

Using Recovery Manager with Oracle Data Guard in Oracle9i

An Oracle White Paper
March 2004

Using Recovery Manager with Oracle Data Guard in Oracle9i

Executive summary	3
Introduction	3
Configuration Settings and Considerations	5
Recommended RMAN Configuration	5
Recommended Oracle Database Configuration.....	7
Archive Log Backup Considerations	7
Backup Procedure	8
Case 1: Backup to Disk, Copy to Tape.....	8
Primary Database Backup Procedure.....	8
Standby Database Backup Procedure.....	9
Case 2: Backups Directly Written to Tape	10
Primary Database Backup Procedure.....	10
Standby Database Backup Procedure.....	10
Recovery Procedure	11
Recovery from Loss of Data Files on Primary Database	11
Recovery from Loss of Data Files on Standby Database	12
Recovery from Loss of Control File on Standby Database.....	12
Recovery from Loss of Control File on Primary Database.....	13
1. Failover to Standby Database	14
2. Create a New Control File.....	14
3. Recover Using Backup Control File	14
Recovery from Loss of an Online Log.....	15
Incomplete Recovery of the Database.....	16
Additional RMAN Backup Considerations	17
Avoiding Standby Re-instantiation after RESETLOGS Operation on the Primary Database	17
Standby Database Instantiation Using RMAN	18
Variant Standby Database Instantiation using RMAN	19
Conclusion.....	20
Appendix	21
Inability to Utilize Backups Taken at the Originating Host	21
Standby Database Configured as Archive Log Repository.....	21
Standby Database Filenames Differ from Primary Database	22

Using Recovery Manager with Oracle Data Guard in Oracle9i

EXECUTIVE SUMMARY

A well-documented and validated system and software recovery plan is critical to an overall high availability strategy. Oracle DBAs rely on Oracle Data Guard to provide continuous uptime in the event of storage subsystem failure, site-wide failure or disaster. Data Guard is the management, monitoring, and automation software infrastructure that creates, maintains, and monitors one or more standby databases to protect enterprise data from failures, disasters, errors, and corruptions.

Similarly, DBAs depend on Oracle Recovery Manager (RMAN) to quickly and efficiently backup control files, data files, archive logs to disk and tape while speedily recovering these files upon a filesystem or media loss. RMAN is an easy-to-use tool that can take backups with minimal impact on production databases and quickly recover from the loss of individual data files, or the entire database.

With the Oracle9i Database, the benefit of these two complementary technologies can be fully realized; DBAs can utilize RMAN to quickly setup and backup standby databases managed by Data Guard. This paper offers a best practice backup and recovery strategy using RMAN in a Data Guard environment. This includes:

- Simplifying the setup of a Data Guard configuration
- Creating database backups at the standby database that can be used to recover the primary or standby database
- Recovering data files on the primary or standby database that were made on the standby database

This paper is for DBAs, IT, and system administrators, who have basic experience with RMAN commands, and are interested in the RMAN procedures and syntax to manage backups in their Data Guard environment. The paper also assumes familiarity with Data Guard concepts and procedures.

INTRODUCTION

Data Guard enables and automates the management of a disaster recovery solution for Oracle databases located on the same campus or across the continent. Data Guard consists of a *production database* (also known as the *primary database*) and one or more *standby database(s)*, which are transactionally consistent copies of the production database. As transactions occur in the primary database, redo data is generated and is written to the local redo logs. Data Guard automatically transfers

this redo data to the standby sites and applies it to the standby databases, synchronizing them with the primary database.

RMAN is an integrated tool with the Oracle Database which satisfies the demands of performant, manageable backup and recovery, for *all* Oracle data formats. Because RMAN is native to the server, it keeps up to date with any database structure changes, and optimizes operations accordingly. In addition, RMAN out-of-the-box can backup to leading tape and storage media vendors' products via the supplied Media Management Library (MML) API.

The RMAN Catalog organizes backup histories and other recovery-related metadata in a centralized location. The recovery catalog is configured in a database and can hold information about backups from many databases. A catalog server, physically separate from the primary and standby sites, is recommended in a Data Guard configuration as disaster striking either site will not affect the ability to recover the latest backups.

