclonedb -o mount|mountdb|recoverdb
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.
reverse_resync_begin_start	◆ dbed_vmsnap -o reverse_resync_begin (failed)	◆ Contact VERITAS support.
reverse_resync_begin_end	◆ dbed_vmsnap -o reverse_resync_begin (successful)	◆ dbed_vmsnap -o reverse_resync_abort ◆ dbed_vmsnap -o reverse_resync_commit
reverse_resync_abort_end	◆ dbed_vmsnap -o reverse_resync_abort (successful)	◆ dbed_vmsnap -o reverse_resync_begin ◆ dbed_vmsnap -o resync ◆ dbed_vmclonedb -o restartdb (with DB_STATUS set to database_recovered)
reverse_resync_commit_start	◆ dbed_vmsnap -o reverse_resync_commit (failed)	◆ Contact VERITAS support.
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.

SNAP_STATUS	Completed Operations	Allowed Operations
`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`
`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` ◆ `dbed_vmsnap -o reverse_resync_begin`

Database Status Details

To view snapshot 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. Database status information is displayed in the command output under the column heading `DB_STATUS`. For information about the various status values, see the following table:

DB_STATUS	Completed Operations
`init`	◆ `dbed_vmchecksnap -o validate` (successful) ◆ `dbed_vmsnap -o snapshot` (successful) ◆ `dbed_vmsnap -o reverse_resync_begin` (successful)
`database_recovered`	◆ `dbed_vmclonedb -o recoverdb` (successful)

Obtaining Database FlashSnap Snapshot Status and Database Status Using the CLI

You can obtain both the snapshot status and the database status from the command line using the `dbed_vmchecksnap` command with the `-o list` option. The snapshot status and database status information may also appear in error messages. The tables in this section provide detailed information about the various status values.

Snapshot Status Details

To view snapshot 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 status information is displayed in the command output under the column heading `SNAP_STATUS`.

For information about the various status values, see the following table:

SNAP_STATUS	Completed Operations	Allowed Operations
`init_full`	◆ `dbed_vmchecksnap -o validate` (successful) ◆ `dbed_vmsnap -o resync` (successful) ◆ `dbed_vmsnap -o reverse_resync_commit` (successful)	◆ `dbed_vmsnap -o snapshot`
`init_db`	◆ `dbed_vmchecksnap -o validate -f snapplan` (failed)	◆ `dbed_vmchecksnap -o validate | list -f` *`snapplan`* ◆ Ensure that your storage configuration has been set up correctly. See "Preparing Hosts and Storage for Database FlashSnap" on page 204.
`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.

SNAP_STATUS	Completed Operations	Allowed Operations
snapshot_end	◆ dbed_vmsnap -o snapshot (successful) ◆ dbed_vmsnap -o reverse_resync_abort (successful)	◆ dbed_vmsnap -o resync ◆ dbed_vmsnap -o reverse_resync_begin ◆ dbed_vmclonedb -o mount|mountdb|recoverdb
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.
reverse_resync_begin_start	◆ dbed_vmsnap -o reverse_resync_begin (failed)	◆ Contact VERITAS support.
reverse_resync_begin_end	◆ dbed_vmsnap -o reverse_resync_begin (successful)	◆ dbed_vmsnap -o reverse_resync_abort ◆ dbed_vmsnap -o reverse_resync_commit
reverse_resync_abort_end	◆ dbed_vmsnap -o reverse_resync_abort (successful)	◆ dbed_vmsnap -o reverse_resync_begin ◆ dbed_vmsnap -o resync ◆ dbed_vmclonedb -o restartdb (with DB_STATUS set to database_recovered)
reverse_resync_commit_start	◆ dbed_vmsnap -o reverse_resync_commit (failed)	◆ Contact VERITAS support.
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.

SNAP_STATUS	Completed Operations	Allowed Operations
`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`
`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` ◆ `dbed_vmsnap -o reverse_resync_begin`

Database Status Details

To view snapshot 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. Database status information is displayed in the command output under the column heading `DB_STATUS`. For information about the various status values, see the following table:

DB_STATUS	Completed Operations
`init`	◆ `dbed_vmchecksnap -o validate` (successful) ◆ `dbed_vmsnap -o snapshot` (successful) ◆ `dbed_vmsnap -o reverse_resync_begin` (successful)
`database_recovered`	◆ `dbed_vmclonedb -o recoverdb` (successful)

Obtaining Database FlashSnap Snapshot Status and Database Status from the GUI

You can obtain both the snapshot status and the snapshot database status from the GUI. The tables in this section provide detailed information regarding the various status values.

