Using Business Copy EVA with Oracle implementation guide

Part number: 5697-5456
First edition: August 2005

Legal and notice information

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The information is provided "as is" without warranty of any kind and is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Using Business Copy EVA with Oracle implementation guide

Contents

Tables

About this guide

This guide provides information about:

- The integration between Oracle and Business Copy EVA
- Procedures for using Business Copy with Oracle
- Using Business Copy to back up an Oracle database
- Restoring an Oracle tablespace that was backed up using Business Copy

Intended audience

This guide is intended for managers, supervisors, administrators, and operators of storage area networks (SANs) that include HP StorageWorks Enterprise Virtual Arrays. This document is intended for readers who are familiar with Oracle database administration.

Prerequisites

Readers should be familiar with the administration of SANs and local area networks. In addition, readers should be familiar with Oracle database administration.

Related documentation

In addition to this guide, please refer to other documents for this product:

- HP StorageWorks Business Copy EVA administrator guide
- HP StorageWorks EVA software compatibility reference
- HP StorageWorks Replication Solutions Manager installation and administrator guide
- HP StorageWorks Replication Solutions Manager V1.1 release notes

These and other HP documents can be found on the HP documents web site: http://www.hp.com/support/.

Document conventions and symbols

Table 1 Document conventions

Convention	Element
Medium blue text: Related documentation	Cross-reference links and e-mail addresses
Medium blue, underlined text (http://www.hp.com)	Web site addresses
Bold font	• Key names • Text typed into a GUI element, such as into a box • GUI elements that are clicked or selected, such as menu and list items, buttons, and check boxes
Italic font	Text emphasis
`Monospace font`	• File and directory names • System output • Code • Text typed at the command line
`Monospace, italic font`	• Code variables • Command-line variables
`Monospace, bold font`	Emphasis of file and directory names, system output, code, and text typed at the command line

△ CAUTION:

Indicates that failure to follow directions could result in damage to equipment or data.

📝 NOTE:

Provides additional information.

HP technical support

Telephone numbers for worldwide technical support are listed on the HP support web site: http://www.hp.com/support/.

Collect the following information before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

For continuous quality improvement, calls may be recorded or monitored.

HP strongly recommends that customers sign up online using the Subscriber's choice web site: http://www.hp.com/go/e-updates.

- Subscribing to this service provides you with e-mail updates on the latest product enhancements, newest versions of drivers, and firmware documentation updates, as well as instant access to numerous other product resources.
- After signing up, you can quickly locate your products by selecting **Business support** and then **Storage** under Product Category.

HP-authorized reseller

For the name of your nearest HP-authorized reseller:

- In the United States, call 1-800-282-6672.
- Elsewhere, visit the HP web site: http://www.hp.com. Then click **Contact HP** to find locations and telephone numbers.

HP storage web site

The HP storage web site has the latest information on this product. Access the storage web site from http://www.hp.com/country/us/eng/prodserv/storage.html. From this web site, select the appropriate product or solution.

Helpful web sites

For other product information, see the following HP web sites:

- http://www.hp.com
- http://www.hp.com/go/storage
- http://www.hp.com/support/
- http://www.docs.hp.com

1 Introduction

The purpose of this document is to define the scope of the integration between Oracle and Business Copy (BC) EVA. Procedures for using BC with Oracle, including sample SQL scripts, are provided. This document also provides sufficiently detailed descriptions of both the Oracle database architecture and BC to help you develop customized scripts that maximize the features of BC for use with your Oracle database.

This document is intended for readers who are familiar with Oracle database administration. Many of the commands and functions described within require DBA privileges and can be disruptive to database operation if used incorrectly. If you are not completely comfortable using the commands and functions described herein, consult your Oracle documentation before proceeding.

2 Support information

Oracle version support

This document describes methods that are supported with the following Oracle versions:

- Oracle 8i
- Oracle 9i
- Oracle 10g

Business Copy version support

The methods described in this document are supported by Business Copy v2.3 and Replication Solutions Manager (RSM) v1.1 or later.

