Usage Guide for
Sun StorEdge™ Availability Suite 3.2
Point-in-Time Copy Software
With Oracle Databases

Submit comments about this document at: http://www.sun.com/hwdocs/feedback

Please
Recycle

Adobe PostScript

Contents

Overview

Chapter contents:

- "Volume Groups" on page 3
- "Shadow Volume Set Types" on page 4
- "Point-in-Time Copy Software Features" on page 6

Sun StorEdge™ Availability Suite Software point-in-time copy software is a *point-in-time copy snapshot* utility that can be used with Oracle 8 to create database copies that are available for use almost immediately. For example, if you are required to provide a copy of daily transactions for accounting purposes, you need only create a snapshot of the data with the point-in-time copy software. This snapshot is available almost instantly for accounting purposes, is related to an exact point in time (close-of-business, for instance) even though Oracle transactions continue, and is available with little or no impact on ongoing database operations. You only need to quiesce Oracle long enough to issue a point-in-time copy software enable, update, or copy command.

Point-in-time copy snapshots are referred to as *shadow volumes* in the point-in-time copy software. You structure an Oracle database volume into a point-in-time copy software *shadow volume set*, which includes a *master volume* (the database volume), a *shadow volume*, and a *bitmap volume*. The shadow volume is the copy of the master volume that is independently accessible by other applications besides Oracle. The bitmap volume tracks changes to both the master volume and the shadow volume. The point-in-time copy software uses the information in the bitmap volume to perform updates of the shadow volume or the master volume.

In addition to providing instantly available and easily updated point-in-time copies of Oracle volumes, the point-in-time copy software includes additional, related functionality that helps extend the software's capabilities and ease-of-use.

- Supports *I/O volume groups* for simplified management of storage volumes
- Supports *compact dependent shadow volumes* for more efficient use of available storage
- Enables you to *export* a shadow volume for processing external to the originating host
- Enables you to maintain multiple shadow volumes of a single master volume for processing multiple instances of the data

The point-in-time copy software works with the remote mirror software, which is also part of the Sun StorEdge Availability Suite Software. The remote mirror software helps enable remote shadowing almost anywhere in the world.

This Usage Guide discusses all of these features with regard to Oracle database software.

Oracle Storage Compatibility Program

Oracle provides a suite of tests under the Oracle Storage Compatibility Program (OSCP) that are used to validate the compatibility of point-in-time copy software solutions with Oracle databases.

These tests were performed in various scenarios with the Sun StorEdge Availability Suite Software point-in-time copy software. This document provides guidelines for using this point-in-time copy software with Oracle databases to perform the OSCP test suite.

Volume Groups

A point-in-time copy shadow volume set includes a master volume, a shadow volume, a bitmap volume, and an optional overflow volume. The exact relationship between the master volume and the shadow volume is dictated by the type of shadow volume set that you enable. In all cases, the shadow volume provides access to data within the master volume that is referenced to a particular point-in-time: the time at which you issued the enable, update, or copy command to the shadow volume set.

You can collect shadow volume sets of all types into named I/O volume groups. After you have created these I/O volume groups, which are composed of one or more shadow volume sets, you can issue commands to the I/O volume group and have the commands executed on all included shadow volume sets at once. In this way, you can cause multiple snapshots of many shadow volume sets to have the same time stamp. Most point-in-time copy software commands function as group commands.

Shadow Volume Set Types

The point-in-time copy software enables you to create shadow volume sets in several configurations, each of which has its own advantages.

- *Independent shadow volume set*
- *Dependent shadow volume set*
- Compact dependent shadow volume set
- Compact dependent shadow volume group with attached *overflow volume*

Independent Shadow Volume Set

You create, with the *enable* command, an independent shadow volume set whenever you want to have a full copy of the data. You might want the shadow volume to stand alone as a backup; you might want reads of the shadow volume not to affect transactions at the master volume; or you might want to export the shadow volume to another host entirely. Enable an independent shadow volume set in any situation in which you require a full, independent copy of the data that is referenced to a point in time. An independent shadow volume requires the same amount of disk space as the master volume.

When you enable an independent shadow volume set, a full copy of the data is started from master volume to shadow volume. Even though the copy is not completed, the master and shadow volumes are available, but the shadow is treated as a dependent shadow volume until the copy is complete. Once the copy is complete, you can disassociate the volumes in an independent shadow volume set with the *disable* command. Disabling releases the shadow volume from its master volume so you can use it as a complete, independent copy of the data.

Dependent Shadow Volume Set

A dependent shadow volume depends on the master volume to service read requests for data that has not changed at the master. Because of this, a dependent shadow volume must remain enabled with its master volume. When data at the master volume changes, the old data is written to the dependent shadow volume before the new data is written to the master volume. When a read request is received for this older data (data related to a point in time), the request is serviced from the shadow volume.

Enable a dependent shadow volume set when:

- You want to have a snapshot of the data
- There is no problem with using primary host resources to service read requests
- The master volume and the shadow volume can remain in an enabled shadow volume set

Compact Dependent Shadow Volume Set

A compact dependent shadow volume set is a dependent shadow volume set that has less storage space allotted to its shadow volume than to its master volume. Using compact dependent shadow volume sets enables efficient use of storage space when few changes to the master volume are expected. The compact dependent shadow volume can be any size. When you enable a dependent shadow volume set with a shadow volume that is smaller than the master volume, a compact dependent shadow volume is created.

Use a compact dependent shadow volume set when you can predict the volume of writes to the master volume over the period of time that you expect to be using the volume set before updating the volumes. However, be aware that attempted data writes to a full compact dependent shadow volume will result in the compact dependent shadow volume set being marked inconsistent.

