Using Recovery Manager with Oracle Data Guard in Oracle Database 10*g*

An Oracle White Paper
April 2009

ORACLE

Using Recovery Manager with Oracle Data Guard in Oracle Database 10*g*

Using Recovery Manager with Oracle Data Guard in Oracle Database 10*g*

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 backup control files, data files, and archived log files to disk and tape while efficiently recovering these files upon a file system or media loss. RMAN is Oracle's database backup and recovery utility that backs up all data with minimal impact on production databases and quickly recovers from the loss of individual files, or the entire database. In Oracle Database 10g, RMAN offers new features for the Data Guard configuration, including:

- Flash Recovery Area, a single filesystem or ASM disk group to organize and manage database recovery related files such as backups and archived logs.
- Standby database server-specific persistent configurations, including channel, device, backup optimization, and control file autobackup settings
- Primary and standby database server-specific configuration to enable automatic deletion of archived logs that have been applied to remote standby destinations. Archived logs are automatically deleted when more space in the Flash Recovery Area is needed for new files.

This paper outlines RMAN procedures to setup and backup *physical* standby databases managed by Data Guard in an Oracle Database 10g environment. Note that only backups from a *physical* standby database can be used to recover the primary database.

The procedures outlined include:

- Configuring RMAN persistent settings on primary and standby databases
- 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 using backups that are made on the standby database

This paper is for DBAs, IT, and system administrators who are interested in the RMAN procedures to manage backups in their Data Guard configuration. The paper assumes familiarity with Data Guard and RMAN concepts and procedures.

Note: For the RMAN procedures in an Oracle9i Database configuration, refer to Using Recovery Manager with Oracle Data Guard in Oracle9i [1].

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 a tool integrated with the Oracle Database that satisfies the demands of high performance, manageable backup and recovery. RMAN is designed to work intimately with the server, providing block-level corruption detection during backup and restore. RMAN optimizes performance and space consumption during backup with file multiplexing and compression, and operates with leading backup software systems via the supplied Media Management Library (MML) API.

RMAN brings rich functionality such as online backups, incremental backups, block media recovery, automation of backup management tasks, and integration with 3rd party media management systems into the Data Guard configuration. Since RMAN and Data Guard are part of the integrated Oracle High Availability technology stack, RMAN backups can be seamlessly offloaded to a physical standby database, allowing customers to gain more value out of their disaster recovery investment. Backups do not impact normal Data Guard operation – they can be taken while the standby database is in recovery or read-only mode. Backups can be used to recover either primary or standby database servers.