Database FlashSnap Snapshot Status Details

To view snapshot status information from the GUI, click on a specific snapplan in the object tree. The snapshot status can be seen on the right side of the window in the **Snapplan State** field. For information regarding the various status values, see the following table:

Snapshot Status (as seen in the Snapplan State field)	Completed Operations	Allowed Operations
`init_full`	◆ Modify/Validate Snapplan (successful) ◆ Resync Snapshot (successful) ◆ Reverse Resync Snapshot with the **commit** option (successful)	◆ Create Snapshot
`snapshot_start`	◆ Create Snapshot (failed)	◆ If the “Create Snapshot” operation failed, contact your system administrator for help. You can use the VxVM utilities to create a snapshot and resynchronize the snapshot volumes, then use the “Create Snapshot” operation with the **Force snapshot creation** option for the subsequent snapshot.
`snapshot_end`	◆ Create Snapshot (successful)	◆ Resync Snapshot ◆ Reverse Resync Snapshot with the **begin** option ◆ Create Snapshot Database with the **Create database** option

Snapshot Status (as seen in the Snapplan State field)	Completed Operations	Allowed Operations
`resync_start`	◆ Resync Snapshot (failed)	◆ If the "Resync Snapshot" operation failed, contact your system administrator for help. You can use the VxVM utilities to resynchronize the snapshot volumes, then use the "Create Snapshot" operation with the **Force snapshot creation** option for the subsequent snapshot.
`reverse_resync_begin_start`	◆ Reverse Resync Snapshot with the **begin** option (failed)	◆ Contact VERITAS support.
`reverse_resync_begin_end`	◆ Reverse Resync Snapshot with the **begin** option (successful)	◆ Reverse Resync with the **commit** option ◆ Reverse Resync with the **abort** option
`reverse_resync_abort_start`	◆ Reverse Resync Snapshot with the **abort** option (failed)	◆ Contact VERITAS support.
`reverse_resync_abort_end`	◆ Reverse Resync Snapshot with the **abort** option (successful)	◆ Reverse Resync Snapshot with the **begin** option ◆ Resync Snapshot ◆ Create Snapshot Database with the **Restart database** option
`reverse_resync_commit_start`	◆ Reverse Resync Snapshot with the **commit** option (failed)	◆ Contact VERITAS support.
`restartdb_start`	◆ Start Up Snapshot Database with the **Restart database** option (failed)	◆ Try to start the snapshot database manually.
`restartdb_end`	◆ Create Snapshot Database with the **Restart database** option (successful)	◆ Shut Down Database with the **unmount** option

Snapshot Status (as seen in the Snapplan State field)	Completed Operations	Allowed Operations
`mountdb_start`	◆ `dbed_vmclonedb -o mountdb` command failed from the CLI **Note** This option is not supported in the GUI.	◆ Recover the snapshot database manually, then run the `dbed_vmclonedb -o update_status` command from the CLI **Note** This option is not supported in the GUI.
`mountdb_end`	◆ `dbed_vmclonedb -o mountdb` command from the CLI was successful **Note** This option is not supported in the GUI.	◆ Umount Database **FlashSnap** ◆ `dbed_vmclonedb -o update_status` command from the CLI **Note** This option is not supported in the GUI.
`recoverdb_start`	◆ Create Snapshot Database with the **Restart database** option (failed)	◆ Recover the snapshot database manually, then run the `dbed_vmclonedb -o update_status` command from the CLI **Note** This option is not supported in the GUI.
`recoverdb_end`	◆ Create Snapshot Database with the **Restart database** option (successful)	◆ Shut Down Database with the **umount** option
`umount_start`	◆ `dbed_vmclonedb -o umount` command failed from the CLI **Note** This option is not supported in the GUI.	◆ Verify that your file system(s) are not busy and retry the command.
`umount_end`	◆ `dbed_vmclonedb -o umount` command from the CLI was successful **Note** This option is not supported in the GUI.	◆ Start Up Snapshot Database with the **restart database** option ◆ Resync Snapshot ◆ Reverse Resync Snapshot with the **begin** option

Snapshot Database Status Details

To view snapshot database status information from the GUI, click on a specific snapplan in the object tree. The database status can be seen on the right side of the window in the **Database Status** field. For information regarding the various status values, see the following table:

Database Status (as seen in the Database State field)	Completed Operations
`init`	◆ Modify/Validate Snapplan (successful) ◆ Create Snapshot (successful) ◆ Reverse Resync Snapshot (successful)
`database_recovered`	◆ Start Up Snapshot Database with the **startup database** option (successful)

Accessibility and VERITAS Storage Foundation for Oracle

D

VERITAS products meet federal accessibility requirements for software as defined in Section 508 of the Rehabilitation Act:

http://www.access-board.gov/508.htm

VERITAS Storage Foundation *for Oracle* Graphical User Interface (GUI) inherits the font size and color settings from the operating system it is running on.

Keyboard shortcuts are available for all major GUI operations and menu items. VERITAS products are compatible with operating system accessibility settings as well as a variety of assistive technologies. All manuals also are provided as accessible PDF files, and the online help is provided as HTML displayed in a compliant viewer.

The following topics detail accessibility features and compliance in VERITAS Storage Foundation *for Oracle*:

- "Keyboard Navigation and Shortcuts in VERITAS Storage Foundation for Oracle" on page 510
- "Keyboard Shortcuts" on page 514
- "Menu Hot Keys" on page 516
- "Support for Accessibility Settings and Assistive Technologies" on page 519

Keyboard Navigation and Shortcuts in VERITAS Storage Foundation for Oracle

All program functions and menu items are accessible using the keyboard exclusively. VERITAS Storage Foundation *for Oracle* uses standard operating system navigation keys and keyboard shortcuts.