Compact Dependent Shadow Volume Set With an Attached Overflow Volume

A compact dependent shadow volume is always open to becoming filled up unexpectedly. To prevent this occurrence from causing difficulty, you can assign an overflow volume to a compact dependent shadow volume set. If the compact dependent shadow volume becomes filled, any further writes are directed to the overflow volume, which prevents the data in the volume set from becoming inconsistent.

Any number of compact dependent shadow volume sets can be attached to a single overflow volume. However, attempted data writes to a full overflow volume will result in the data being labeled inconsistent and the volume set being taken offline.

It is good practice to assign all your compact dependent shadow volume sets to overflow volumes.

Point-in-Time Copy Software Features

Regardless of the type of shadow volume set that you choose to enable, your goal is to have a master volume that is an Oracle database volume, a shadow volume that provides access to this master data referenced to a chosen point-in-time, and a bitmap volume that the point-in-time copy software uses to perform quick updates. In addition, if you enabled compact dependent shadow volume sets, you should have overflow volumes attached to these sets to prevent your data from becoming inconsistent if you experience an unexpected volume of data writes.

You can now perform the following basic tasks:

- Update the shadow volume from the master volume (new snapshot)
- Update the master volume from the shadow volume (possibly disaster recovery)
- Access current data from the master volume
- Access snapshot data from the shadow volume (analysis)

In addition to the baseline point-in-time copy software capabilities, there are a number of other related capabilities that extend and expand the usage of the Sun StorEdge Availability Suite Software point-in-time copy software

Exporting Shadow Volumes

You can export an independent shadow volume if it is located physically on a dual-ported disk drive. If a shadow is exported from its shadow volume group, it can be imported by another networked host and used for any purpose. The shadow volume can subsequently be rejoined to its original master volume using bitmap data that the importing host is required to maintain.

By exporting a shadow volume so that another host can import it, you can switch the processing of Oracle data from the first host to the other host. Now you can perform maintenance of the master volume's host or simply let the workload pass to another machine. Either way, the Oracle database users see little disruption in their service. Later, when the master volume and the shadow volume are rejoined, the bitmap data from the importing host is used to update the master volume.

Another reason to export a shadow volume might be that the analysis you want to perform would present too big an I/O load to the shadow volume's original host. By exporting it to another host, you can perform the analysis on the data contained in the shadow volume without presenting any I/O load to the first host. The master volume can continue to process Oracle database transactions and maintain a record of those changes in its bitmap for later updating of the shadow volume. The Oracle users see no disruption.

Multiple Shadow Volumes

With the point-in-time copy software, you can create any number of shadow volume sets of any type using the same master volume. This means that a master volume can have multiple shadow volumes. Each of these shadow volumes performs just as though it were the only shadow volume. In other words, the presence of other shadow volumes has no effect on any shadow volume or on the master volume beyond additional I/O volume.

Using multiple shadow volumes, you can have any number of tasks running with the same or different sets of point-in-time data from the same master volume. The only limit would be system resources. Practically speaking, you wouldn't have a huge number of shadow volumes configured to a single master volume because of limited system resources. However, having a dependent shadow volume that is updated daily for analysis coexist with an independent shadow volume of the same master for use in offloading one host in peak hours would be a reasonable use for the point-in-time copy software's multiple shadow capability.

Configuration and Installation

Chapter contents:

System Requirements

This section provides the hardware and software requirements for running the OSCP tests.

Point-in-Time Copy Software Requirements

Hardware	A CD-ROM drive connected to the host server where the Sun StorEdge Availability Suite Software point-in-time copy software is to be installed.
	The Sun StorEdge Availability Suite Software is supported on server hosts using the Solaris™ Operating Environment. Hosts include but are not limited to: • Sun Enterprise™ Server models 2X0 through 4x0 • Sun Enterprise Server models 3X00 through 10000
	Disk space: • The point-in-time copy software requires approximately 1 Mbyte • Supporting packages require approximately 5 Mbytes
	If you plan to export shadow volumes, you must store the shadow volume on a dual-ported drive

Software	Solaris 8 and Solaris 9u3 and later Operating Environment or a subsequent compatible version
	Sun StorEdge Availability Suite 3.2 Software CD

Oracle Software Requirements

Hardware	Approximately 2 Gbytes of disk space for Oracle server software binaries only

Software	Oracle 8 and 8i

Installing Oracle

The following information is provided as a reference to install the Oracle binaries. See the *Oracle Installation Guide* for installation details.

▼ To Prepare for Oracle Installation

You must perform the following steps before you install the Oracle binaries.

1. **Add the following entries to the** `/etc/system` **file by using a text editor:**

```
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmns=200
set semsys:seminfo_semmni=100
set semsys:seminfo_semmsl=100
set semsys:seminfo_semopm=100
set semsys:seminfo_semvmx=32767
```

2. **Create the directory** `/opt/bin`.

3. **Create the group** `oinstall`.

4. **Create the user** `oracle` **and attach the** `oinstall` **group to** `oracle`.

5. **Have a mount point ready and make** `oracle` **the owner.**

6. **Reboot the server.**

▼ To Install Oracle

1. **Log into the server as user** `oracle`.

2. **Insert the Oracle 8.x Universal Install CD into the CD-ROM drive.**

3. **Accept the 8i Server Installation option and follow the instructions.**

Oracle Installation Notes

- You can use the path `/oracle/8.`x`.`x (where x is a version number) as the `ORACLE_HOME` environment variable. The software then creates various directories under `$ORACLE_HOME` — `/bin` contains all binaries and `/rdbms/admin` contains all utility SQL files.