Operating system version support

The solution described in this document is supported on any operating system that is supported by Oracle and the BC or RSM Host Agent. Refer to the *HP StorageWorks EVA software compatibility reference* for details on Host Agent support.

3 Solution Overview

Business Copy is a web-based software tool that provides a method for creating instant point-in-time copies of EVA-based storage volumes. These copies, called Business Continuance Volumes or BCVs, can be migrated to and mounted on any other supported host in the BC storage network. One advantage of this feature to the Oracle DBA is the ability to perform backup operations on a copy of the production data, and to do so on a completely different host. This off-loads the sometimes significant CPU overhead associated with the backup process from the database server to a dedicated backup host.

Online backup

Oracle's online or "hot" backup utility allows tablespaces to be backed up while the database is open. Although the online backup feature eliminates the need to take the database offline before performing a backup, there is a performance penalty associated with its use. The performance penalty occurs because it is necessary to leave each tablespace in hot backup mode for as long as the backup requires to complete, possibly several hours or more. With BC, it is only necessary to leave the tablespaces in hot backup mode for a few seconds while the BCVs are split from the production volumes. In short, BC greatly reduces the performance penalty associated with online backups and expands the backup window to 24 hours a day, 365 days a year.

In addition to facilitating backup, BC also provides the ability to make instant, independent copies of one or more database volumes. You can mount these copies on a separate host for batch processing, data mining, application testing, data warehousing, or other processes that could benefit from having an independent copy of the database.

In order for a snapshot- or snapclone-based backup to be valid, it is necessary to ensure that the BCVs contain an application-consistent copy of the database. This involves ensuring that there are no dirty buffers and that the data remains valid during the brief period while the source volume and the BCV are being separated from each other. BC accomplishes this through the use of the **Suspend** and **Resume** job commands.

These commands are used in conjunction with Oracle's online backup feature to ensure that the BCVs contain a consistent copy of the database files.

The **Suspend** and **Resume** commands both accept a standard path/filename as input and are used to execute batch files, scripts, or compiled executables on the specified server at certain points during the execution of the BC job. While creating these files is left up to the user, they are intended to be used to execute a short sequence of specific SQL commands, which will be described later. The examples given below use the Oracle command line mode to execute short SQL scripts. There are many other ways of accomplishing this same goal, and so long as the end result is the same, the details of the implementation are not important.

BC will execute the file contained in the PRE field just before the BCV is split from the source volume. This file should checkpoint the database and place the necessary tablespaces in online backup mode. The POST file is executed just after the BCV is split and should be used to take each tablespace out of online backup mode. If used correctly, the PRE and POST files will ensure that the BCV contains a valid and consistent copy of the database.

More about online backup

Oracle has a built-in feature that allows the execution of online or "hot" backups. This function allows backup operations to take place while the database is running and users are continuing to access the database. This eliminates the need to shut down the database each time a backup is needed.

Oracle's implementation of this feature is via the `backup` parameter of the **alter tablespace** command.

For example:

```
ALTER TABLESPACE tablespace_name BEGIN BACKUP;
```

would place *tablespace_name* in online backup mode.

Similarly:

```
ALTER TABLESPACE tablespace_name END BACKUP;
```

would take *tablespace_name* out of online backup mode.

Note that the unit of backup for this function is the tablespace. In order to back up an entire database, it will be necessary to issue the above command for each tablespace. Although online backups deal with individual tablespaces, it is possible to back up several tablespaces at once by issuing multiple online backup commands. Normally, this would not be recommended due to the undesirable side effects of having multiple tablespaces in online backup mode simultaneously. Fortunately, BC eliminates these problems by reducing the amount of time that the tablespaces need to be in hot backup to only a few seconds.

In order to use the online backup feature, the database must be operated in ARCHIVELOG mode. Otherwise, online backups will not work.