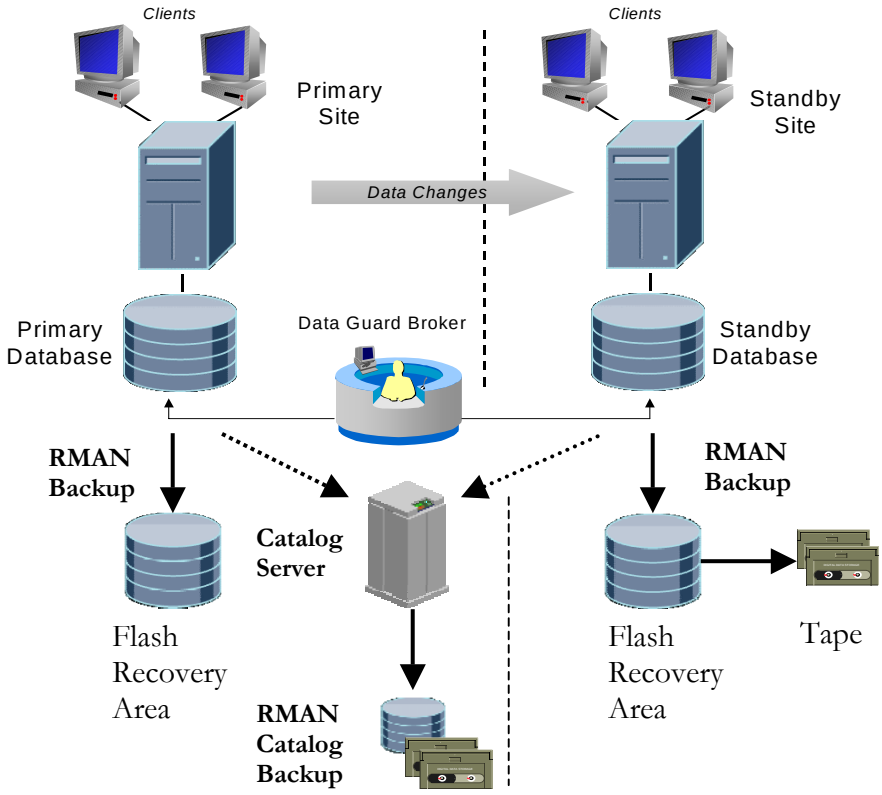

Fig. 1: Data Guard & RMAN Configuration

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. Data Guard and RMAN are both fully supported features of the Oracle Database Enterprise Edition (RMAN is also provided with Oracle Database Standard Edition).

The following sections cover:

- 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

SETUP ASSUMPTIONS

The assumptions for this setup are:

- The standby database is a *physical* standby database and backups are only taken on the standby database. Refer to the Appendix for procedural changes if backups are taken on both primary and standby databases.
- The data file directories on the primary and standby database are identical. This simplifies the RMAN backup and recovery operations no matter which host is used.
- RMAN Recovery Catalog is required so that backups taken on one database server can be restored onto another database server. Using just the control file as the RMAN repository is not sufficient, as the primary database will have no knowledge of backups taken on the standby database.

  The RMAN Recovery Catalog organizes backup histories and other recovery-related metadata in a centralized location. The recovery catalog is configured in a database and maintains backup metadata. A recovery catalog does not have the space limitations of the control file and can store more historical data about backups.

  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.
- All databases in the configuration use Oracle Database 10g Release 1 or Release 2.
- Primary database does not use Oracle Managed Files (OMF). When using OMF, standby database filenames can vary from those on the primary. Refer to the Appendix for modifications to the restore procedures when standby database filenames are different than those on the primary.
- 3rd party media management software is configured with RMAN to make backups to tape.

Note: The Appendix describes modifications to these procedures for three alternate 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 archived log repository
- Standby database file names are different than those on the primary

CONFIGURATION SETTINGS AND CONSIDERATIONS

In a Data Guard configuration, the process of backing up data files and archived 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.

The following settings for RMAN and the Oracle database simplify the backup and recovery operations, and are based on datafile and archived log backups being made at one of the standby database servers.

Recommended Oracle Database Configuration

On primary and standby databases:

- Configure Flash Recovery Area

  The Flash Recovery Area is a single storage location on a filesystem or Automatic Storage Management (ASM) disk group where all files needed for recovery reside. These files include the control file, archived logs, online log copies, flashback logs, and RMAN backups. As new backups and archived logs are created in the Flash Recovery Area, older files (which are either outside of the retention period, or have been backed up to tertiary storage) are automatically deleted to make room for them. In addition, notifications can be setup to alert the DBA when flash recovery area space consumption is nearing its predefined limit; the DBA can then take action such as increasing the recovery area space limit, adding disk hardware, or decreasing the retention period.

  Configure the Flash Recovery Area, by setting the following `init.ora` parameters:

  DB_RECOVERY_FILE_DEST = <mount point or ASM Disk Group>

  DB_RECOVERY_FILE_DEST_SIZE = <disk space quota>

- Use a system parameter file (SPFILE) so that it can be used with any database in the Data Guard configuration. This allows restoring of the SPFILE from a backup taken on another database.

- Uniquely name archived log and backup directories for each database. For example, on a primary database named 'BOS', archived logs are written to '/archivelog/BOS' directory, whereas on a standby database named 'SF', archived logs are written to '/archivelog/SF' directory. This allows RMAN commands, such as BACKUP ARCHIVELOG, CROSSCHECK, and DELETE, to be used with the LIKE option to select the appropriate archived logs for a particular database. Refer to Archived Log and Backup Management for more details.