- The installation software automatically creates a default database named `starter`. You must assign a four-character `ORACLE_SID` system identifier (SID) name to the database.

- When the installation is complete, you can create numerous Oracle databases on the server. The number of databases per server is restricted to the resource availability. You can create databases using the `dbassist` utility.

Installing the Point-in-Time Copy Software

If necessary, install the Sun StorEdge Availability Suite Software point-in-time copy software. Follow the instructions in the documents supplied with the software.

Application	Title	Part Number
Installation	*Sun StorEdge Availability Suite 3.2 Software Installation Guide*	817-2783-10
Release Information	*Sun StorEdge Availability Suite 3.2 Software Release Notes*	817-2782-10
System Administration	*Sun StorEdge Availability Suite 3.2 Point-in-Time Copy Software Administrator and Operator Guide*	817-2781-10

man pages	`iiadm`
	`dscfg`
	`pkgrm`
	`pkgask`
	`scmadm`
	`pkgadd`
		N/A

Using Point-in-Time Copy Software with Oracle Software

Chapter contents:

For testing, development, and reporting purposes, you ordinarily use a copy of an Oracle database, not the master database volumes. Using point-in-time copy software, you can easily and quickly acquire a copy, which is called a snapshot, of the Oracle database whenever you choose.

You can acquire a snapshot of your database in two Oracle database modes: *hot-backup mode* and *cold-backup mode*. You only have to put the tablespaces into hot-backup mode for as long as it takes to initiate the point-in-time copy software's enable, copy, or update function, which enables the snapshot. You can release the tablespaces from hot-backup mode immediately without waiting for the copy to complete. Therefore, the tablespaces are in hot-backup mode for only a short time. In cold-backup mode, you shut down the database before initiating the point-in-time copy enable, copy, or update.

You can use the snapshot for many purposes, including the following:

- Backing up your database to tape without causing any performance degradation to the master volumes
- Launching another instance on a different node (if dual-ported)
- Restoring your database easily if it becomes corrupted (you can only restore the volume that was corrupted)
- Replicating your database to a system at a different location using Sun StorEdge Availability Suite Software remote mirror software

See Chapter 1 for an introductory discussion of the point-in-time copy software. See particularly "Volume Groups" on page 3 and following sections for a discussion of dependent and independent shadow volume sets.

When you first enable an independent shadow volume set, it will take a longer time to complete the full copy than it will for subsequent resynchronizations of the shadow volume with the Oracle database. When you issue an update command to resynchronize the shadow volume with the master volume, only data that has been changed in the master volume since the last update is copied to the shadow volume. Therefore, an update can be much faster than the initial copy. This updated shadow volume can be used for backup or for testing purposes.

Dependent mode is the fastest way to create a copy of the database. However, even though any read of the shadow volume is done at the shadow volume, most of the actual I/O is performed on the master volume. This may impact performance at the master volume to some degree. However, a dependent shadow volume is useful for quickly launching another snapshot of the database or to do a backup at a particular point in time.

You can export a shadow volume if it is part of an independent shadow volume set and if it resides on a dual-ported disk drive. Once exported with the point-in-time copy software's export command, another host can import the shadow volume and use it without affecting the original host. Exporting a shadow volume is useful if you want to perform intensive analysis on the snapshot, but you don't want to steal cycles from the Oracle host.

You can group any number of shadow volume sets in an I/O volume group and execute atomic commands on all the volume sets at once. For instance, if you have five shadow volume sets and you want to take a point-in-time copy snapshot of all the shadow volume sets at the same time, group them together and execute one update command against the I/O volume group. The point-in-time copy snapshot on the shadow volume of each member of the I/O volume group will have the same time stamp.

Considerations

- The point-in-time copy software supports multiple volume snapshots. Therefore you can distribute the data across multiple volumes. You can use UFS, VxFS, or raw devices for the data files.
- The shadow volume in an independent shadow volume set must be the same size as its master volume, or larger.
- The shadow volume in a dependent shadow volume set can be the same size as its master volume or it can be smaller than its master volume, in which case it is called a compact dependent shadow volume.
- Many compact dependent shadow volumes can be attached to an overflow volume. In the event that the compact dependent shadow volume becomes full, further writes are directed to the overflow volume.

Performing a Hot Backup Procedure

The general steps to perform a hot backup procedure are as follows:

▼ To Perform a Hot Backup Procedure

1. **Choose as the master volumes the volumes where the production data files, control files, redo logs, and archived logs reside.**

2. **Create shadow volumes the same size as the master volumes.**

3. **Create the bitmap volume for each point-in-time copy software shadow volume set.**

4. **Assuming the database is running, ensure the following:**
   - The shadow volumes are not mounted
   - The volumes are in logging mode (under /etc/vfstab control)

5. **Perform the following steps from the SQL*Plus prompt.**

   a. **Place all the tablespaces in hot backup mode (refer** v$tablespaces**).**

   ```
   SQL> alter tablespace tablespacename begin backup;
   ```

   b. **Take a SQL copy of the control file.**

   ```
   SQL> alter database backup controlfile to filename;
   ```

   c. **Archive the latest redo logs (refer** v$logfile, v$log**).**

   ```
   SQL> alter system switch logfile;
   ```

6. **Freeze the I/O for the master volumes.**

   If UFS is used, then lock the file systems for these volumes. For example:

   ```
   # lockfs -af
   # lockfs -f -w master_volume_mount-point
   ```

   If you are using either VxFS, use the following commands:

   ```
   # sync
   SQL> alter system suspend;
   ```

   If you are using raw volumes or slices, use the following command:

   ```
   SQL> alter system suspend;
   ```

7. **Issue the command to enable the point-in-time copy volume set, which takes the snapshot.**