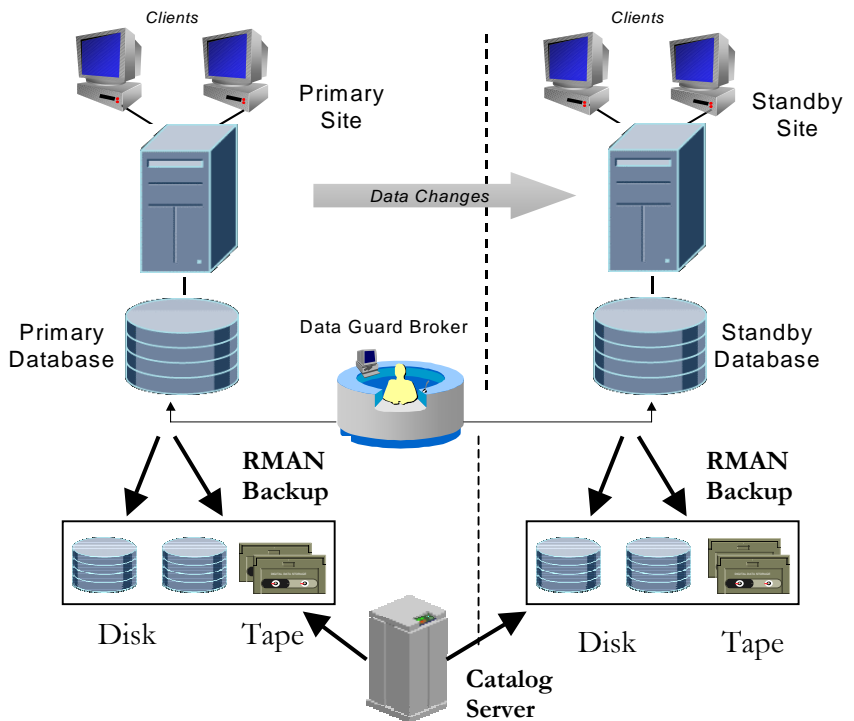


Fig. 1: Data Guard & RMAN Architecture

Data Guard and RMAN were both designed with the Oracle database architecture in mind. Together, they offer the most reliable and tightly integrated solution to achieve superior levels of Oracle database availability supporting your mission critical applications. Both technologies are available at no additional cost to the Oracle Database Enterprise Edition, and are fully supported by Oracle.

RMAN brings rich functionality such as online backups, incremental backups, block media recovery, automation of backup management tasks, and integration with 3rd party tape management systems into the Data Guard environment. In addition, backups can be offloaded to the standby database, making more efficient use of resources on the primary system; backups from either database can be used for recovery.

This paper discusses:

- Initial RMAN and Data Guard configuration settings
- Backup procedures for primary and standby, to disk and tape
- Recovery scenarios on primary and standby
- RMAN-based instantiation of standby database

The assumptions for this setup are:

- Standby database is a physical standby database.
- The directory structures on the primary and standby database are identical. This simplifies the RMAN backup and recovery operations no matter which host is used.
- All databases in the configuration use Oracle9i Database Release 2.
- RMAN recovery catalog is mandatory.
- A 3rd party media management software is configured and integrated with RMAN to make backups to tape.
- The RMAN backup retention policy is set to 7 days.
- The databases do not use Oracle Managed Files (OMF).

Note: The [Appendix](#) describes modifications to these procedures for three variant configurations:

- Backups are made at both the primary and standby database sites due to the inability to access the backup from the originating host.
- Standby database is configured as an archive log repository
- Standby database file names are different than those on the primary

CONFIGURATION SETTINGS AND CONSIDERATIONS

The following recommended settings for RMAN and the Oracle database will simplify the backup and recovery operations, and are based on datafile backups being made at the standby database.

Recommended RMAN Configuration

RMAN introduced the capability to store persistent configuration data in Oracle9i. This allows a DBA to setup the RMAN backup and restore behavior one time and be consistent no matter who is responsible for the backup and recovery of the

Oracle database. Best practices recommend that the same RMAN settings and configurations be used by the primary and standby databases.

These persistent configurations are used for various operations; for example, channel and device settings are used for allocating channels, and retention policy setting is used for reporting/deleting obsolete backups. If either the standby database control file, prior to changing role to PRIMARY, or the newly created control file on primary database is missing RMAN persistent configuration, they must be modified as explained in this section.

The following configuration settings are required:

- Same RMAN PARMS string on primary and standby systems so that the Media Management software is configured identically on both systems. The tape devices (RMAN SBT channels) are configured such that the database backups can be read from and written to the same directory or tape device from the primary and standby database systems
- Use a RMAN recovery catalog for the following reasons:
 - Backups taken on one node can be restored on another node
 - A recovery catalog database does not have the space limitations of the control file and can store more historical data about backups.
 - Improved performance during restore and maintenance operations

The following RMAN CONFIGURE commands should be issued at the RMAN prompt, after connecting to the primary and recovery catalog databases. The RMAN SHOW ALL command can be used to save the commands that will re-create the RMAN persistent configuration.