- Uniquely tag backups, if backups are taken on primary and standby databases. For example, primary database full backups can be tagged 'BOS_FULL_BACKUP', while standby database full backups can be tagged 'SF_FULL_BACKUP'. This allows for proper selection of backups that were taken on a particular database when performing RMAN maintenance operations. Refer to Archived Log and Backup Management for more details. Tags can also be used to select specific backups to be used for restore and recovery operations.

- Enable Flashback Database on primary and standby databases. When Flashback Database is enabled, Oracle maintains flashback logs in the Flash Recovery Area. These logs can be used to 'rewind' the database back to an earlier point in time, without requiring a complete restore. Refer to Oracle Flashback Technology: Alternative to Point-in-time Recovery [2] in

the Oracle Database Backup and Recovery Advanced User's Guide for more information.

Recommended RMAN Settings

The following CONFIGURE commands should be issued, after connecting to the primary database and recovery catalog:

- CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <n> DAYS
  This command will update the primary database control file and then be immediately recorded in the catalog. The standby database will use this retention policy when connected to the catalog. This command should only be executed at the primary database, since the standby control file contains a subset of the primary control file's records. All backup files and archived logs on the standby database will be retained for at least <n> days. The Flash Recovery Area will automatically delete obsolete files or files already backed up to tape, if needed, to reclaim space for new files.
- CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON STANDBY

  By setting this configuration at the primary database, it will enable automatic deletion of archived logs on the primary database that have been applied to remote standby destinations. By default, this configuration requires that at least one remote destination is set to mandatory.
  Note: Mandatory standby destination can impact the primary database if the standby destination cannot be reached. To enable this feature without using mandatory standby destination, refer to Metalink Note 331924.1 (RMAN backups in Max Performance/Max Availability Data Guard Environment).
- CONFIGURE CONTROLFILE AUTOBACKUP ON
  to enable automatic backups of the control file and SPFILE.
- CONFIGURE CHANNEL DEVICE TYPE SBT PARMS '<channel parameters>'
  to set up the channel parameters that are required by the media management software.

The following commands should be issued, after connecting to the standby database server where backups are made, and the recovery catalog:

- CONFIGURE CONTROLFILE AUTOBACKUP ON
  to enable automatic backups of the control file and SPFILE. The backups will be written to the Flash Recovery Area.
- CONFIGURE BACKUP OPTIMIZATION ON
  to skip backups of database files for which there already exists a valid backup within the retention period.

- CONFIGURE CHANNEL DEVICE TYPE SBT PARMS '<channel parameters>' to set up the channel parameters that are required by the media management software.
- CONFIGURE ARCHIVELOG DELETION POLICY TO NONE

  This will enable automatic deletion of archived logs on the standby database (where backups are being taken) that are outside of the retention period or have already been backed up to tape, if additional space is needed for new backups or archived logs.

The following commands should be issued, after connecting to each of the other standby database servers, and the recovery catalog:

- CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON STANDBY

  By setting this configuration on each of the other standby databases (where backups are not being taken), it will enable automatic deletion of archived logs on this standby database that have been applied to all other remote standby destinations. By default, this configuration requires that at least one remote destination is set to mandatory. Archived logs are deleted if space in the Flash Recovery Area needs to be reclaimed for new files.

  Note: Mandatory standby destination can impact the primary database if the standby destination cannot be reached. To enable this feature without using mandatory standby destination, refer to Metalink Note 331924.1 (RMAN backups in Max Performance/Max Availability Data Guard Environment).

Note that in the event of a switchover or failover, the database role changes and the appropriate CONFIGURE commands must be re-executed on new primary and standby databases. Refer to the section "Modifications to Procedure/Configuration Upon Switchover/Failover" for additional changes .

BACKUP PROCEDURES

This section details the RMAN scripts and procedures to backup the Oracle database in a Data Guard configuration.

Note that Oracle's Maximum Availability Architecture (MAA) best practices recommend that backups be taken at both primary *and* standby databases to reduce MTTR in case of double outages and to avoid introducing new site practices upon switchover/failover. Refer to the Appendix for modifications to the general procedures, if backups are taken on both the primary and standby databases.