   ■ Independent mode:

   ```
   # iiadm -e ind master_vol shadow_vol bitmap_vol
   ```

   ■ Dependent mode:

   ```
   # iiadm -e dep master_vol shadow_vol bitmap_vol
   ```

8. **When the prompt returns, unlock the file systems.**

   If you are using UFS, issue the following command:

   ```
   # lockfs -u master_volume_mount-point
   ```

   If you are using either VxFS or raw volumes or slices, use the following command:

   ```
   SQL> alter system resume;
   ```

9. **From the SQL*Plus prompt, issue the following commands for all tablespaces to end the hot backup mode (refer** v$tablespace**).**

```
SQL> alter tablespace tablespacename end backup;
```

10. **Check the status of the snapshot by issuing the** iiadm -i all **command.**

   ■ A copy-in-progress message is similar to the following:

```
/dev/vx/rdsk/oracle816/ii_dbs_m: (master volume)
/dev/vx/rdsk/oracle816/ii_dbs_s: (shadow volume)
/dev/vx/rdsk/oracle816/ii_dbs_bm: (bitmap volume)
Independent Copy, copy in progress, copying master to shadow
Percent of bitmap set: 100
```

   ■ An independent-copy-completed message is similar to the following:

```
/dev/vx/rdsk/oracle816/ii_dbs_m: (master volume)
/dev/vx/rdsk/oracle816/ii_dbs_s: (shadow volume)
/dev/vx/rdsk/oracle816/ii_dbs_bm: (bitmap volume)
Independent Copy
Percent of bitmap set: 0
```

   ■ A dependent-copy-in-completed message is similar to the following:

```
/dev/vx/rdsk/oracle816/ii_dbs_m: (master volume)
/dev/vx/rdsk/oracle816/ii_dbs_s: (shadow volume)
/dev/vx/rdsk/oracle816/ii_dbs_bm: (bitmap volume)
Dependent Copy
Percent of bitmap set: 0
```

Using the Shadow Volume Resulting from a Hot Backup

TABLE 3-1 shows how you can use the shadow volume.

TABLE 3-1   Using the Shadow Volume Resulting from a Hot Backup

Volume Type	Usage
Any shadow volume	• Copy the shadow volume contents to the master volume in case of production data problems • Update the shadow volume from the master volume with new data periodically • Use the shadow volume for backup purposes
Independent shadow volume	Once the copy is completed: • Split the master and shadow volume pair to open the database with a different instance name • Split the master and shadow volume pair to copy the independent copy volume to a remote site to create another instance of the volume • Export the shadow to another host for processing if it resides on a dual-ported disk drive.
Dependent shadow volume	Once the copy is completed: • The shadow volume is available immediately as a snapshot copy • *Do not* split the volume pair or disable the point-in-time copy software • Use the shadow volume for backup purposes

▼ To Update the Shadow Volume

1. Unmount the shadow volumes (if file system is used).

```
# umount shadow_vol
```

2. Update the shadow volume.

```
# iiadm -u s shadow_vol
```

3. Check the status of the snapshot by issuing the `iiadm -i all` **command.**

▼ To Restore Full Database From the Shadow Volume to the Master Volume

1. If the database is running, stop the database.

```
SQL> shutdown immediate;
```

2. Unmount the master volumes (if UFS is used).

```
# umount master_vol_mount-point
```

3. Issue the reverse update command to restore data from shadow to master.

```
# iiadm -c m shadow_vol
```

Note – Use a complete copy (`-c`) or update copy (`-u`) command.

4. Check the status of the snapshot by issuing the `iiadm -i all` **command.**

5. When the update is complete, check the file system and mount the volumes.

```
# fsck master_vol
# mount master_vol_mount-point
```

6. **Bring the database to the mounted state from the SQL*Plus prompt.**

```
SQL> startup mount initsid.ora;
```

7. **If the datafile locations are the same as the original database, perform these steps:**

   a. **Recover the database.**

   ```
   SQL> recover automatic database;
   ```

   b. **When recovery is complete, open the database as follows:**

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

8. **If the datafile locations are different from the original database, perform these steps:**

   a. **Rename the datafiles in the new location (not the original database).**

   ```
   SQL> alter database rename file 'original_file_name' to 'new_file_name';
   ```

   b. **Recover the database.**

   ```
   SQL> recover automatic database;
   ```

   c. **When the recovery is complete, open the database as follows:**

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

▼ To Restore and Recover Selective Datafiles

1. **From the SQL*Plus prompt, take the tablespaces offline.**

   Do the following for the tablespaces that need to be restored and recovered.

   ```
   SQL> alter tablespace tablespacename offline;
   ```

2. **Unmount the volumes if the filesystem is being used.**

```
# umount mount_point
```

3. **Reverse sync only those volumes that contain the required datafiles.**

   Repeat the following for the datafiles for all tablespaces (refer v$datafile, v$tablespace).

```
# iiadm -c m shadow_vol_to_be_restored
```

Note – Use a complete copy (-c) or update copy (-u) command.

---

4. **Check the status of the snapshot operation.**

```
# iiadm -i shadow_vol_to_be_restored
```

5. **Once the update is complete, check and mount the filesystem.**

```
# fsck master_vol
# mount master_vol_mount_point
```

6. **Recover the tablespaces from the SQL*Plus prompt, and repeat this command for each** *tablespacename***.**

```
SQL> recover automatic tablespace tablespacename;
```

7. **Bring the tablespace online from the SQL*Plus prompt, and repeat this command for each** *tablespacename***.**

```
SQL> alter tablespace tablespacename online;
```

Performing a Cold Backup Procedure

These are the general steps to perform a cold backup procedure.