To enable ARCHIVELOG mode, place the database in the mounted (but not open) state and issue the following commands from the Server Manager prompt.

```
ALTER DATABASE ARCHIVELOG;
ARCHIVE LOG START
ALTER DATABASE OPEN;
```

To make these changes permanent, you must modify your `init.ora` file.

4 Creating a backup plan

A comprehensive Oracle backup plan should include the data files, the archived redo logs, and the control file. It is assumed that the Oracle system files are included as part of the OS backup.

Data files

Data files are the physical files that contain a tablespace's data and are the most critical components of any backup. By default, Oracle creates the SYSTEM and USER tablespaces and their associated data files. In addition to these default data files, each additional tablespace will contain one or more data files. As a database grows, new data files may be added, thereby complicating the backup plan. In order to develop an effective backup plan, it is important to understand where each data file is located and what tablespace it belongs to.

Archived redo logs

As mentioned earlier, the online backup function requires the database to be run in ARCHIVELOG mode. In this mode, as each online redo log fills up, Oracle copies it to the location specified by the *`ARCHIVE_LOG_DEST`* parameter and begins filling the next online redo log. In this manner, the online redo logs are recycled and a sequential trail of redo is maintained so that any changes that are lost can be rolled forward easily.

Snapshot– or snapclone–based backups

Snapshot- or snapclone-based backups are point-in-time images of the database as it exists when the backup is created. By themselves, these backups can be used to restore the database back to the point in time the backup was created. In most cases though, it is desirable to roll the database forward from the point of the backup to a more current time. The ability to roll forward from a snapshot or snapclone backup depends on the availability of a continuous set of archived redo logs for the tablespaces in question.

Backing up archived redo logs

There are several methods of backing up archived redo logs. The method you choose will depend primarily on where the logs are located. If they reside on an HP storage array, you may be able to create a separate snapshot or snapclone job and back the archived redo logs up in the same manner as the data files. Regardless of the method used, the primary point to remember is that a restored database can only be rolled forward to the point in time of the most recent archived redo log backup. This determines the "data exposure" or the maximum amount of data that may be lost in the event of a failure. This is a critical aspect of the backup and restore process and should be given careful consideration when designing the backup plan.

Additional redo information is generated constantly and it is necessary to back up the redo logs periodically. Remember that complete database recovery requires the most recent backup of the data files, plus all of the archived redo logs that were generated since the data files were backed up. Any gap in the archived redo log sequence will prevent making a complete recovery. Of course, it is possible to roll the transactions forward to the point of the first missing log.

Once archived redo logs have been backed up, they can be deleted to free up space. However, if you have enough free space, keeping the archived redo logs on disk between data file backups can significantly speed the recovery process if only data files are lost.

Control file

The control file contains the schema of the database, and, among other things, includes the data necessary to restore the database from an online backup. However, this data is not written to the control file until after all tablespaces have been taken out of online backup mode. Therefore, the control file cannot be backed up until after online backup mode has ended. By that time, each BCV has already been split from its source volume, so there is no simple way to get the current control file onto the BCV. Remember that any copies of the control file that may be located on the BCVs are old copies, in the sense that they do not contain the data needed to restore the database from the online backup. If possible, it is best not to back these files up at all, since restoring them will overwrite the current control file, the one containing the information needed to perform the recovery.

Considering the problems that can result from improperly restoring a control file, many Oracle administrators prefer not to back it up at all. By default, the control file is mirrored to at least two locations, so the loss of one copy will not seriously impact database operation. Whether or not to back up the control file is in important decision and should be made carefully after fully considering the implications. Consult the Oracle documentation for more information on this subject.

Online redo logs

Online redo log files should never be backed up. While actually backing them up is not a problem, restoring them will overwrite the current online redo logs and will make it impossible to completely recover the database. Considering the dangers involved, it is best never to back them up in the first place. As with the control file, it is possible to mirror the online redo logs to multiple locations so that the loss of one set will not adversely affect the operation of the database.

5 Using Business Copy to back up an Oracle database

The following is the general sequence of steps to follow for implementing snapshot- or snapclone-based backups of an Oracle database using Business Copy.