Case 1: Use Disk as Cache for Tape Backup

In this scenario, the Flash Recovery Area on the standby database serves as a disk cache for tape backup. Disk is used as the primary storage for backups, with tape providing long term, archival storage. Full backups are taken weekly, with incremental taken daily.

Primary Database Backup Procedure

The primary database control file and SPFILE autobackups should be backed up to tape using the following RMAN command, after connecting to the primary database (as the target database) and the recovery catalog:

```
BACKUP DEVICE TYPE SBT BACKUPSET ALL;
```

Note that any existing disk backups (from a previous backup strategy) will be placed on tape when this command is run.

How often to run this backup is dependent on your recovery window. Upon recovery, a more recent backup control file requires less application of redo, and hence, less time, to get up to date, than an older backup control file. It is recommended to back up the primary database control file to tape at least once a week.

Standby Database Backup Procedure

Daily Backup Script

The recommended backup strategy is the "Oracle-suggested strategy", which takes advantage of *incrementally updated backups*. With this feature, datafile image copies can be rolled forward with the latest incremental backups, thereby providing always up to date datafile image copies. RMAN uses the resulting image copy for media recovery just as it would use a full image copy taken at that SCN, without the overhead of performing a full image copy of the database every day. Time-to-recover is also reduced since the image copy is updated with the latest block changes and fewer redo logs are required to bring the database back to the current state.

In the "Oracle-suggested strategy", a full database backup is taken on the first day, followed by an incremental backup on day two. Archived redo logs can be used to recover the database to any point in either day. For day three and onwards, the previous day's incremental backup is merged with the data file copy and a current incremental backup is taken, allowing fast recovery to any point within the last day and redo logs can be used to recover the database to any point during the current day.

The following commands can be executed, after connecting to the standby database (as target database) and recovery catalog:

1. RECOVER COPY OF DATABASE WITH TAG 'OSS';

   Roll forward level 0 copy of the database by applying the level 1 incremental taken the day before. In the example script below, the previous day's incremental level 1 was tagged 'OSS'. This incremental is generated by the BACKUP command in step 2.

   On the first day the script is run, there will be no roll forward, as there is no incremental level 1, yet. A level 0 incremental will be created in step 2.

   On the second day, there is again, no roll forward, as we only have a level 0 incremental. A level 1 incremental tagged 'OSS' will be created in step 2.

On the third and following days, the roll forward will be performed using the level 1 incremental tagged 'OSS' created in the previous day.

2. BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'OSS' DATABASE;

   Create a new level 1 incremental. On the first day the script is run, this will be a level 0 incremental. On the second and following days, this will be a level 1 incremental.

3. BACKUP ARCHIVELOG ALL NOT BACKED UP TO SBT;

   BACKUP BACKUPSET ALL;

   On the first day the script is run, any existing disk backups (from a previous backup strategy) will be placed on tape when this command is run.

   On the following days, archived log and incremental backups in the recovery area will be backed up to tape.

The full command sequence is:

```
RECOVER COPY OF DATABASE WITH TAG 'OSS';
BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 1 FOR
  RECOVER OF COPY WITH TAG 'OSS' DATABASE;
BACKUP ARCHIVELOG ALL NOT BACKED UP TO SBT;
BACKUP BACKUPSET ALL;
```

The standby control file will be automatically backed up at the conclusion of the backup operation since control file autobackup is enabled.

Weekly Backup Script

All recovery area files on disk are backed up once a week to tape with the following command:

```
BACKUP RECOVERY FILES;
```

This ensures that all current incremental, image copy, and archived log backups on disk are backed up to tape.

Case 2: Backups Directly Written to Tape

After connecting to the standby database server (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;
```

In this scenario, full backups are taken weekly, with incremental taken daily on the standby database.

Primary Database Backup Procedure

Backup control file and SPFILE autobackups to tape with the following RMAN command, after connecting to the primary database (as target database) and recovery catalog:

```
BACKUP BACKUPSET ALL;
```

How often to run this backup is dependent on your recovery requirements; the more frequent control file backups are taken, the lower the risk of lost redo if an older backup control file must be used. It is recommended to back up the primary database control file to tape at least once a week.

Standby Database Backup Procedure

Daily Backup Script

After connecting to the standby database (as target database) and the recovery catalog, the following command should be executed to:

- Create a level 1 incremental backup of the database, including all archived logs. On the first day this is run, if no level 0 backups are found, then a level 0 backup will be created.