The following RMAN persistent configurations are suggested:

- CONFIGURE CONTROLFILE AUTOBACKUP ON
to enable automatic backups of the control file and SPFILE.
- CONFIGURE CONTROLFILE AUTOBACKUP FORMAT '<backup directory>/%F'
to specify the location for control file and SPFILE auto backups.
- CONFIGURE BACKUP OPTIMIZATION ON
to prevent backing up of database files for which there already exists a backup.
- CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <n> DAYS
to ensure that RMAN's DELETE OBSOLETE command will not delete backups required for point-in-time recovery of the database within <n> days recovery window. The number of days in the recovery window is a function of your business requirements. In the backup procedures for this paper, n=7 days.
- CONFIGURE CHANNEL DEVICE TYPE SBT PARMS '<channel parameters>'
to set up the channel parameters that are required by the Media Management software. You should verify that RMAN can allocate

SBT channels using the same configuration on both the primary and standby instances. If that is not possible, then use manually allocated channels in the RMAN scripts that run on the standby database.

- CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '<backup directory>/%U'
to set the location for disk-based backups. You should verify that both the primary and standby system can read and write files in this location.

The RMAN procedures outlined in this paper require connection to either the primary or standby database (as the target database) and connection to the recovery catalog. This can be done when starting RMAN from the command prompt:

```
> RMAN TARGET SYS/<TARGET SYS PASSWORD>@<TARGET NET SERVICE NAME>  
CATALOG <CATALOG USERNAME>/<CATALOG PWD>@<CATALOG NET SERVICE  
NAME>
```

Recommended Oracle Database Configuration

- Multiplex the control files for the primary and standby database instances, and ensure that the control files are on separate disks.
- Multiplex the online logs for the primary database instance, and ensure that the members of same group are on separate disks.
- Set the STANDBY_FILE_MANAGEMENT initialization parameter to AUTO, so that when data files are added to or dropped from the primary database, corresponding changes are made in the standby database without manual intervention.
- Use a system parameter file (SPFILE) for both the primary and standby, and configure the SPFILE so that it contains all parameters needed to start either the primary or the standby database. The SPFILE must be backed up on the primary database, and can be restored using the RMAN RESTORE SPFILE command, in the event of loss of all system files.

Archive Log Backup Considerations

There are special cases where archive logs generated after an archive log backup must be manually cataloged using the following RMAN command:

```
CATALOG ARCHIVELOG '<archive log file name>';
```

This is required so that the archive logs are known to RMAN for future backups. The cases when the CATALOG command must be issued are:

- The primary or standby control file is re-created. RMAN uses the controlfile to determine which archive logs must be backed up.
- The primary database role changes to standby after switchover or failover operation. The archive logs must be re-cataloged because a change in database role resets the version time of the mounted control file.

Only those archive logs received by the standby instance can be backed up at standby site. Archive logs that were created before the standby was instantiated must be backed up on the primary database.

Oracle recommends that only RMAN be used to delete archive logs. By using the BACKED UP option in the DELETE command, RMAN can verify that a backup exists before deleting the file.

BACKUP PROCEDURE

In a Data Guard environment, backups of data files and archive logs taken on the primary or standby system are usable on either system for recovery. The process of backing up data files and archive logs can be offloaded to the standby system to minimize the impact of backup operations on the production system. These backups can be used to recover the primary or standby database. This section describe the RMAN scripts and procedures to backup the Oracle database in a Data Guard configuration.

Case 1: Backup to Disk, Copy to Tape

The following RMAN scripts perform:

- Weekly full database backups
- Daily incremental backups
- Daily archive logs backups to at least two different backup sets.
- Backup to disk, then copy to tape as soon as possible.

Primary Database Backup Procedure

Although backups are offloaded to the standby database, the control file and SPFILE must continue to be backed up at the primary database.

The following script can be run every day or once a week, depending on how much redo application can be tolerated in the event that all current control files are lost. If all control files are lost, refer to [Recover Using Backup Control File](#) section. If the script is run less frequently than once a week, the value of 7 days in the script should be changed to be greater than the interval between backups.

The following RMAN command sequence should be executed, after connecting to the primary database (as the target database) and the recovery catalog:

1. Frees disk space by deleting control files older than one week.
2. Deletes archive logs older than one week that have already been backed up to two separate backup sets on tape. This allows recovery to any point within the last week to be performed from disk, not tape, thus shortening overall completion time. You only need to backup the archive logs at the primary database site if you have not previously backed them up by running the [Standby Database Backup Procedure](#).
3. Copy all disk backups to tape

```
DELETE BACKUP OF CONTROLFILE COMPLETED BEFORE 'SYSDATE-7' DEVICE
TYPE DISK;

DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7' BACKED UP 2 TIMES
TO DEVICE TYPE SBT;

BACKUP DEVICE TYPE SBT BACKUPSET ALL;
```

Standby Database Backup Procedure

Weekly Backup Script

The following script can be run every week to make full backups of the standby database to disk and tape. The RMAN commands given in this section conform to a two-week recovery window.