1. Determine which tablespaces need to be backed up.
2. Create the batch and script files to suspend and resume the desired tablespaces.
3. Determine which volumes need to be replicated.
4. Construct the BC or RSM job to replicate the volumes identified in step 3.
5. Test the process.

Determining which tablespaces need to be backed up

There are a few things to keep in mind when choosing the tablespaces to be backed up.

- Any tablespace can be placed in online backup mode, but only those tablespaces whose data files reside entirely on storage that can be replicated by BC can be backed up using snapshot or snapclone methods.
- Since a BC-created snapshot or snapclone is a point-in-time copy of an entire physical disk, it is important to realize that the snapshot copies may contain data from other tablespaces that may not have been placed into online backup mode before the snapshot was created. Only those datafiles belonging to tablespaces that were placed in online backup mode will be useful for recovery purposes.

Creating the batch and script files

Once you know which tablespaces will be replicated, you can create the batch files that will be used to take the tablespaces into and out of online backup mode. Two batch files will be needed: a `pre.bat` to place the tablespaces into online backup mode, and a `post.bat` to take them back out. The actual names of the batch files are not important, but we will use `pre.bat` and `post.bat` for all of the examples in this guide.

The following examples show the two different methods that can be used. These are provided as examples only. It is the user's responsibility to verify the effectiveness and suitability of these batch files for their intended application.

Sample PRE routine

The suggested method of issuing the following SQL scripts is to use Oracle's SQLPLUS utility in command line mode. The command to execute the SQL script `BC-pre.sql` on Windows might look something like this:

```
c:\orant\bin\sqlplus "@BC-pre" > pre.log
```

This command specifies the path to the SQLPLUS utility, passes the `BC-pre.sql` file as an input, and redirects the output to the `pre.log` file. Redirecting the output to a log file is a very important step that will be explained shortly. This command should be placed in the `pre.bat` file.

The actual SQL commands that are issued to Oracle are contained in the `BC-pre.sql` file. The following samples can be used as a starting point for developing your own.

As an example, let's say you wanted to back up tablespaces USR and WEB. A suitable command file would look something like this:

```
****
connect internal/letmein;
archive log list;
alter tablespace USR begin backup;
alter tablespace WEB begin backup;
quit;
****
```

Let's examine this file line-by-line.

1 `connect internal/letmein;`

The first line connects to the database with the specified username and password. In this case the user is 'internal' and the password is 'letmein'.

2 `archive log list;`

The next line issues the **archive log** command with the *`list`* parameter. The output of this command will look something like this:

```
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            H:\archive
Oldest online log sequence     266
Next log sequence to archive   267
Current log sequence           267
```

The important piece of information here is the `Oldest online log sequence`, which in our case is `266`. The need to identify the oldest online log sequence is why we redirected the output from the SQLPLUS command to a log file. This information will be needed later in order to determine which archived redo logs need to be backed up as part of the current online backup.

3–4

```
alter tablespace USR begin backup;
alter tablespace WEB begin backup;
```

The third and fourth lines of the script are the commands that actually place the tablespaces USR and WEB into online backup mode.

5 `quit;`

The last line exits the script and returns control to the BC job.

When the PRE routine completes, the specified tablespaces will have been placed into online backup mode, thereby ensuring that the data files are consistent. BC can now create snapshot or snapclone copies. This takes a few seconds, and once done, the next step is to take the tablespaces out of online backup mode through the use of the POST routine.

Sample POST routine

As with the PRE routine, we need to issue a series of SQL commands to Oracle. We can accomplish this by using a `post.bat` file to again call the SQLPLUS utility as follows:

```
c:\orant\bin\sqlplus "@BC-post" > post.log
```

This time we will pass a different set of commands, which are located in the `BC-post.sql` command file and will redirect the o

```
connect internal/letmein;
alter tablespace USR end backup;
alter tablespace WEB end backup;
archive log list;
alter system switch logfile;
quit;
****
```

Let’s examine this file line-by-line.