```
BACKUP AS BACKUPSET INCREMENTAL LEVEL 1 DATABASE PLUS
    ARCHIVELOG;
```

The standby control file will be automatically backed up at the conclusion of the backup operation since control file autobackup is enabled.

Weekly Backup Script

One day per week, after connecting to the standby database (as target database) and recovery catalog, the following RMAN command should be executed to:

- Create a level 0 database backup, including all archived logs

```
BACKUP AS BACKUPSET INCREMENTAL LEVEL 0 DATABASE PLUS
    ARCHIVELOG;
```

Backup Encryption

Resilient data encryption is a cornerstone of an effective data security plan. Even if data somehow falls into the wrong hands, as long as it cannot be comprehensibly read, it cannot be used for malicious purposes. This is no less true for database backups and RMAN offers 128, 192, and 256-bit AES encryption for backup sets taken to disk or tape, with Oracle Wallet and/or passphrase-based authentication. Refer to the Backup & Recovery Advanced User's Guide [3] for guidance on configuring backup encryption.

ARCHIVED LOG AND BACKUP MAINTENANCE

If automatic archived log deletion is not configured (or is set to NONE) at any of the standby servers, the RMAN DELETE command with 'BACKED UP <n> TIMES TO DEVICE TYPE SBT' option can be used to explicitly delete archived logs that have been backed up to at least <n> copies on tape. Using the 'BACKED UP' option will ensure that a backup exists before archived logs are deleted.

Note that the Flash Recovery Area will automatically delete archived logs that are not required for the retention period or already backed up to tape, when more disk space for the Flash Recovery Area is needed for new files. The DELETE command is useful if additional space needs to be immediately reclaimed.

For example, the following command deletes all archived logs that were generated prior to 7 days ago and have at least 2 backup copies on tape:

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

Provided that archived log directories are uniquely named for each database, RMAN commands, such as BACKUP ARCHIVELOG, CROSSCHECK, and DELETE, with the LIKE option can be used to select the appropriate archived logs when connected to primary or standby database.

For example, to crosscheck and delete expired archived logs on a standby database, where '/archivelog/SF' is the archived log directory, connect to the standby database and recovery catalog, then:

```
CROSSCHECK ARCHIVELOG LIKE '/archivelog/SF%';

DELETE EXPIRED ARCHIVELOG LIKE '/archivelog/SF%';
```

Provided that backups are uniquely tagged for each database, RMAN maintenance commands, such as CROSSCHECK and DELETE, with the TAG option can be used to select the appropriate backups when connected to primary or standby database.

For example, to crosscheck database backups taken on the standby database named 'SF' that were previously tagged with 'SF_FULL_BACKUP', connect to the standby database and recovery catalog, then:

```
CROSSCHECK BACKUP OF DATABASE TAG 'SF_FULL_BACKUP';
```

Note: DELETE with 'FORCE' option should not be used in this environment as this can remove metadata from the recovery catalog for a file (e.g. archived log, or backup) that does not physically exist at the connected target database, but was created on a different database, whether a primary or standby database. If this occurs, the missing file must be re-cataloged at the appropriate database.

RECOVERY PROCEDURES

The following recovery scripts assume that primary and standby hosts have identical directory pathnames. If they are different, refer to the section "Standby Database Filenames Differ from Primary Database" for the required syntax.

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. If Real Time Apply is enabled, then MRP applies directly from the standby redo logs. When restoring and recovering a datafile(s) on the standby database, it is important that the archived 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. Stop the MRP
2. Determine the standby database's current SCN

```
SQL> SELECT MAX(NEXT_CHANGE#)+1 UNTIL_SCN FROM
V$LOG_HISTORY WHERE RESETLOGS_CHANGE# = (SELECT
RESETLOGS_CHANGE# FROM V$DATABASE);