The following RMAN command sequence is executed, after connecting to the standby database (as target database) and the recovery catalog:

1. Frees disk space by deleting full backups made more than two weeks ago.
2. Deletes archive logs older than a week that have already been backed up to two separate backup sets on tape. This allows recovery to any point within the last week to be performed from disk, not tape, thus shortening overall completion time.
3. Creates an incremental level 0 database backup
4. Allocates SBT channels and backup archive logs if they do not already exist on two different backup sets on tape
5. Copies all disk backups to tape

```
DELETE BACKUP OF DATABASE COMPLETED BEFORE 'SYSDATE-13' DEVICE TYPE
DISK;

DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7' BACKED UP 2 TIMES
TO DEVICE TYPE SBT;

BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 0 DATABASE;

BACKUP DEVICE TYPE SBT ARCHIVELOG ALL NOT BACKED UP 2 TIMES;

BACKUP DEVICE TYPE SBT BACKUPSET ALL;
```

Daily Backup Script

On the other days of your backup cycle, the following commands can be executed, after connecting to the standby database (as target database) and recovery catalog to:

1. Free disk space by deleting full backups made more than two weeks ago
2. Delete archive logs older than a week that have already been backed up to two separate backup sets on tape. This allows recovery to any point within the last week to be performed from disk, not tape, thus shortening overall completion time.

3. Make a level 1 incremental backup of the database
4. Allocate SBT channels and backup archive logs do not already exist on tape in two different backup sets.
5. Copy all disk backups to tape

```
DELETE BACKUP OF DATABASE COMPLETED BEFORE 'SYSDATE-13' DEVICE TYPE
DISK;

DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7' BACKED UP 2 TIMES
TO DEVICE TYPE SBT;

BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 1 DATABASE;

BACKUP DEVICE TYPE SBT ARCHIVELOG ALL NOT BACKED UP 2 TIMES;

BACKUP DEVICE TYPE SBT BACKUPSET ALL;
```

Case 2: Backups Directly Written to Tape

If all backups are to be written directly to tape, after connecting to the primary database (as the target database) and recovery catalog, set the default device type to tape by executing the following RMAN command:

```
CONFIGURE DEFAULT DEVICE TYPE TO SBT;
```

Primary Database Backup Procedure

The following script can be run every day or once a week, depending on how much redo application can be tolerated in the event that all current control files are lost. If all control files are lost, refer to [Recover Using Backup Control File](#). If the script is run less frequently than once a week, the value of 7 days in the script should be changed to be greater than the interval between backups.

The following RMAN commands should be executed, after connecting to the primary database (as target database) and recovery catalog:

1. Delete archive logs older than a week that have already been backed up to two separate backup sets on tape. This allows recovery to any point within the last week to be performed from disk, not tape, thus shortening overall completion time.
2. Copy all disk backups to tape

```
DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7' BACKED UP 2
TIMES TO DEVICE TYPE SBT;

BACKUP BACKUPSET ALL DELETE INPUT;
```

Standby Database Backup Procedure

Weekly Backup Script

As in Case 1, the frequency at which this script is run depends on how much redo application can be tolerated during recovery in the event that all current control files are lost. If all control files are lost, refer to [Recover Using Backup Control File](#).

If the script is run less frequently than once a week, the value of 7 days in the script should be changed so that it is longer than the interval between backups.

The following RMAN command sequence should be executed, after connecting to the standby database (as target database) and recovery catalog:

1. Delete archive logs older than a week that have already been backed up to two separate backup sets on tape. This allows recovery to any point within the last week to be performed from disk, not tape, thus shortening overall completion time.
2. Creates a database level 0 backup
3. Backup all archive logs that have not already been backed up to two separate backup sets on tape.

```
DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7' BACKED UP 2
TIMES TO DEVICE TYPE SBT;

BACKUP INCREMENTAL LEVEL 0 DATABASE PLUS ARCHIVELOG NOT BACKED
UP 2 TIMES;
```

Daily Backup Script

On the other days of the backup cycle, the following script should be executed, after connecting to the standby database (as target database) and the recovery catalog:

1. Delete archive logs older than a week that have already been backed up to two separate backup sets on tape. This allows recovery to any point within the last week to be performed from disk, not tape, thus shortening overall completion time.
2. Create a level 1 incremental backup of the database
3. Backup all archive logs that have not already been backed up to two separate backup sets on tape.

```
DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7' BACKED UP 2
TIMES TO DEVICE TYPE SBT;

BACKUP INCREMENTAL LEVEL 1 DATABASE PLUS ARCHIVELOG NOT BACKED
UP 2 TIMES;
```

RECOVERY PROCEDURE

These scripts assume that the backup can be restored to the standby or primary database since both host have configured the directory systems to be identical.