To see a table of the standard Microsoft navigation keys and keyboard shortcuts, select your version of Microsoft Windows from the drop-down listbox at:

http://www.microsoft.com/enable/products/keyboard.aspx

VERITAS Enterprise Administrator (VEA) has the following exceptions and additions to the Microsoft Keyboard standard:

VEA Exceptions to Microsoft Keyboard Conventions

Convention	Mnemonic	Accelerator
Cancel	C	ALT+C
OK	O	ALT+O
Refresh	e	None
Yes	Y	ALT+Y
No	N	ALT+N
Shift+F10	None	None

VEA Additions to Microsoft Keyboard Conventions

Convention	Mnemonic	Accelerator
Connect	C	None
Contents	C	None
Delete Now	D	ALT+ D
Delete Temporary Files on Exit	e	ALT+ E
Disconnect	D	None

VEA Additions to Microsoft Keyboard Conventions

Convention	Mnemonic	Accelerator
Help	H	ALT+ H
Number of Items in History	N	ALT+ N
Rescan	s	None
Set to defaults	S	ALT+ S
Remember Password	R	ALT+ R
Username	U	ALT+ U

VEA Help Additions to Microsoft Keyboard Conventions

Convention	Mnemonic	Accelerator
Find in Topic..	F	Ctrl + F
Find Next	n	F3
Find Previous	P	None
Hide Navigation Tabs	H	None
Index	I	None
Print	P	None
Search	S	None
Synchronize	n	None

General Keyboard Navigation Within the GUI

You can navigate and use VERITAS Storage Foundation *for Oracle* with only the keyboard. In the GUI, the current active tree or table has a dark blue highlight, and the current active tab, radio button, or checkbox is enclosed within a rectangle formed by dotted lines. These areas are said to have *focus* and will respond to commands.

All VERITAS GUIs use the following keyboard navigation standards:

- Tab moves the focus to the next active area, field, or control, following a preset sequence. Shift+Tab moves the focus in the reverse direction through the sequence.
- Ctrl+Tab exits any Console area that you internally navigate with Tab.
- Up and Down arrow keys move focus up and down the items of a list.
- ALT in combination with the underlined mnemonic letter for a field or command button shifts the focus to that field or button.
- Enter activates your selection. For example, after pressing Tab to select the **Next** button in a wizard panel, press Enter to display the next screen.

Keyboard Navigation Within Dialog Boxes

Dialog boxes contain groups of controls necessary to set options or settings for programs. Here are some general rules about dialog box navigation:

- Tab moves focus between controls within the dialog box along a preset sequence.
- Controls displaying a mnemonic (an underlined letter) can be selected regardless of focus by typing ALT and the underlined letter.
- A dark border indicates the default command button. Press Enter at any time to choose the button with a dark border.
- ALT C chooses the **Cancel** button if one exists.
- Spacebar chooses a control you select with Tab.
- Spacebar changes the state of a checkbox or radio button that has focus. Typing a mnemonic (if one is available) will move the focus to the checkbox or radio button and change its state.
- Arrow keys move focus within listboxes, sliders, groups of option controls, or groups of page tabs.
- Items that cannot be changed are not visited by the Tab key sequence. Options that are unavailable are grayed-out and can neither be selected nor given focus.

While the controls described here are typically found in dialog boxes, they also can occur in other contexts. The same navigation standards will apply.

Tabbed Dialog Boxes

Some dialog boxes use tabbed pages to subcategorize groups of many options. Each tabbed page contains different groups of controls. Use Tab to move the focus between tabs within a dialog box. Typing the mnemonic for a tab also moves the focus to the tab and displays its page of controls.

The following table lists keyboard navigation rules within tabbed dialog boxes:

Keyboard Navigation within Tabbed Dialog Boxes

Keyboard input	**Result**
Ctrl+Page Down	Switches to the next tab and displays the page
Ctrl+Page Up	Switches to the previous tab and displays the page
Right arrow or Left arrow	When the focus is on a tab selector, chooses the next or previous tab in the current row and displays the page

Listboxes

Listboxes display a column of available choices. There are different kinds of listboxes with additional navigation conventions:

- *Drop-down listboxes* by default show only the selected item. A small button to the right of the control shows a downward-pointing arrow. Select the arrow to display more items from the listbox. If there are more choices than can fit in the preset listbox area, a slider appears along the side of the listbox. Show or hide the list using F4. Enter selects or deselects an item.
- *Extended selection listboxes* support selecting single items, blocks of items, or combinations of the two. After selecting an item, hold down Shift+ or Ctrl+navigation keys to select or deselect additional items or blocks of items.