UNTIL_SCN

------- ----------------

967786
```

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 database's current SCN. If any archived logs are not on disk, RMAN will automatically restore them from a backup and apply them to the restore datafile.

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

5. Restart the MRP

Recovery from Loss of Data Files on Primary Database

Execute the following RMAN commands to restore and recover data files for the primary database. 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 in the primary database. 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 Control File on Standby Database

Loss of One Control File

Oracle provides the ability to multiplex the standby control file. To confirm that the standby control file is multiplexed, check the CONTROL_FILES initialization parameter using the following 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: '/../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 CONTROL_FILES initialization parameter, or
- Edit the CONTROL_FILES initialization parameter to use only the available control files and re-start the standby database. Refer to the procedures for re-starting the standby database [4] in the Oracle Data Guard Concepts and Administration Guide for additional details.

Loss of all Standby Database Control Files

If all standby control files are lost, then use RMAN to restore the backup control file taken on the standby server.

1. Connect to the standby database (as target database) and recovery catalog.
2. Restore backup control file

```
RESTORE STANDBY CONTROLFILE;
```

3. All archived logs generated since the last archived log backup must be manually cataloged as outlined in the section "Archived Log Backup Considerations".

If a backup control file for the standby server is not available, then a new control file must be created from the primary database.

The steps are:

1. Create a new standby control file from the primary database
2. Copy the new control file to all multiplexed locations on the standby database as specified in the SPFILE and mount the standby database.
3. Restart the MRP
4. Connect to the standby database (as target database) and recovery catalog in RMAN
5. Manually catalog all the archived logs generated since the last archived log backup as described in the section "Archived 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 steps described in the section "Recovery from Loss of a Control File on Standby Database", 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. <Flashback is not possible if all control files are lost. So, restore & recover is only option> If the old primary database is still intact, it can simply be flashed back to the "failover SCN", the SCN at which the old standby database became the new primary database. The "failover SCN" can be found using the following SQL command:

```
SELECT TO_CHAR(STANDBY_BECAME_PRIMARY_SCN) FROM
   V$DATABASE;
```

After the flashback procedure, the new standby database will automatically catch up on redo from the new primary database. The flashback procedure [5] can be found in the Oracle Data Guard and Administration documentation.

If the old primary database is not intact or recoverable, it will need to be re-created from a backup of the new primary database and brought back as a new standby database. The steps to re-create the new standby database [6] can be found in the Oracle Data Guard Concepts and Administration documentation.

Refer to the physical standby database failover procedure [7] in the Oracle Data Guard Concepts and Administration documentation for additional 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 archived logs generated prior to control file creation time. For releases prior to 10.1.0.5, if archived log backups are being performed on the primary database, all the archived logs generated since the last archived log backup must be re-cataloged as described in Archived 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 from the primary database, perform complete recovery, and open with RESETLOGS.

To recover the control file and make the primary database available, 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;

ALTER DATABASE OPEN RESETLOGS;
```

When the archived log from the new redo branch created by the above RESETLOGS option is received at the standby database, the standby database will automatically register the new redo branch and terminate the MRP. Upon restarting MRP at the standby database server, the standby database will automatically follow the new redo branch.

The new recovery through RESETLOGS feature introduced in Oracle Database 10g allows administrators to recover primary and standby databases from backups taken in a previous redo branch (incarnation). Therefore, there is no need to make a full backup of the database after a RESETLOGS operation.

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

Recovery from Loss of an Online Log

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. Refer to the High Availability Architecture and Best Practices documentation for the procedures [8]on recovering from the loss of partial or entire online log groups.

Block Media Recovery of the Primary Database

In the event of block corruption on primary database, block media recovery (BMR) can be used to quickly repair the bad blocks. This type of recovery is most beneficial when a small number of blocks are corrupted or lost, rather than widespread datafile corruption. All blocks backed up or restored operation are checked for physical and, optionally, logical corruption. The RMAN VALIDATE command can also be used to check for block corruption.

BMR works by replacing previously identified corrupted blocks with valid blocks from the most recent backup, and performs media recovery using the necessary archived logs on just the affected blocks. Whether BMR is performed on primary or standby database, all backups and archived logs must be accessible.