Recovery from Loss of Data Files on Primary Database

Execute the following RMAN commands to restore and recover data files. You must be connected to the primary database (as target database) and the recovery catalog. These scripts assume that the datafile to be recovered is offline.

```
RESTORE DATAFILE <n,m...>; # where n, m are datafile numbers or names
```

```
RECOVER DATAFILE <n,m...>;
```

Execute the following RMAN commands to restore and recover tablespaces. You must be connected to the primary database (as target database) and the recovery catalog.

```
RESTORE TABLESPACE <tbs_name1, tbs_name2, ...>  
RECOVER TABLESPACE <tbs_name1, tbs_name2, ...>
```

Recovery from Loss of Data Files on Standby Database

The managed recovery process (MRP) applies information from the archived redo logs to the standby database. When restoring and recovering a datafile(s) on the standby database, it is important that the archive logs are available on disk to satisfy MRP. You must be connected to both the standby and recovery catalog databases.

The following steps are required to recover a standby database datafile:

1. Determine the standby databases's current SCN

```
SQL> SELECT MAX(NEXT_CHANGE#)+1 UNTIL_SCN FROM  
V$LOG_HISTORY;  
  
UNTIL_SCN  
-----  
967786
```

2. Stop the MRP
3. Restore the datafile using RMAN

```
RESTORE DATAFILE <n,m...>; # where n, m are datafile numbers or names
```

4. Recover the datafile using RMAN until the standby databases's current SCN. If any archive logs are not on disk, RMAN will automatically restore them from a backup and apply them to the instance:

```
RECOVER DATABASE UNTIL SCN 967786;
```

5. Restart the MRP

Recovery from Loss of Control File on Standby Database

Loss of One Control File

Oracle allows multiplexing of the standby control file. To ensure that the standby control file is multiplexed, check the CONTROL_FILES initialization parameter, as follows:

```
SQL> SHOW PARAMETER CONTROL_FILES  
  
NAME                                TYPE        VALUE  
-----  
control_files                        string      <cfilepath1>,<cfilepath2>
```

If one of the multiplexed standby control files is lost or not accessible, Oracle stops the instance and writes the following messages to the alert log:

```
ORA-00210: cannot open the specified controlfile
ORA-00202: controlfile: '/ade/banand_hosted6/oracle/dbs/scf3_2.f'
ORA-27041: unable to open file
```

To recover from the loss of one control file you can either:

- Copy one of the other control files to the directory of the corrupt or missing locations specified by the SPFILE, or
- Edit the SPFILE to only use the existing control files.

Loss of all Standby Database Control Files

If all standby control files are lost, then you can restore the backup control file taken on the standby node. You must be connected to the standby database (as target database) and recovery catalog. All archive logs generated since the last archive log backup must be manually cataloged as outlined in [Archive Log Backup Considerations](#).

If a backup control file from the standby node is not available, then you must create a new control file from the primary database. You must be connected to the standby database (as target database) and recovery catalog.

The steps are:

1. Create a backup control file for the standby database from the primary database
2. Copy the control file to all multiplexed locations on the standby database as specified in the SPFILE
3. Restart the MRP
4. Recatalog archive logs

The newly created control file loses all information about archive logs generated prior to its creation.

Since RMAN parses the control file for the list of archive logs to backup, all the archive logs generated since the last archive log backup must be manually cataloged as described in [Archive Log Backup Considerations](#).

Recovery from Loss of Control File on Primary Database

Oracle allows multiplexing of the control file on the primary database. If one of the control files cannot be updated on the primary database, the primary database instance is shut down automatically. Similar to the [Recovery from Loss of a Control File on Standby Database](#) section, a good copy of the control file can be copied over the failed copy and the instance can be restarted without restore or recovery.

Loss of all control files

If all control files are lost on the primary, there are three options, depending on the length of acceptable downtime:

1. Failover to Standby Database

This option minimizes downtime. However, some amount of data may be lost, depending on the protection mode configured for the standby database.

The old primary database must also be re-created with its own backup and brought back as a standby, either prior to or after the failover SCN. Refer to the physical standby database failover procedure in the Oracle Data Guard Concepts and Administration documentation for more details.

2. Create a New Control File

This option incurs additional downtime compared to failover. A new control file can be created using the NORESETLOGS option followed by media recovery. The following SQL can be run on the standby database instance to generate a trace file:

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE  
NORESETLOGS;
```

The resulting trace file contains a SQL script that can be used to re-create the control file on the primary database in NOMOUNT state.

The re-created control file loses all information about the archive logs generated prior to control file creation time. If archive log backups are being performed on the primary database, all the archive logs generated since the last archive log backup must be manually cataloged as described in [Archive Log Backup Considerations](#).

3. Recover Using Backup Control File

If you are unable to create a control file using the previous procedure, then you can use a backup control file created at the primary database, perform complete recovery, and open with RESETLOGS.

To restore the control file and recover the primary database, use the following RMAN commands after connecting to the primary database (as target database) in NOMOUNT and recovery catalog:

```
RESTORE CONTROLFILE;  
  
ALTER DATABASE MOUNT;  
  
RECOVER DATABASE;
```

The RMAN message output will include information about which online log members were last applied for the current online redo log group. These online redo logs should be manually copied to and applied on the standby database before opening the primary database with RESETLOGS operation. In the example given below, `t_log1.f` and `t_log2.f` are active online log members that must be applied on standby to make it follow the new RESETLOGS branch.