Keyboard Shortcuts

All menu items can be selected by using accelerator or mnemonic keyboard shortcuts. An accelerator is a key combination that provides shortcut access to a GUI function. A mnemonic (sometimes referred to as a "hot key") is a single-key equivalent (used in combination with the ALT key) for selecting GUI components such as menu items. The mnemonic "hot key" letter is underlined in the GUI. For example:

- ALT to go into menu pull-down mode
- F key to access the **File** menu
- O key to activate the open command

Mnemonics are case-insensitive. Keys can be pressed sequentially instead of simultaneously.

Keyboard Shortcuts

Keyboard Input	Action
Tab, Shift-Tab (for reversing the action)	Navigates between main components of the user-interface
Shift-F10	Display Context-sensitive menu
Ctrl-A	Selects all items in list
F3	Find Next
Enter, Return	Activates default button (does not require keyboard focus)

Routine functions such as opening, saving, and printing files can be performed using the standard Microsoft keyboard shortcuts.

Keyboard shortcuts are not case-sensitive. Mnemonic keystrokes may be pressed either sequentially or simultaneously. All menu items have mnemonics, but not all menu items have accelerators.

Keyboard Navigation

The following table lists some of the keys frequently used to navigate with the keyboard:

Keyboard Navigation

Keyboard Input	Result
TAB	Move forward between panes in the active Console window.
SHIFT+TAB	Move backwards between panes in the active Console window.
SHIFT+ UP ARROW	Move up one item in the tree view.
SHIFT+DOWN ARROW	Move down one item in the tree view.
SHIFT+PAGE UP	Move to the top item visible in the tree view.
HOME	Move to the first item in the tree view.
END	Move to the last item in the tree view.
RIGHT ARROW	Expands the highlighted item. If the highlighted item does not contain hidden items, behaves like DOWN ARROW.
LEFT ARROW	Collapses the highlighted item. If the highlighted item does not contain expanded items, behaves like UP ARROW.

Menu Hot Keys

The following table lists the hot keys associated with the different menus:

Menu Hot Keys

Action	Keyboard Input
File	ALT+F to open the menu, then: ◆ C– Display the connection dialog window ◆ D-Display the disconnection dialog window ◆ r–Display the properties of the connected host ◆ u–Page setup for print ◆ w–Print preview ◆ P–Print the page ◆ x –Exit
Tools	ALT+T to open the menu, then: ◆ P–Set the display preferences ◆ M–Manage user profiles ◆ E–Display the error console ◆ H– Customize the table headers ◆ c–Scan the host for devices ◆ S–Search for storage devices
Actions	ALT+ A to open the menu, then: ◆ e–Refresh ◆ s–rescan ◆ g–Foreign Device: ◆A–Add a device ◆R–Remove a device ◆L–List the devices ◆
Window	◆ ALT+ W to open the menu, then: ◆ T–Tear off a component from being shown 1-To tear off details 2-To tear off the system details

Menu Hot Keys

Action	Keyboard Input
Help	ALT+H to open the menu, then: ◆ C– Display the contents ◆ b– Display information about the Help viewer
Oracle Databases	ALT+O to open the menu, then: ◆ S– Start the Oracle Database ◆ D–Start the Snapshot Database ◆ v–Reverse resync a snapshot ◆ U– Update the rescan intervals ◆ R–Rescan the databases
Oracle Instance	ALT+ O to open the menu, then: ◆ S– Startup the database instance ◆ D– Shut down the database ◆ C–Create Clone Database ◆ o–Resync Repository ◆ h– Check System configuration ◆ a– Save system configuration ◆ R–Rescan Oracle instance
Monitoring Agent	ALT+M to open the menu, then: ◆ S–Start the monitoring agent ◆ o–Stop the monitoring agent ◆ E–Enable the agent at boot time ◆ D–Disable the agent at boot time
Snapplans	ALT+ S to open menu, then: ◆ C–Create a snapplan ◆ M–Modify/Validate a snapplan ◆ R–Rescan Snapplans

Menu Hot Keys

Action	Keyboard Input
Snapplan	ALT+ S to open menu, then: ◆ M–Modify/Validate a snapplan ◆ R–Remove a Snapplan ◆ C–Create a snapshot using snapplan ◆ y–Resync a snapshot ◆ v–Reverse Resync a snapshot ◆ L–View log for the Snapplan
Storage Checkpoints	ALT+S to open menu, then: ◆ C–Create a Storage Checkpoint ◆ P–Create a Storage Checkpoint policy ◆ R– Rescan system information
Storage Checkpoint	ALT+S to open menu, then: ◆ B–Rollback a storage checkpoint ◆ M–Mount a storage checkpoint ◆ U–Unmount a storage checkpoint ◆ R–Remove a storage checkpoint
Tablespaces	ALT+ S to open the menu, then: ◆ R–Rescan tablespaces
Datafile	ALT+D to open the menu, then: ◆ T– Topology/Statistics to generate datafile statistics ◆ C–Conversion (convert Oracle datafiles to Quick I/O files)

Support for Accessibility Settings and Assistive Technologies

VERITAS software responds to operating system accessibility settings.