BMR can be performed while the database remains online and operational.

More information on performing block media recovery can be found in the Backup and Recovery Advanced User's Guide.

Incomplete Recovery of the Database

Point-in-time recovery is normally done in cases when the primary database is logically corrupted (by a user or application), or when a tablespace or data file is accidentally dropped from the database. The options to recover from such errors are, in preferential order:

- If logical corruption has not propagated to the standby database, flashback primary database prior to logical corruption, open RESETLOGS on primary database, and restart redo apply on standby database. If flashback is not configured at primary database, then perform recovery on standby database to a point-in-time prior to the logical corruption, perform failover, and activate the standby database as the new primary database. The new standby database at the old primary site must be re-created.
- If logical corruption has propagated to the standby database, flashback primary and standby database prior to logical corruption, open RESETLOGS on primary database, and re-start apply on standby database. If flashback is configured on only one standby database server, perform flashback prior to logical corruption on that server, failover, and activate the standby database as the new primary database. All other standby databases must be re-created as new standby databases that will follow the new redo branch.

These procedures [9] are covered more in-depth in the Oracle Data Guard Concepts and Administration Guide.

RESYNC AFTER STRUCTURAL CHANGES ON PRIMARY DATABASE

Any structural changes to the primary database (e.g. add/drop data file) requires a resync to the recovery catalog, so that the standby database will know about these changes for subsequent restore operations. This can be accomplished by taking daily backups of the control file and SPFILE autobackups to tape (outlined previously), which perform an implicit full resync, or by connecting to the primary database (as target database) and recovery catalog, and running:

```
RESYNC CATALOG;
```

MODIFICATIONS TO PROCEDURE/CONFIGURATION FOLLOWING SWITCHOVER/FAILOVER

In the event of a switchover or failover, the database role changes and RMAN configuration settings also need to change. Refer to Recommended RMAN Configuration to set the appropriate configurations for the new primary database, the new standby database where backups will be made, and any other standby databases.

Archived Log Backup Considerations

There are two cases at the standby database where archived logs received after the last archived log backup must be explicitly made known to RMAN, or "re-cataloged". This occurs when:

- The primary or standby control file is re-created.
- The database role changes to standby or primary after switchover or failover operation. Note: this does not apply to 10.1.0.5 or higher database releases.

For example, in the case of a switchover or failover, connect to the new primary database, and execute the following RMAN command:

```
CATALOG ARCHIVELOG '<archived log filename 1>', '<archived log filename 2>', etc. ;
```

'<Archived log filename 1>', etc. refers to archived logs that are generated after the last archived log backup. For example, if your regular backup job starts at 10 am every day, and finishes at 11 am, then you will need to catalog all archived logs generated after 11 am up to the time of the switchover. These archived logs will then be backed up during the next regular backup job.

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

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 incremental and archived logs backups) to the current or a specified UNTIL time/SCN. This procedure can be useful for setting up a Data Guard configuration, recovering a standby database after media failure or disaster, or re-instantiating the old primary database as a new standby database after a failover operation.

1. Install Oracle database on the standby server.
2. Create an initialization parameter file for the standby database. The SPFILE can be restored from backups using the RMAN RESTORE SPFILE command.
3. Start the standby instance in NOMOUNT using the SPFILE.
4. Perform any Oracle Net setup required to connect to the standby database host.
5. Generate a backup of the control file by executing the following RMAN command. You should be connected to the primary database as target and recovery catalog.