▼ To Perform a Cold Backup Procedure

1. **Choose as the master volumes the volumes where the production data files, control files, redo logs, and archived logs reside.**

2. **Create the shadow volumes of the same size of the master volumes.**

3. **Create the bitmap volume for each point-in-time copy software volume pair.**

4. **Assuming the database is running, ensure the following:**
   - The shadow volumes are not mounted
   - The volumes are in logging mode (under /etc/vfstab control)

5. **Shut down the database.**

   ```
   SQL> shutdown normal;
   ```

6. **If UFS is used, then lock the file systems for these volumes.**

   For example:

   ```
   # lockfs -f -w master_volume_mount-point
   ```

7. **Issue the command to enable the point-in-time copy volume set, which takes the snapshot.**
   - Independent mode:

   ```
   # iiadm -e ind master_vol shadow_vol bitmap_vol
   ```

   - Dependent mode:

   ```
   # iiadm -e dep master_vol shadow_vol bitmap_vol
   ```

8. **When the prompt returns, unlock the file systems.**

   For example:

   ```
   # lockfs -u master_volume_mount-point
   ```

9. **Check the status of the snapshot by issuing the** `iiadm -i all` **command.**

   ■ A copy-in-progress message is similar to the following:

   ```
   /dev/vx/rdsk/oracle816/ii_dbs_m: (master volume)
   /dev/vx/rdsk/sifydg/dbs_125: (shadow volume)
   /dev/vx/rdsk/oracle816/ii_dbs_bm: (bitmap volume)
   Independent Copy, copy in progress, copying master to shadow
   Percent of bitmap set: 100
   ```

   ■ An independent-copy-completed message is similar to the following:

   ```
   /dev/vx/rdsk/oracle816/ii_dbs_m: (master volume)
   /dev/vx/rdsk/oracle816/ii_dbs_s: (shadow volume)
   /dev/vx/rdsk/oracle816/ii_dbs_bm: (bitmap volume)
   Independent Copy
   Percent of bitmap set: 0
   ```

   ■ A dependent-copy-completed message is similar to the following:

   ```
   /dev/vx/rdsk/oracle816/ii_dbs_m: (master volume)
   /dev/vx/rdsk/oracle816/ii_dbs_s: (shadow volume)
   /dev/vx/rdsk/oracle816/ii_dbs_bm: (bitmap volume)
   Dependent Copy
   Percent of bitmap set: 0
   ```

Using the Shadow Volume Resulting from a Cold Backup

TABLE 3-1 shows how you can use the shadow volume.

TABLE 3-2   Using the Shadow Volume Resulting from a Cold Backup

Volume Type	Usage
Any shadow volume	• Copy the shadow volume contents to the master volume in case of production data problems • Update the shadow volume from the master volume with new data periodically • Use the shadow volume for backup purposes
Independent shadow volume	Once the copy is completed: • Split the master and shadow volume pair to open the database with a different instance name • Copy the independent copy volume to a remote site with the remote mirror software to create another instance of the volume • Export the shadow to another host for processing if it resides on a dual-ported disk drive.
Dependent shadow volume	Once the copy is completed: • The shadow volume is available immediately as a snapshot copy • *Do not* split the volume pair or disable the point-in-time copy software • Use the shadow volume for backup purposes

▼ To Update the Shadow Volume

1. Unmount the shadow volumes (if UFS is used).

```
# umount shadow_vol
```

2. Update the shadow volume.

```
# iiadm -u s shadow_vol
```

3. Check the status of the snapshot by issuing the `iiadm -i all` **command.**

▼ To Restore Data from The Shadow Volume to the Master Volume

1. Unmount the master volumes (if UFS is used).

```
# umount master_vol_mount-point
```

2. Issue the reverse update command to restore data from shadow to master.

```
# iiadm -c m shadow_vol
```

Note – Use a complete copy (-c) or update copy (-u) command.

3. Check the status of the snapshot by issuing the `iiadm -i all` **command.**

4. When the update is complete, check the file system and mount the volumes.

```
# fsck master_vol
# mount master_vol_mount-point
```

5. Bring the database to the mounted state from the SQL*Plus prompt.

```
SQL> startup mount initsid.ora;
```

6. **If the datafile locations are the same as the original database, perform these steps:**

   a. **Recover the database.**

   ```
   SQL> recover automatic database;
   ```

   b. **When recovery is complete, open the database as follows:**

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

7. **If the datafile locations are different from the original database, perform these steps:**

   a. **Rename the datafiles in the new location (not the original database).**

   ```
   SQL> alter database rename file 'original_file_name' to 'new_file_name';
   ```

   b. **Recover the database.**

   ```
   SQL> recover automatic database;
   ```

   c. **When the recovery is complete, open the database as follows:**

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

Launching the Shadow on a Different Host

▼ To Launch Another Instance on a Secondary Host

On the Primary Host

1. **Choose as the master volumes the volumes where the production data files, control files, redo logs, and archived logs reside.**

2. **Create the shadow volumes of the same size of the master volumes.**

3. **Create the bitmap file or volume for each point-in-time copy software volume pair.**

4. **Place the primary database in hot-backup mode (if database is running).**

5. **Unmount the shadow volume.**

```
# umount mount_point
```

6. **Export the shadow volume.**

```
# iiadm -E shadow_vol
```

7. **Create *bitmap2* to be the same size as *bitmap1*.**

8. **Copy bitmap 1 to bitmap2.**

```
# dd if=bitmap1 of=bitmap2
```

9. **If VxVM or SDS(SLVM), deport *shadow_vol* and *bitmap2*.**

On the Secondary Host

10. **If VxVM or SDS(SLVM), import** *shadow_vol* **and** *bitmap2*.