VERITAS products are compatible with Microsoft's accessibility utilities. In Windows 2000, accessibility options involving keyboard responsiveness, display contrast, alert sounds, and mouse operation can be set through the Control Panel (**Start > Settings > Control Panel > Accessibility Options**) and through the Accessibility Wizard (**Start > Programs > Accessories > Accessibility > Accessibility Wizard**).

Note Though all graphics in VERITAS documentation can be read by screen readers, setting your screen reader to ignore graphics may improve performance.

Glossary

address-length pair

Identifies the starting block address and the length of an extent (in file system or logical blocks).

archived log mode

Used to retrieve information on transactions that occur during a hot backup.

asynchronous I/O

A format of I/O that performs non-blocking reads and writes. This enables the system to handle multiple I/O requests simultaneously.

autoextend

An Oracle feature that automatically grows a database file by a prespecified size, up to a prespecified maximum size.

atomic operation

An operation that either succeeds completely or fails and leaves everything as it was before the operation was started. If the operation succeeds, all aspects of the operation take effect at once and the intermediate states of change are invisible. If any aspect of the operation fails, then the operation aborts without leaving partial changes.

backup mode

A state of the Oracle tablespace that lets you perform online backup.

Block-Level Incremental (BLI) Backup

A method used to back up only changed data blocks, not changed files, since the last backup.

block map

A file system is divided into fixed-size blocks when it is created. As data is written to a file, unused blocks are allocated in ranges of blocks, called extents. The extents are listed or pointed to from the inode. The term used for the data that represents how to translate an offset in a file to a file system block is the “block map” for the file.

boot disk

A disk used for booting an operating system purposes.

buffered I/O

A mode of I/O operation (where I/O is any operation, program, or device that transfers data to or from a computer) that first transfers data into the Operating System buffer cache. For a read, it is then copied to the application buffer; for a write, it is written to the storage system.

cache

Any memory used to reduce the time required to respond to an I/O request. The read cache holds data in anticipation that it will be requested by a client. The write cache holds data written until it can be safely stored on non-volatile storage media.

Cached Quick I/O

Cached Quick I/O allows databases to make more efficient use of large system memory while still maintaining the performance benefits of Quick I/O. Cached Quick I/O provides an efficient, selective buffering mechanism to complement asynchronous I/O.

cluster

A set of hosts that share a set of disks.

cluster-shareable disk group

A disk group in which the disks are shared between more than one host.

cold backup

The process of backing up of a database that is not in active use.

command launcher

A graphical user interface (GUI) window that displays a list of tasks that can be performed by VERITAS Volume Manager or other objects. Each task is listed with the object type, task (action), and a description of the task. A task is launched by clicking on the task in the Command Launcher.

concatenation

A VERITAS volume manager layout style characterized by subdisks that are arranged sequentially and contiguously.

configuration database

A set of records containing detailed information on existing VERITAS Volume Manager objects (such as disk and volume attributes). A single copy of a configuration database is called a configuration copy.

control file

An Oracle control file specifies the physical structure of an Oracle database, including such things as the database name, names and locations of the datafiles and redo log files, and the timestamp of when the database was created. When you start an Oracle database, the control file is used to identify the database instance name redo log files that must be opened for transaction recording and recovery and datafiles where data is stored.

copy-on-write

A technique for preserving the original of some data. As data is modified by a write operation, the original copy of data is copied in some fashion.

Applicable to Storage Checkpoint technology, where original data, at the time of the Storage Checkpoint, must be copied from the file system to the Storage Checkpoint when it is to be overwritten. This preserves the frozen image of the file system in the Storage Checkpoint.

data block

A logical database data storage unit. Blocks contain the actual data. When a database is created, a data block size is specified. The database then uses and allocates database space in data blocks.

data change object (DCO)

A VERITAS Volume Manager object used to manage FastResync map information in the DCO log volume. A DCO and a DCO log volume must be associated with a volume in order to implement persistent FastResync on that volume.

Data Storage Checkpoint

A Storage Checkpoint that is a complete image of a file system. For more information, see “Storage Checkpoint.”

database

A database is a collection of information that is organized in a structured fashion. Two examples of databases are Relational Databases (such as Oracle, Sybase, or DB2), where data is stored in tables and generally accessed by one or more keys and Flat File Databases, where data is not generally broken up into tables and relationships. Databases generally provide tools and/or interfaces to retrieve data.

VxDBA maintains a flat file database of information on the available Oracle instances on a system.

datafile

A physical database attribute that contains database data. An Oracle datafile can only be associated with a single database. One or more datafiles form a logical database storage unit called a tablespace.

DCO

See "data change object (DCO)."

DCO log volume

A volume used to hold persistent FastResync change maps.

Decision Support Systems

Decision Support Systems (DSS) are computer-based systems used to model, identify, and solve problems, and make decisions.

defragmentation

The act of reorganizing data to reduce fragmentation. Data in file systems become fragmented over time.

device file

A block- or character-special file located in the `/dev` directory representing a device.

device name

The name of a device file. It represents a device. The `c#t#d#s#` syntax identifies the controller, target address, disk, and partition.

direct I/O

An unbuffered form of I/O that bypasses the kernel's buffering of data. With direct I/O, data is transferred directly between the disk and the user application.