1

```
connect internal/letmein;
```

Here again, the first line connects to the database using the specified user ID and password.

2–3

```
alter tablespace USR end backup;
alter tablespace WEB end backup;
```

The second and third lines take the tablespaces USR and WEB out of online backup mode.

4

```
archive log list;
```

The fourth line is the same as it was in the PRE routine except now the output will look something like this:

```
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            H:\archive
Oldest online log sequence     312
Next log sequence to archive   313
Current log sequence           313
```

The important piece of information here is the `Current log sequence number`, `313` in our case. Once again, the need to know the current log sequence number is the reason the output from SQLPLUS was redirected to a log file.

5

```
alter system switch logfile;
```

The next line of the script forces a logfile switch. This causes Oracle to stop recording redo in the current log sequence number and forces a switch to the next sequential log file: 314, in this case. The log file that was current (313) is then archived. The Oldest online log sequence identified in the PRE routine and the Current log sequence from above now specify the entire range of archived redo logs that must be backed up in order to ensure a complete recovery. Since the snapshots and snapclones that have just been created will not contain the most recent log, they must be backed up separately, as must any additional logs that are generated. This can be done using another BC job, or outside of BC altogether.

6

```
quit;
```

As before, the last line exits SQLPLUS and returns control to the BC job.

When the POST routine completes, the specified tablespaces will have been taken out of online backup mode. BC can now present and mount the snapshot and snapclone copies on the specified host.

Determining which volumes need to be replicated

Most Oracle databases are comprised of multiple storage volumes. Since the unit of online backup for Oracle is the tablespace, any BC jobs you create must take into account that a single tablespace may span multiple volumes. If this is the case, it is necessary to create a BCV of every volume that contains a portion of the tablespace. Let’s say your Oracle implementation involves four volumes (E:, F:, G:, and H:) and a total of four tablespaces (SYSTEM, USER, TEMP, and WEB). Assume the data files for these tablespaces are located as follows:

Volume E:

`SYSTEMorc1.ora`	Data file #1 for SYSTEM
`USERorc1.ora`	Data file #1 for USER
`TEMPorc1.ora`	Data file #1 for TEMP
`WEBorc1.ora`	Data file #1 for WEB

Volume F:

`SYSTEMorc2.ora`	Data file #2 for SYSTEM
`USERorc2.ora`	Data file #2 for USER

`WEBorc2.ora`	Data file #2 for WEB
`WEBorc3.ora`	Data file #3 for WEB

Volume G:

`WEBorc4.ora`	Data file #4 for WEB
`WEBorc5.ora`	Data file #5 for WEB

Volume H:

Archived redo logs only

In order to capture all of the data files for the WEB tablespace, a BC job would need to create BCVs for E:, F:, and G:. This is because the WEB tablespace has data files located on all three volumes. The SYSTEM and USER tablespaces would need BCVs of both E: and F:, and TEMP would need a BCV of E: only.

Constructing the BC or RSM job

Once you have created the `pre.bat` and `post.bat` files and determined the list of volumes that need to be replicated, you have enough information to construct the BC or RSM jobs. The following BC and RSM sample jobs are used to replicate the WEB tablespace described in the previous step. These samples use the Fully Allocated snapshot method. Depending on the storage array that is being used, other replication methods may also be available.

NOTE:

These samples are included for informational purposes only and are not intended to be used as is.

Sample BC job

This sample job was created using the `snap two volumes.suspend.mount` template in BC v2.3. One additional **SNAP VOLUME** step and one additional **MOUNT** step were added to accommodate the three volumes that need to be replicated in this job. The comments that are normally included with this template have been removed for clarity.

```
;Sample BC job for WEB tablespace
SUSPEND WAIT TRUE TRUE INTEGER 0 Server1 pre.bat
SNAP VOLUME Server1 E: $BCV1 FULLY_ALLOCATED SAME_AS_SOURCE
SNAP VOLUME Server1 F: $BCV2 FULLY_ALLOCATED SAME_AS_SOURCE
SNAP VOLUME Server1 G: $BCV3 FULLY_ALLOCATED SAME_AS_SOURCE
RESUME WAIT TRUE TRUE INTEGER 0 Server1 post.bat
MOUNT VOLUME_SINGLE SS $BCV1 Backup-Server N/A N/A 1 L:
MOUNT VOLUME_SINGLE SS $BCV2 Backup-Server N/A N/A 1 M:
MOUNT VOLUME_SINGLE SS $BCV3 Backup-Server N/A N/A 1 N:
LAUNCH WAIT TRUE TRUE INTEGER 0 Backup-Server backup.bat
```