11. **Import the shadow volume.**

```
# iiadm -I shadow_vol  bitmap2
```

12. **Mount the shadow volume.**

```
# mount shadow_vol  mount_point
```

13. **Bring the database to the mounted state from the SQL*Plus prompt.**

```
SQL> startup mount initsid.ora;
```

14. **If the datafile locations are the same as the original database, perform these steps:**

   a. **Recover the database.**

```
SQL> recover automatic database;
```

   b. **When recovery is complete, open the database as follows:**

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

15. **If the datafile locations are different from the original database, perform these steps:**

   a. **Rename the datafiles in the new location (not the original database).**

```
SQL> alter database rename file 'original_file_name' to 'new_file_name';
```

   b. **Recover the database.**

```
SQL> recover automatic database;
```

c. **When the recovery is complete, open the database as follows:**

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

▼ To Switch Back to the Primary Host

On the Secondary Host

1. **Stop the application using the shadow volume.**

2. **Unmount the shadow volume.**

```
# umount mount_point
```

3. **Disable the shadow volume.**

```
# iiadm -d shadow_vol
```

4. **If VxVM or SDS(SLVM), deport** *shadow_vol* **and** *bitmap2***.**

On the Primary Host

5. **If VxVM or SDS(SLVM), import** *shadow_vol* **and** *bitmap2***.**

6. **Join the shadow volume.**

```
# iiadm -J shadow_vol bitmap2
```

Methodologies

Chapter contents:

Application Requirements

Before you start planning the actual point-in-time copy software configuration, be sure that you understand your application requirements for consistency and application recovery methods. Sun Microsystems recommends that you make a full copy of the entire database at once. (You can subsequently copy only specific data files.)

- Copy all volumes in independent mode

  At a given point-in-time, use a shell script to start an independent copy of all selected volumes. You can continue with normal operations when the system prompt returns.

- Copy all volumes in dependent mode

  At a given point-in-time, use a shell script to start a dependent copy of all selected volumes. You can continue with normal operations on the master and shadow volumes when the system prompt returns.

The access method providing the most practical solution is described in this document. The database files may be split into groups based on your requirements for availability and importance for the application recoverability process. See TABLE 4-1.

TABLE 4-1   Volume Access Method

Oracle File Entities	Access Method
System TableSpace and RollBack TableSpace	Raw volumes
Control files	Raw volumes or File system
Archive Logs	File system
Log Files	Raw volumes
Data Files	Raw volumes
Index Files	Raw volumes
Temp Files	Raw volumes
Application Binaries	File system

As determined by the customer's application requirements, you will need to regularly create coordinated point-in-time copy software snapshots of all volumes into the secondary host's storage. These volumes are recoverable using standard Oracle database procedures, will be used if the primary server fails, and must be rolled forward to the time of failure.

As these point-in-time snapshots will need to be rolled forward before the database is brought into production on the secondary server, it is important that these snapshots are taken frequently to ensure that update times are within the customer's recovery timeframes. Create scripts to update the database files and logs to the snapshot as often as possible.

Although the application binaries are not being replicated in this scenario, this replication is easy to implement if required.

Volume and Bitmap Requirements

Normally, the shadow volume must be the same size or larger than the master volume. However, if you use the shadow after disabling the shadow volume group (as part of a failover scenario) and the master needs to be re-synchronized with the shadow (also known as fallback), both volumes must be the same size.

A point-in-time copy software bitmap volume's size is based on the size of the master volume and the type of volume set being created (independent, full-sized dependent, or compact dependent) as follows:

- For independent or full-sized shadow dependent volume sets:

  8 Kbytes per 1 Gbyte of master volume size (rounded-up to the nearest whole Gigabyte), plus an additional 24 Kbytes for overhead.

  For example, to shadow a 3-Gbyte master volume, the bitmap size must be $(3 \times 8$ Kbytes$) + 24$ Kbytes, or 48 Kbytes in size.

- For compact dependent shadow volume sets:

  256 Kbytes per 1 Gbyte of master volume size (rounded up to the nearest whole Gbyte), plus 8 Kbytes per 1 Gbyte of master volume (rounded up to the nearest whole Gbyte), plus an additional 24 Kbytes for overhead.

  For example, to shadow a 3-Gbyte master volume, the bitmap size must be $(3 \times 256$ Kbytes$) + (3 \times 8$ Kbytes$) + 24$ Kbytes, or 816 Kbytes in size.

If you enable a volume set with a bitmap that is too large, the volume set is created even though space may be wasted. If you enable a volume set with a bitmap that is too small, the enable command fails with an error message.

You can have multiple pairs of master and shadow volumes; the number is limited only by the amount of storage you have physically available.

Using volume manager software like the Veritas Volume Manager, create disk groups. Also create volumes for:

- Oracle binaries
- Database data files
- Online logs
- Archive log files

Other Requirements and Suggestions

- For optimal performance, use the Sun StorEdge Availability Suite Software remote mirror software.
- Create directories under root as mount points for volumes.
- If UFS is used, make entries in the `/etc/vfstab` file. Do not mount the shadow volume.

Business Solutions Using the Point-in-Time Copy Software

This section describes business solutions using the point-in-time copy software and the remote mirror software:

General Operational Issues

Establish automated or manual operational procedures to use the point-in-time copy software. Examples of operations issues include deciding when and how often the snapshots will occur and how to notify the secondary application when the snapshot is available.

In most cases, before establishing a snapshot, your data must be consistent and usable. This consistent condition ensures that the snapshot image is logically consistent (that is, you have not captured any incomplete transactions).