Dirty Region Logging

The procedure by which the VERITAS Volume Manager monitors and logs modifications to a plex. A bitmap of changed regions is kept in an associated subdisk called a *log subdisk*.

disk access name

The name used to access a physical disk, such as `c0t0d0s2`. The `c#t#d#s#` syntax identifies the controller, target address, partition, and disk. The term *device name* can also be used to refer to the disk access name.

disk array

A collection of disks logically and physically arranged into an object. Arrays provide benefits including data redundancy and improved performance.

disk cache

A section of RAM that provides a cache between the disk and the application. Disk cache enables the computer to operate faster. Because retrieving data from hard disk can be slow, a disk caching program helps solve this problem by placing recently accessed data in the disk cache. Next time that data is needed, it may already be available in the disk cache; otherwise a time-consuming operation to the hard disk is necessary.

disk group

A collection of disks that share a common configuration. A disk group configuration is a set of records containing detailed information on existing VERITAS Volume Manager objects (such as disk and volume attributes) and their relationships. Each disk group has an administrator-assigned name and an internally defined unique ID. The root disk group (`rootdg`) is a special private disk group that always exists.

disk name

A VERITAS Volume Manager logical or administrative name chosen for the disk, such as `disk03`. The term *disk media name* is also used to refer to the disk name.

DMP

See “Dynamic Multipathing.”

DSS

See “Decision Support Systems.”

Dynamic Multipathing

Dynamic Multipathing (DMP) is a VERITAS Volume Manager feature that allows the use of multiple paths to the same storage device for load balancing and redundancy.

error handling

Routines in a program that respond to errors. The measurement of quality in error handling is based on how the system informs the user of such conditions and what alternatives it provides for dealing with them.

evacuate

Moving subdisks from the source disks to target disks.

extent

A logical database attribute that defines a group of contiguous file system data blocks that are treated as a unit. An extent is defined by a starting block and a length.

extent attributes

The extent allocation policies associated with a file and/or file system. For example, see "address-length pair."

failover

The act of moving a service from a failure state back to a running/available state. Services are generally applications running on machines and failover is the process of restarting these applications on a second system when the first has suffered some form of failure.

file system

A collection of files organized together into a structure. File systems are based on a hierarchical structure consisting of directories and files.

file system block

The fundamental minimum size of allocation in a file system.

fileset

A collection of files within a file system.

fixed extent size

An extent attribute associated with overriding the default allocation policy of the file system.

fragmentation

Storage of data in non-contiguous areas on disk. As files are updated, new data is stored in available free space, which may not be contiguous. Fragmented files cause extra read/write head movement, slowing disk accesses.

gigabyte

Approximately one billion bytes. Also GB, Gbyte, G-byte.

high availability (HA)

The ability of a system to perform its function continuously (without significant interruption) for a significantly longer period of time than the combined reliabilities of its individual components. High availability is most often achieved through failure tolerance and inclusion of redundancy; from redundant disk to systems, networks, and entire sites.

hot backup

The process of backing up a database that is online and in active use.

hot pluggable

To pull a component out of a system and plug in a new one while the power is still on and the unit is still operating. Redundant systems can be designed to swap disk drives, circuit boards, power supplies, CPUs, or virtually anything else that is duplexed within the computer. Also known as hot swappable.

hot-relocation

A VERITAS Volume Manager technique of automatically restoring redundancy and access to mirrored and RAID-5 volumes when a disk fails. This is done by relocating the affected subdisks to disks designated as spares and/or free space in the same disk group.

inode list

An inode is an on-disk data structure in the file system that defines everything about the file, except its name. Inodes contain information such as user and group ownership, access mode (permissions), access time, file size, file type, and the block map for the data contents of the file. Each inode is identified by a unique inode number in the file system where it resides. The inode number is used to find the inode in the inode list for the file system. The inode list is a series of inodes. There is one inode in the list for every file in the file system.

instance

When you start a database, a system global area (SGA) is allocated and the Oracle processes are started. The SGA is the area of memory used for database information shared by all database users. The Oracle processes and the SGA create what is called an Oracle instance.

intent logging

A logging scheme that records pending changes to a file system structure. These changes are recorded in an *intent log*.

interrupt key

A way to end or break out of any operation and return to the system prompt by pressing Ctrl-C.

kilobyte

One thousand bytes. For technical specifications, it refers to 1,024 bytes. In general usage, it sometimes refers to an even one thousand bytes. Also KB, Kbyte and K-byte.

kernel asynchronous I/O

A form of I/O that performs non-blocking system level reads and writes.
This enables the system to handle multiple I/O requests simultaneously.

large file

A file more than two gigabytes in size. An operating system that uses a 32-bit signed integer to address file contents will not support large files; however, the Version 4 disk layout feature of VxFS supports file sizes of up to two terabytes.

large file system

A file system more than two gigabytes in size. VxFS, in conjunction with VxVM, supports large file systems.

latency

The amount of time it takes for a given piece of work to be completed. For file systems, this typically refers to the amount of time it takes a given file system operation to return to the user. Also commonly used to describe disk seek times.

load balancing

The tuning of a computer system, network tuning, or disk subsystem in order to more evenly distribute the data and/or processing across available resources. For example, in clustering, load balancing might distribute the incoming transactions evenly to all servers, or it might redirect them to the next available server.

load sharing

The division of a task among several components without any attempt to equalize each component's share of the load. When several components are load sharing, it is possible for some of the shared components to be operating at full capacity and limiting performance, while others components are under utilized.