Sample RSM job

This sample job was created using the `Replicate host volumes, mount to a host` template. The *# of volumes to replicate* was set to 3 and the **Suspend source before replication** and **Launch backup after replication** check boxes were selected. The comments and validation steps that are normally included with this template have been removed for clarity.

```
// Sample RSM job for WEB tablespace
Launch ( "Server1", "pre.bat", "", WAIT, "0" ) onerror pauseat E1:
DO {
$Rep1 = SnapshotHostVolume ( "\\Server1\E:\", FULLY_ALLOCATED, SAME, NOWAIT ) onerror pauseat E1:
$Rep2 = SnapshotHostVolume ( "\\Server1\F:\", FULLY_ALLOCATED, SAME, NOWAIT ) onerror pauseat E2:
```

```
$Rep3 = SnapshotHostVolume ( "\\Server1\G:\", FULLY_ALLOCATED, SAME, NOWAIT ) onerror pauseat E
} ALWAYS {
Launch ( "Server1", "post.bat", "", WAIT, "0" )
}
WaitForStorageVolumesDiscovery ( $Rep1 ) onerror pauseat E4:
WaitForStorageVolumesDiscovery ( $Rep2 ) onerror pauseat E4:
WaitForStorageVolumesDiscovery ( $Rep3 ) onerror pauseat E4:
$HV1 = CreateHostVolume ( "\\Server1\E:\", $Rep1, "Backup-Server" ) onerror pauseat E6:
$HV2 = CreateHostVolume ( "\\Server1\F:\", $Rep2, "Backup-Server" ) onerror pauseat E7:
$HV3 = CreateHostVolume ( "\\Server1\G:\", $Rep3, "Backup-Server" ) onerror pauseat E8:
$MP1 = MountHostVolume ( $HV1, "L:" ) onerror pauseat E10:
$MP2 = MountHostVolume ( $HV2, "M:" ) onerror pauseat E11:
$MP3 = MountHostVolume ( $HV3, "N:" ) onerror pauseat E12:
Launch ( "Backup-Server", "backup.bat", "", WAIT, "0" ) onerror pauseat E12:
Pause (  )
//Unwind section
E12: UnmountHostVolume ( $MP3 ) onerror pauseat E11:
E11: UnmountHostVolume ( $MP2 ) onerror pauseat E10:
E10: UnmountHostVolume ( $MP1 ) onerror pauseat E9:
E8: DeleteHostVolume ( $HV3 ) onerror pauseat E8:
E7: DeleteHostVolume ( $HV2 ) onerror pauseat E7:
E6: DeleteHostVolume ( $HV1 ) onerror pauseat E6:
E4: DeleteStorageVolumes ( $Rep3 ) onerror pauseat E4:
E3: DeleteStorageVolumes ( $Rep2 ) onerror pauseat E3:
E2: DeleteStorageVolumes ( $Rep1 ) onerror pauseat E2:
Exit ( SUCCESS )
E1: Exit ( FAILURE )
```

Test the process

If used correctly, BC can significantly increase the efficiency of your Oracle backup, restore, and migration operations. As with any change to your database configuration or backup plan, it is vitally important that you understand all aspects of your backup and restore operations and test them thoroughly before implementing them in a production environment.

6 Restoring an Oracle tablespace that was backed up using BC

NOTE:

The following information is intended to illustrate a possible sequence of steps to restore an Oracle database from an online backup. This information is not intended to represent a step-by-step recovery plan. There are many variables that affect how a database should be recovered. It is the Oracle DBA's responsibility to ensure the suitability of these methods before applying them.