Reducing Backup Windows

Backups are critical to information protection. In today's fast-paced business environments, backup windows are shrinking while the amount of information requiring protection is rapidly expanding. Backing up snapshot images, rather than online data, enables critical online transactions to keep running during the backup process. The general steps for performing online backups are as follows:

- Ensure that the data has reached a point of consistency (flush all data to the disk)
- Create a snapshot image of the data
- Use a backup utility to back up the snapshot image

Operationally, the snapshot copy backup procedures can be automated through various backup utilities. Using automated scripts to invoke the point-in-time copy application, these utilities are able to first take a snapshot copy of the online data and then back up the snapshot once it is available.

Analyzing Up-to-date Information

Businesses are constantly looking for ways to gain a competitive marketplace advantage. The faster they can convert data to information, the better they can make critical decisions. Snapshot copy technologies enable businesses to analyze more up-to-date information, leading to faster and better business decisions.

One way to frequently analyze up-to-date information is to take a snapshot of an online database and then use that snapshot as the data for a decision support application.

General procedures to create an independent database are as follows:

- Ensure that the primary database has reached a point of consistency (no "in-flight" transactions exist and all data is flushed to disk)
- Create a snapshot image of all database files
- Edit any database parameters that are database-instance-specific
- Start the secondary database instance on the snapshot image of the primary database

It is also possible to create a snapshot image without reaching a point of consistency, as follows:

- Create a snapshot image of all database files
- Edit any database parameters that are database-instance-specific (`init.ora` or control file if necessary)
- Start the secondary database instance on the snapshot image of the primary database
- Run recovery on the secondary database to rollback the "in-flight" transactions

New Application Development and Testing

Many enterprise systems are being re-engineered today to respond more efficiently to the networked business environment. Snapshot copy technologies enhance these efforts by enabling business applications to be tested on real, up-to-date data. This technology helps to accelerate new application development and helps to promote faster time-to-market.

Test applications are often not run on the same application server as the production application. Snapshot technologies can enable a snapshot of the production data to be made available to a test application running on a secondary server. The general procedure to accomplish this is as follows:

- Ensure the production data has reached a point of consistency (flush all the data to the disk)
- Create a snapshot image of the production data
- Make the snapshot image available to the secondary server (mount the volumes onto the secondary server)
- Run the test application against the snapshot image

Restoring from Snapshot Copies

Database or file system recovery time can be minimized using snapshot copy technologies by capturing frequent snapshots of online data which can then be used for restoration.

To restore master data from the most recent snapshot image, simply synchronize the master with the snapshot image. In most cases, only the changes from the last established snapshot will need to be copied to the master.

Restoring an older snapshot image can be performed in any one of the following ways:

- Perform a fast resynchronization from an older snapshot image
- Copy from tape to snapshot image volumes and then fully resynchronize the master to the snapshot image

Protecting Against Disasters

Remote mirroring technology, such as Sun StorEdge Availability Suite Software remote mirror software that replicates data between physically separated systems, is used to protect data if a disaster occurs. Refer to the *Usage Guide for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases* for details.

Snapshot copy and remote mirroring technologies can be coupled to provide advanced business continuance solutions. Combining these two technologies allows one or both of the following:

- A point-in-time copy snapshot of data to be remotely mirrored
- A point-in-time copy snapshot of a remote mirror to be established

Consider a business that requires data to be backed up at a secondary site. You can accomplish this task in at least two ways. The first way described minimizes both the overhead on the primary application and the intersite link traffic. The second way can be used for secondary site set up as a disaster recovery site.

Method 1

- Create a snapshot image of the primary data
- Forward the entire snapshot to the secondary site or just forward the changes from the last snapshot to the secondary site
- Back up the remotely mirrored data at the secondary site

Method 2

- Actively forward all primary write traffic to the remotely mirrored data at the secondary site
- At the secondary site, create a snapshot image of the remotely mirrored data
- Backup the snapshot image of the remotely mirrored data at the secondary site

Host-Based Versus Subsystem-Based Solutions

Host-based solutions:

- Provide solutions from low end to high end
- Are more easily integrated with the application
- Enable the snapshot image to span storage subsystems

Subsystem-based solutions:

- Offload the host from any processor cycles required to execute the snapshot copy solution

Suggested Methodologies Using the Remote Mirror Software

You can create the remote snapshot in a variety of ways. FIGURE 4-1 shows the different methodologies that can be applied based on available hardware and software.

Method 1 Using the Remote Mirror Software

- Replicate the production database to the remote host using the remote mirror software
- Take an independent or dependent snapshot of the mirrored image at the remote host
- Break the mirror and use the snapshot for performing backup procedures

See the *Usage Guide for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases* and the *Test Results for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases* for details.