Logical Unit Number

A method of expanding the number of SCSI devices that can be placed on one SCSI bus. Logical Unit Numbers address up to seven devices at each SCSI ID on an 8-bit bus or up to fifteen devices at each ID on a 16-bit bus.

logical volume

See “volume.”

LUN

See “Logical Unit Number.”

master node

A computer which controls another computer or a peripheral.

megabyte

One million bytes, or more precisely 1,048,576 bytes. Also MB, Mbyte and M-byte.

metadata

Data that describes other data. Data dictionaries and repositories are examples of metadata. The term may also refer to any file or database that holds information about another database's structure, attributes, processing, or changes.

mirror

A duplicate copy of a volume and the data therein (in the form of an ordered collection of subdisks). Each mirror is one copy of the volume with which the mirror is associated.

The terms *mirror* and *plex* can be used synonymously.

mirroring

A layout technique that mirrors the contents of a volume onto multiple plexes. Each plex duplicates the data stored on the volume, but the plexes themselves may have different layouts.

mount point

The directory path name at which a file system attaches to the file system hierarchy.

multithreaded

Having multiple concurrent or pseudo-concurrent execution sequences. Used to describe processes in computer systems. Multithreaded processes are one means by which I/O request-intensive applications can use independent access to volumes and disk arrays to increase I/O performance.

NBU

See "VERITAS NetBackup (NBU)."

node

One of the hosts in a cluster.

object (VxVM)

An entity that is defined to and recognized internally by the VERITAS Volume Manager. The VxVM objects include volumes, plexes, subdisks, disks, and disk groups. There are two types of VxVM disk objects—one for the physical aspect of the disk and the other for the logical aspect of the disk.

OLTP

See "Online Transaction Processing."

online administration

An administrative feature that allows configuration changes without system or database down time.

Online Transaction Processing

A type of system designed to support transaction-oriented applications. OLTP systems are designed to respond immediately to user requests and each request is considered to be a single transaction. Requests can involve adding, retrieving, updating or removing data.

paging

The transfer of program segments (pages) into and out of memory. Although paging is the primary mechanism for virtual memory, excessive paging is not desirable.

parity

A calculated value that can be used to reconstruct data after a failure. While data is being written to a RAID-5 volume, parity is also calculated by performing an *exclusive OR* (XOR) procedure on data. The resulting parity is then written to the volume. If a portion of a RAID-5 volume fails, the data that was on that portion of the failed volume can be recreated from the remaining data and the parity.

partition

The logical areas into which a disk is divided.

persistence

Information or state that will survive a system reboot or crash.

plex

A duplicate copy of a volume and its data (in the form of an ordered collection of subdisks). Each plex is one copy of a volume with which the plex is associated. The terms *mirror* and *plex* can be used synonymously.

preallocation

Prespecifying space for a file so that disk blocks will physically be part of a file before they are needed. Enabling an application to preallocate space for a file guarantees that a specified amount of space will be available for that file, even if the file system is otherwise out of space.

Quick I/O

Quick I/O presents a regular VERITAS File System file to an application as a raw character device. This allows Quick I/O files to take advantage of kernel-supported asynchronous I/O and direct I/O to and from the disk device, as well as bypassing the UNIX single-writer lock behavior for most file system files.

Quick I/O file

A regular UNIX file that is accessed using the Quick I/O naming extension (`::cdev:vxfs:`).

RAID

A Redundant Array of Independent Disks (RAID) is a disk array set up with part of the combined storage capacity used for storing duplicate information about the data stored in that array. This makes it possible to regenerate the data if a disk failure occurs.

redo log files

Redo log files record transactions pending on a database. If a failure prevents data from being permanently written to datafiles, changes can be obtained from the redo log files. Every Oracle database has a set of two or more redo log files.

recovery mode

The process of recovering the file systems once the file systems have been restored. In a media failure scenario, the backups must be restored before the database can be recovered.

repository

A repository holds the name, type, range of values, source, and authorization for access for each data element in a database. Pertinent information, needed to display configuration information and interact with the database, is stored in VxDBA's repository. The database maintains a repository for administrative and reporting use.

root disk

The disk containing the root file system.

root disk group

A special private disk group that always exists on the system. The root disk group is named `rootdg`.

root file system

The initial file system mounted as part of the UNIX kernel startup sequence.

script

A file, containing one or more commands that can be run to perform processing.

shared disk group

A disk group in which the disks are shared by multiple hosts (also referred to as a *cluster-shareable disk group*).

sector

A minimal unit of the disk partitioning. The size of a sector can vary between systems. A sector is commonly 512 bytes.

segment

Any partition, reserved area, partial component, or piece of a larger structure.