Restoring a database that has been backed up using the BC methods typically involves the following basic steps.

1. Identifying and preparing resources
2. Preparing the database for recovery
3. Replacing the needed files
4. Recovering the database and applying redo logs

Identifying and preparing resources

Before recovery can begin, you must know where you will be recovering from.

- Is the data on BCV or on tape?
- If it's on tape, what is the name of the backup application's restore job that must be run?
- Have the appropriate steps been taken to ensure that the restored data will be written to the correct location?
- Are the necessary tapes on hand and loaded?
- If the data to be restored is on BCV, where is the BCV currently located?
- Will a reboot be required in order to make the BCV visible to the database server?

Preparing the database for recovery

Shut down the database in NORMAL, IMMEDIATE, or ABORT mode.

NOTE:

On Windows you cannot use ABORT alone because it does not clear the operating system file locks, and you cannot overwrite the datafiles with your backup copies. If ABORT must be used, shut down and restart the operating system, or use a file system utility to clear the file locks.

Replacing the needed files

Several approaches are possible, depending on the situation and the type of backups that are available. Be sure to choose an appropriate recovery model. Choosing an incorrect model may prevent archived redo logs from being applied for recovery and could result in being unable to restore to the latest log backup.

In general, there are two basic recovery methods that can be used with snapshot- or snapclone-based backups: volume replacement and selective file replacement, as described in the following two sections.

Volume replacement

Volume replacement involves completely removing the existing source volume and replacing it with a backup copy. This method is preferred when a physical failure of the source volume has occurred, or if the combined size of the files that need to be restored is large enough that a selective file replacement (described below) would take too long to complete.

There are two ways that volume replacement can be accomplished, as described in the following two sections.

Via direct snapclone

With this approach, the source volume is removed from the source host and replaced with a snapclone copy, using the same drive letter or mount point.

From tape via intermediate device

If no snapclone copy is available but the required fileset is available on tape, it is possible to:

1. create a new volume of the desired size and redundancy level,
2. present it to the backup server,
3. restore the data from tape to the new volume, and
4. use this new volume as a replacement for the original source.

△ CAUTION:

The volume replacement method replaces the entire contents of the source. If the source volume contains data or log files for tablespaces that were not placed into online backup mode when the copy was created, then replacing the entire volume with a snapclone copy will corrupt these tablespaces. When using the volume replacement method, be sure that *all* of the files on the replacement volume belong to databases that were backed up at the same time.

NOTE:

Backups based on snapshots (as opposed to snapclones) are not suitable for volume replacement, since snapshots are virtual volumes that depend on the integrity of the source. If the source volume has physically failed, any snapshots of it that exist will have likely failed as well.

Selective file replacement

This method involves leaving the original source volume in place and replacing some or all of the files that are contained on it with backup copies. This method is generally preferred over volume replacement in cases where only a subset of the tablespaces that reside on the volume need to be restored.

There are two ways that selective file replacement can be accomplished, as described in the following two sections.

Drag and drop from snap

If a snapshot or snapclone copy of the required data and log files is available, then:

1. the snapshot or snapclone devices can be presented to the source host,
2. the devices can be manually mounted using the OS utilities, and
3. the required files can simply be dragged and dropped from the snapshot or snapclone back to the original source volumes.

From tape

If the backup exists only on tape, it may be possible to restore the required files from tape directly back to the source volume. This process will be slightly complicated by the fact that the files being restored were originally backed up from the snapshot or snapclone copies, probably on a different server. To restore back to the source volume, the restore process will need to be redirected to the desired location. Most backup applications support this feature.

Recovering the database and applying redo logs

This step in the recovery process depends on the nature of the failure that is being recovered from, and whether the database needs to be online during recovery. Oracle provides a variety of methods for performing data file recovery, tablespace recovery, and database recovery. Consult the Oracle documentation for more information on these functions.