Method 2 Using the Remote Mirror Software

- Take an independent snapshot of the complete database at the local host
- Use the remote mirror software to replicate the snapshot to the remote host
- Use the database at the remote host to launch a second instance or for performing backup procedures
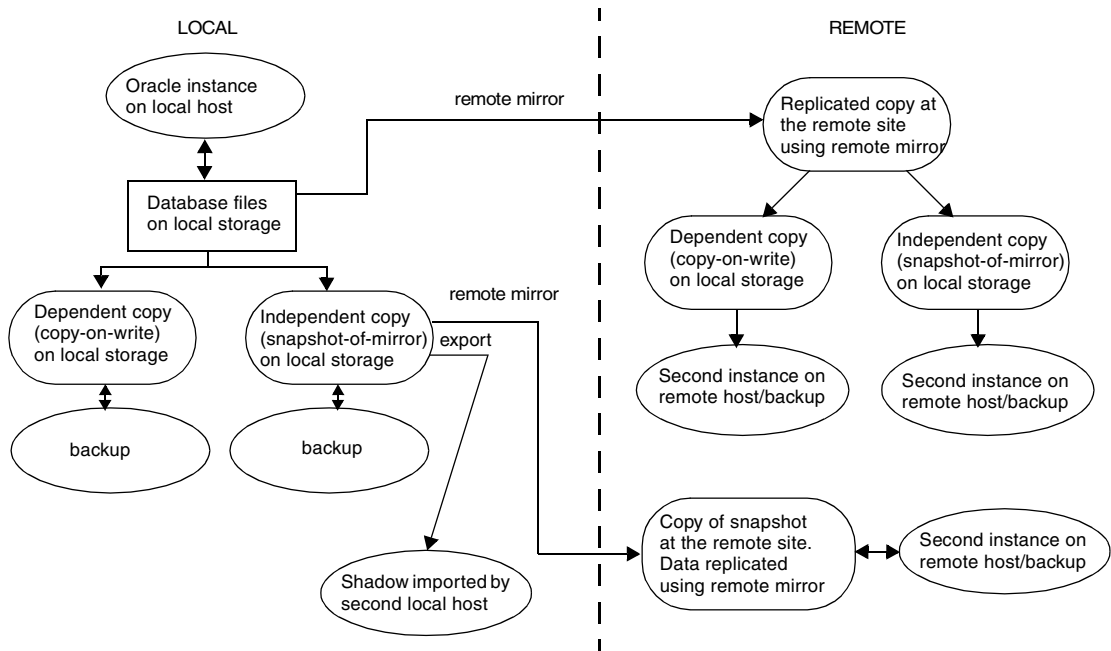
See the *Usage Guide for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases* and the *Test Results for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases* for details.

REMOTE

Oracle instance
on local host

remote mirror

Replicated copy at
the remote site
using remote mirror

Database files
on local storage

Dependent copy
(copy-on-write)
on local storage

Independent copy
(snapshot-of-mirror)
on local storage

Dependent copy
(copy-on-write)
on local storage

Independent copy
(snapshot-of-mirror)
on local storage

remote mirror

export

Second instance on
remote host/backup

Second instance on
remote host/backup

backup

backup

Shadow imported by
second local host

Copy of snapshot
at the remote site.
Data replicated
using remote mirror

Second instance on
remote host/backup

remote mirror = Sun StorEdge Availability Suite Software remote mirror software
export = shadow exported so second host can import it

FIGURE 4-1   Different Methods of Creating Remote Point-In-Time Copies

References

Related publications include:

- *Guidelines for Using Snapshot Storage Systems for Oracle Databases*
- *Sun StorEdge Availability Suite 3.2 Software Point-in-Time Copy Software Administrator and Operator Guide*
- *Usage Guide for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases*
- *Test Results for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases*

Glossary

bitmap volume The volume that tracks data changes on the shadow and master volumes.

cold-backup mode In cold-backup mode, you take the Oracle database that you are backing up offline at least until the backup is complete.

compact dependent shadow volume A shadow volume that is smaller than its associated master volume, but large enough to accept the expected number of writes.

dependent shadow volume set You can specify a shadow volume set as dependent when you enable it. The point-in-time copy software does not perform a full volume copy; data is not duplicated to the shadow volume until it is changed on the master volume. The dependent shadow volume is a copy that relies on the master for all unmodified data blocks.

disable This point-in-time copy software CLI command breaks the relationship between the volumes in a shadow volume set. Disabling an independent shadow volume set results in two usable copies of the data. Disabling a dependent shadow volume set causes the shadow volume to be marked invalid.

enable This point-in-time copy software CLI command creates the relationship between the volumes in a shadow volume set. You can enable either an independent shadow volume set or a dependent shadow volume set.

export The shadow volume of an independent shadow volume set can be exported if it resides on a dual-ported device. When a shadow volume is exported, it retains its relationship with its original shadow volume group, but can be imported by another host on the network.

fast resynchronization The point-in-time copy software copies all segments flagged as different between the master and shadow volumes, depending on which volume is being updated. It does not copy all volume data, only changed data. It is quicker than a full volume copy.

hot-backup mode	In hot-backup mode, you quiesce the Oracle database long enough to start the point-in-time copy software copy or update.
I/O volume group	An I/O volume group is a construct that includes more than one shadow volume set. You can run point-in-time copy software commands atomically against a group, thus affecting all shadow volume sets in the I/O volume group at the same time.
independent shadow volume set	You can specify a shadow volume set as independent when you enable it. The point-in-time copy software performs a full volume copy from master volume to shadow volume. The independent shadow volume is a copy that can stand alone even if the shadow volume set is disabled.
invalidate	After disabling a dependent shadow volume set, the point-in-time copy software clears the first 128 Kbytes of data in the volume to prevent users from accessing invalid or inconsistent data.
master volume	The volume containing the original data that is copied to the shadow volume.
multiple shadow volumes	You can enable any number of shadow volume sets with the same volume as the master volume, which gives that master volume multiple shadows.
overflow volume	You can attach any number of compact dependent shadow volumes to an overflow volume that will accept writes when the shadow volume is full rather than having the compact dependent shadow volume taken offline, which invalidates the shadow volume's data.
point-in-time copy snapshot	A copy or image of volume data captured or copied at and referenced to a particular point in time by the point-in-time copy software. Also called a snapshot.
resynchronization	See fast resynchronization.
shadow copy	The term, shadow copy, is a synonym for point-in-time copy or snapshot.
shadow volume	The volume containing a copy of data from the master volume.
shadow volume set	The master volume and its associated shadow and bitmap volumes.
volume pair	The master and shadow volumes in a shadow volume set.