```
archive log filename=/ade/banand_hosted6/oracle/work/arc_dest/arc1_23.dbf
thread=1 sequence=23

archive log filename=/ade/banand_hosted6/oracle/dbs/t_log1.f thread=1
sequence=24

archive log filename=/ade/banand_hosted6/oracle/dbs/t_log2.f thread=1
sequence=25

media recovery complete

Finished recover at 15-APR-03
```

Refer to the Oracle Data Guard Concepts and Administration documentation for procedure to apply redo logs on standby database.

To prevent invalidation of the standby database, perform manual recovery through RESETLOGS operation on the standby, as described in [Avoiding Standby Re-instantiation after RESETLOGS Operation on the Primary Database](#).

This is the most time-consuming option, but may be the only recourse if failover or re-creating the control file from standby options are not possible, or do not succeed.

Recovery from Loss of an Online Log

Oracle highly recommends multiplexing the online logs. The loss of all members of an online log group will cause Oracle to terminate the instance. If any of the members of a log file group cannot be written, they will not be used until they become accessible. The views V\$LOGFILE and V\$LOG contain more information about the current status of log file members in the primary database instance.

When Oracle is not able to write to one of the online log members, the following alert messages will be written:

```
ORA-00313: open failed for members of log group 1 of thread 1

ORA-00312: online log 1 thread 1:
'/ade/banand_hosted6/oracle/dbs/t1_log1.f'

ORA-27037: unable to obtain file status

SVR4 Error: 2: No such file or directory

Additional information: 3
```

If the access problem is temporary due to a hardware issue, correct the problem and Oracle will continue automatically. If the loss is permanent, a new member can be added and the old one dropped from the group.

To add a new member to a redo log group, use the SQL command:

```
ALTER DATABASE ADD LOGFILE MEMBER '<log_file_name>' REUSE
TO GROUP <n>;
```

To drop a member of a redo log group, use the SQL command:

```
ALTER DATABASE DROP LOGFILE '<lost_log_file_name>';
```

You can do this when the database is open, without affecting database availability.

Additional Scenarios Involving Loss of Online Logs

If all the members of a non-active group that have been archived are lost, the group can be dropped and re-created.

If all online log members for the current active group, or for an inactive group which has not yet been archived are lost and associated datafiles are lost, a failover to the standby database is necessary. Refer to the Oracle Data Guard Concepts and Administration documentation for more details on completing the physical standby failover procedure.

If a non-active member that hasn't been archived is lost, but no datafiles have been lost, clear that log group and continue. You must take a full backup of the primary, and the standby must be reinstated.

Incomplete Recovery of the Database

Database point-in-time recovery of a database is normally done in cases when the database is logically corrupted (by some user or application), or when a tablespace or data file is accidentally dropped from the database. The options to recovery from such errors are:

- Failover to the Standby Database. This option is only feasible if the logical corruption has not been propagated and applied to the standby database. Follow the physical standby failover procedures as prescribed in the Oracle Data Guard Concepts and Administration documentation.
- Perform database point-in-time (DBPITR) recovery on the primary database, as detailed below. This is the only option if the logical corruption has been propagated and applied to all standby database sites.

Database Point-In-Time Recovery

If all of the standby database instances have already been recovered past the point in time where the logical corruption was introduced, the following procedure is the only way to perform point-in-time recovery on the primary database:

- 1) Use LogMiner or other means to identify the time or SCN at which all the data in the database is known to be good.
- 2) Startup primary instance in MOUNT state. Connect to the recovery catalog database and primary instance. Using the time or SCN, execute the following RMAN commands to perform DBPITR

```
SET UNTIL TIME '<time>';  
RESTORE DATABASE;  
RECOVER DATABASE;
```

3) Open the database read only to verify that it has been recovered to the desired point in time. Additional redo can be applied if you need to roll the database further in time. Once the desired point is reached, perform the following command:

```
ALTER DATABASE OPEN RESETLOGS;  
RESET DATABASE;
```

4) All standby database instances must be re-instantiated. Follow the procedures outlined in [Standby Database Instantiation Using RMAN](#). For additional information, refer to the Oracle Data Guard Concepts and Administration documentation on standby database re-instantiation procedures.

Additional RMAN Backup Considerations

Avoiding Standby Re-instantiation after RESETLOGS Operation on the Primary Database

Re-instantiating the standby database is typically needed after a RESETLOGS operation on the primary database has been performed. However, re-instantiation of the standby database can be avoided if the standby database's current SCN is behind the primary database's SCN. Usually the SCN on the standby database is behind the primary database when it has implemented a delay in redo apply. For example, the standby database has implemented a delay in redo for 2 hours and a point-in-time recovery to 1 hour ago is performed on the primary.

The following procedure outlines the steps to avoid standby re-instantiation:

- On the primary database, obtain the SCN prior to and after the RESETLOGS operation.