SGA

See "System Global Area."

single threading

The processing of one transaction to completion before starting the next.

slave node

A node that is not designated as a master node.

slice

The standard division of a logical disk device. The terms *partition* and *slice* can be used synonymously.

snapped file system

A file system whose exact image has been used to create a snapshot file system.

snapped volume

A volume whose exact image has been used to create a snapshot volume.

snapshot

A point-in-time image of a volume or file system that can be used as a backup.

snapshot file system

An exact copy of a mounted file system, at a specific point in time, that is used for online backup. A snapshot file system is not persistent and it will not survive a crash or reboot of the system.

snapshot volume

An exact copy of a volume, at a specific point in time. The snapshot is created based on disk mirroring and is used for online backup purposes.

spanning

A layout technique that permits a volume (and its file system or database) too large to fit on a single disk to distribute its data across multiple disks or volumes.

Storage Checkpoint

An efficient snapshot technology for creating a point-in-time image of a currently mounted VxFS file system. A Storage Checkpoint presents a consistent, point-in-time view of the file system by identifying and maintaining modified file system blocks. The Storage Checkpoint facility is an enabling technology for Block-Level Incremental (BLI) Backup and Storage Rollback.

Storage Rollback

On-disk restore capability for faster recovery from logical errors, such as accidentally deleting a file. Because each Storage Checkpoint is a point-in-time image of a file system, Storage Rollback simply restores or rolls back a file or entire file system to a Storage Checkpoint.

stripe

A set of stripe units that occupy the same positions across a series of columns in a multi-disk layout.

stripe unit

Equally sized areas that are allocated alternately on the subdisks (within columns) of each striped plex. In an array, this is a set of logically contiguous blocks that exist on each disk before allocations are made from the next disk in the array.

stripe unit size

The size of each stripe unit. The default stripe unit size for VxVM is 32 sectors (16K). For RAID 0 stripping, the stripe unit size is 128 sectors (64K). For VxVM RAID 5, the stripe unit size is 32 sectors (16K). A *stripe unit size* has also historically been referred to as a *stripe width*.

striping

A layout technique that spreads data across several physical disks using stripes. The data is allocated alternately to the stripes within the subdisks of each plex.

subdisk

A consecutive set of contiguous disk blocks that form a logical disk segment. Subdisks can be associated with plexes to form volumes.

superuser

A user with unlimited access privileges who can perform any and all operations on a computer. In UNIX, this user may also be referred to as the “root” user. On Windows/NT, it is the “Administrator.”

System Global Area

The area of memory used for database information shared by all database users. Each SGA contains the data and control information for a single Oracle instance.

tablespace

In an Oracle database, an allocation of space used to hold schema objects (triggers, stored procedures, tables, etc.). A tablespace is associated with one or more datafiles.

terabyte

Shorthand for 1,000,000,000,000 (10^{12}) bytes (or approximately 1000 GB).

throughput

A measure of work accomplished in a given amount of time. For file systems, this typically refers to the number of I/O operations in a given period of time.

UFS

The Solaris name for a file system type derived from the 4.2 Berkeley Fast File System.

unbuffered I/O

I/O that bypasses the file system cache for the purpose of increasing I/O performance (also known as *direct I/O*).

VERITAS Enterprise Administrator

Application that is required to access graphical user interface (GUI) functionality.

VERITAS Extension for Oracle Disk Manager

A feature of the VERITAS Database Edition for Oracle that lets Oracle create and manage database storage, as well as performing I/Os in a file system without the performance degradation typically associated with running databases on file systems.

VERITAS File Replicator (VFR)

An enterprise data replication solution used to distribute Web or file server data. It enables multi-host processing and protects against critical data loss.

VERITAS NetBackup (NBU)

A product that lets you back up, archive, and restore files, directories, or raw partitions that reside on your client system.

VERITAS Volume Replicator (VVR)

A feature of VERITAS Volume Manager, VVR is a data replication tool designed to contribute to an effective disaster recovery plan.

volume

A logical disk device that appears to applications, databases, and file systems as a physical disk partition. A logical disk can encompass multiple or one to many physical volumes.

volume layout

A variety of layouts that allows you to configure your database to meet performance and availability requirements. This includes spanning, striping (RAID-0), mirroring (RAID-1), mirrored stripe volumes (RAID-0+1), striped mirror volumes (RAID-1+0), and RAID 5.

volume manager objects

Volumes and their virtual components. See "object (VxVM)."

VFR

See "VERITAS File Replicator (VFR)."

VVR

See "VERITAS Volume Replicator (VVR)."

VxDBA

A VERITAS Storage Foundation *for Oracle* menu-driven utility or graphical user interface (GUI) that helps you manage your database environment.

vxfs or VxFS

The acronym for VERITAS File System.

vxvm or VxVM

The acronym for VERITAS Volume Manager.

Index