```
SELECT RESETLOGS_CHANGE#, PRIOR_RESETLOGS_CHANGE#  
FROM V$DATABASE;
```

- After connecting to the primary database (as target database) and recovery catalog in RMAN, reset the primary database information in the recovery catalog to the incarnation corresponding to the PRIOR_RESETLOGS_CHANGE# obtained from the above SQL. Perform LIST INCARNATION to find the incarnation key corresponding to the PRIOR_RESETLOGS_CHANGE# and then RESET DATABASE TO INCARNATION.

```
LIST INCARNATION;  
  
RESET DATABASE TO INCARNATION <incarnation key corresponding  
to PRIOR_RESETLOGS_CHANGE#>;
```

- Copy any missing archive logs from primary to standby and catalog them using the following command:

```
CATALOG ARCHIVELOG 'archive log file name';
```

- If needed, copy the online logs (that have not yet been archived) from the primary database to the standby database as explained in [Recover Using Backup Control File](#) section.
- Connect to the standby database with SQL*Plus and recover the standby database to RESETLOGS SCN-1.

```
RECOVER STANDBY DATABASE UNTIL CHANGE
<RESETLOGS SCN - 1>;
```

Caution: If an incorrect online log was supplied for recovery, the standby database may need to be re-instantiated.

- Connect to the primary database, and create a standby control file using:

```
ALTER DATABASE CREATE STANDBY CONTROLFILE AS
'<filename>;
```

- Copy newly created standby control file to standby site and mount the standby using the new control file.
- Re-start the MRP to continue application of remaining logs. Note that when the MRP is started on standby instance, all the data file headers will be updated with new branch RESETLOGS data, as the data file headers are checkpointed at the RESETLOGS branch offline SCN (i.e. RESETLOGS SCN-1).
- Connect to the primary database (as target database) and recovery catalog in RMAN, and reset the primary database to the original RESETLOGS branch (RESETLOGS_CHANGE# from first step) to undo the previous RESET DATABASE operation.

```
RESET DATABASE TO INCARNATION <Incarnation Key corresponding to
RESETLOGS_CHANGE#>;
```

STANDBY DATABASE INSTANTIATION USING RMAN

The following procedure outlines a typical method using RMAN to instantiate standby databases. RMAN's DUPLICATE command restores the data files from backup sets and recovers the database (applying incrementals and archive logs) to the current or a specified UNTIL time/SCN. This can be useful for setting up a Data Guard environment, recover standby database after media failure or disaster, or re-instantiating the old primary database as a new standby database after a failover operation.

1. Create an initialization parameter file for the standby database. The SPFILE can be restored from backups using the RMAN RESTORE SPFILE command.
2. Start the standby instance in NOMOUNT using the SPFILE.

3. Perform any Oracle Net setup required to connect to the standby database host.
4. Generate a backup of the control file by executing the following RMAN command. You should be connected to the primary database and recovery catalog.

```
BACKUP CURRENT CONTROLFILE FOR STANDBY;
```

5. Use the control file backup and existing backups of data files and archive logs to instantiate a new standby database. Ensure that RMAN is connected to the primary database, catalog database, and standby database instance. Use the AUXILIARY keyword to connect to standby instance in NOMOUNT state:

```
RMAN TARGET <primary_db> CATALOG <catalog_db>  
AUXILIARY <new_standby_db>
```

6. Execute the following RMAN command to create a new standby database with the current time/SCN:

```
Duplicate target database for standby;
```

Variation Standby Database Instantiation using RMAN

The preceding procedure automates creation of standby database and works well when all backup sets are available at standby site. You can also use the following procedure to instantiate a standby database when all backup sets cannot be efficiently transmitted over the network to the standby site due to file sizes (e.g. terabyte databases). In this scenario, full backups on tape are shipped to the standby site while incremental backups are sent over the network.

1. Copy the instance parameter file and start the standby database instance in NOMOUNT.
2. Create a standby control file from the primary database.
3. Ship tapes of full backups from primary to standby node. Incremental backups can be sent over the network.
4. While incremental backups are being transmitted to the standby node, datafiles from the full backups can first be restored. Restore the datafiles after connecting to recovery catalog and standby database (as target database):

```
RESTORE DATAFILE <n,m>;
```

5. If all archive log backups are present at standby site, then execute:

```
RECOVER DATABASE;
```

to apply incrementals, and restore & apply archive logs from backups.

If archive logs backups are not present at standby site, once all datafiles are restored and all incremental backups are present at standby site, apply the incrementals only:

```
RECOVER DATABASE NOREDO;
```

6. Re-start MRP. If an archive log repository has been previously setup, any missing archive logs will be automatically retrieved and applied on the standby database. Otherwise, any missing archive logs must be manually copied to the standby database and applied.

Refer to the Oracle Data Guard Concepts and Administration documentation for additional details on creating a physical standby database.

CONCLUSION

Oracle Data Guard offers the most comprehensive disaster protection of your Oracle data assets, including extensive management and monitoring capabilities. Coupled with Oracle Recovery Manager, an out-of-the-box backup and recovery tool installed with the database, your environment can now offer restoration of the Oracle database in the event of media loss on the primary or standby database sites. RMAN greatly simplifies file restoration over tedious, OS-level copy commands. Creation and re-instantiation of standby databases can also be performed utilizing RMAN.

By incorporating these procedures into your recovery plan and performing thorough validation, you will have at your disposal a range of techniques to effectively respond to anything from media failure to complete disaster in your Data Guard environment.

APPENDIX

Inability to Utilize Backups Taken at the Originating Host

In certain environments, it may not be feasible to share backups across the primary and standby database sites due to geographical location, firewall, or other factors. In these cases, the RMAN TAG functionality should be used in backup and recovery procedures outlined in this paper. In addition, complete backups of the primary and standby database are necessary.

You can use the general strategies described in this paper, with the following changes:

- Backup files created by RMAN must be tagged with local system name, and on restores that tag must be used to restrict RMAN from selecting backups taken on the remote host. The BACKUP command must use the TAG '<node name>' option when creating backups, RESTORE command must use the FROM TAG '<node name>' option, and the RECOVER command must use FROM TAG '<node name>' ARCHIVELOG TAG '<node name>' options.
- Re-instantiation of the standby database must utilize the TAG syntax. The steps to re-instantiate the standby database is as follows.
 - Start the standby instance in NOMOUNT state using the same parameter files that the standby was operating earlier.
 - Create a standby control file on primary instance using the following SQL:

```
ALTER DATABASE CREATE STANDBY CONTROLFILE  
AS '<file name>';
```

- Use the new control file to mount standby instance.
- Execute the following RMAN commands to restore and recover the database files:

```
RESTORE DATABASE FROM TAG '<node name used in BACKUP command>';  
RECOVER DATABASE FROM TAG '<node name used in BACKUP command>'  
ARCHIVELOG TAG '<node name used in BACKUP command>';
```

- Re-start the MRP.

Standby Database Configured as Archive Log Repository

A standby database can be configured as an archive log repository to serve as a backup for archive logs. The repository does not contain data files, and can be used by other standby databases to retrieve missing archive logs. For more information on this configuration, see the Oracle Data Guard Concepts and Administration documentation.

The scripts provided in section [Backup Procedure](#) are still valid for backing up archive log repositories. However, omit the RMAN commands that back up data files, since there are no data files kept, and MRP is not run.

Standby Database Filenames Differ from Primary Database

This section addresses restoring and recovering either primary or standby database when the filenames differ between the two. When RMAN registers a database in the recovery catalog, it records the datafile names as they are known by the control file. When using RMAN in the Data Guard environment, the datafile names are recorded in the recovery catalog based on the primary database control file.

Due to this behavior, the restore and recover commands will be slightly different than the ones specified earlier in this paper. For example, when restoring the standby database from the primary database backup, the actual data file names on the standby database can be obtained from V\$DATAFILE view and must be specified in SET NEWNAME option for all the data files, as follows:

```
RUN
{
SET NEWNAME FOR DATAFILE 1 TO '<existing file location for file#1 from
V$DATAFILE>';
SET NEWNAME FOR DATAFILE 2 TO '<existing file location for file#2 from
V$DATAFILE>';
...
...
SET NEWNAME FOR DATAFILE n TO '<existing file location for file#n from
V$DATAFILE>';
RESTORE {DATAFILE <n,m,...> | TABLESPACE <tbs_name_1, 2, ... | DATABASE};
SWITCH DATAFILE ALL;
RECOVER DATABASE {NOREDO};
}
```

Likewise, before executing an RMAN DUPLICATE, SET NEWNAME should be used to specify new filenames during the standby database creation.

This can also work when restoring files to the primary database from the standby database, as all backups except control file backups are usable on all databases that have the same DBID.



Using Oracle Recovery Manager with Oracle Data Guard in Oracle9i
March 2004

Author: Anand Beldalker, Timothy Chien

Contributing Authors: Steven Wertheimer, Antonio Romero, Ashish Ray, Lawrence To, Douglas Utzig, Tammy Bednar

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2004 Oracle Corporation
All rights reserved.