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

6. Use the control file backup and existing backups of data files and archived 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>
   ```

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

   ```
   DUPLICATE TARGET DATABASE FOR STANDBY
     NOFILENAMECHECK DORECOVER;
   ```

Alternative Method for Standby Database Instantiation using RMAN

The above procedure automates creation of the standby database and works well when all backup sets are available at the 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 or archived log backups are sent over the network.

1. Take an incremental level 0 database backup on the primary database. Ship full backup on tape to standby site.
2. While backup is being taken to tape and shipped to standby site, install the Oracle database software on the standby server.
3. Copy the instance parameter file (SPFILE) to the standby server. Set DB_UNIQUE_NAME at standby server to a different name than on primary.
4. Create a standby control file from the primary database, copy it to standby database server, and start the standby database in MOUNT.
5. Once all tapes are accessible by the standby server, connect to recovery catalog and standby database (as target database) and restore all data files:

   ```
   RESTORE DATABASE;
   ```

6. Once all data files are restored:

   - If the amount of archived logs generated since the level 0 backup cannot be applied in a reasonable timeframe to recover the standby database, then an incremental backup can be used for faster recovery. Refer to the section "Standby Database Roll Forward Using RMAN Incremental Backup" for the procedure.
   - Otherwise, to use archived logs to recover the standby database:

     a. Take a backup of all archived logs at primary site:

        ```
        BACKUP ARCHIVELOG ALL NOT BACKED UP SINCE
        TIME 'SYSDATE - <# of days since level 0 backup>';
        ```

b. Send this backup over the network to the standby site.

c. If the archived log backup on the standby is in a different directory than on primary, while connected to standby database (as target database), catalog the archived log backup:

```
CATALOG START WITH '<directory for archived log backup>';
```

d. Recover the database:

```
RECOVER DATABASE;
```

7. At primary database, enable redo transport to standby database.

8. Re-start MRP. If FAL_SERVER and FAL_CLIENT parameter has been previously setup, any missing archived logs will be automatically fetched and applied on the standby database. Otherwise, any missing archived logs should be manually copied to the standby database, re-cataloged as stated in the section "Archived Log Backup Considerations", and applied.

STANDBY DATABASE ROLL FORWARD USING RMAN INCREMENTAL BACKUP

In cases where a physical standby database is far behind the primary database, an RMAN incremental backup can be used to roll the standby database forward faster than redo log apply. In this procedure, the RMAN `BACKUP INCREMENTAL FROM SCN` command, available with Oracle Database 10*g* Release 2, is used to create an incremental backup on the primary database that starts at the current SCN of the standby and is used to roll forward the standby database.

1. On standby database, stop managed recovery process (MRP):

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

2. On standby database, find the SCN which will be used for the incremental backup at the primary database:

```
SQL> SELECT CURRENT_SCN FROM V$DATABASE;
```

3. In RMAN, connect to primary database and create incremental backup from the SCN derived in previous step:

```
BACKUP INCREMENTAL FROM SCN <SCN from previous step>
DATABASE FORMAT '/tmp/ForStandby_%U' tag
'FORSTANDBY';
```

4. In RMAN, connect to primary database and create standby control file backup:

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

5. Copy the standby control file backup to standby system (e.g. /tmp/)

6. Shutdown standby database and startup nomount.

7. In RMAN, connect to standby database and restore the standby control file:

```
RESTORE STANDBY CONTROLFILE FROM
'/tmp/o1_mf_TAG20070220T151030_.bkp';
```

8. Shutdown standby database and startup mount.

9. If primary and standby database data file directories are identical, skip to step 10. Otherwise, if data file directories are different between primary and standby database, in RMAN, connect to standby database, catalog the standby data files, and switch standby database to use the just-cataloged data files:

```
CATALOG START WITH '+DATA_1/CHICAGO/DATAFILE/';

SWITCH DATABASE TO COPY;
```

10. Copy or ftp the incremental backup to the standby system (e.g. /tmp/ForStandby/). In RMAN, connect to standby database and catalog the backup:

```
CATALOG START WITH '/tmp/ForStandby';
```

11. Recover the standby database with the cataloged incremental backup:

```
RECOVER DATABASE NOREDO;
```

12. If primary and standby database redo log directories are identical, skip to step 14. Otherwise, on standby database, use an OS utility or `asmcmd (` if ASM-managed database) to remove all online and standby redo logs from the standby directories and ensure that the `LOG_FILE_NAME_CONVERT` parameter is properly defined to translate log directory paths, e.g. `LOG_FILE_NAME_CONVERT='/BOSTON/','/CHICAGO/'`.

13. On standby database, clear all standby redo log groups:

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 1;
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 2;
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 3;
…
```

14. On standby database, restart Flashback Database:

```
SQL> ALTER DATABASE FLASHBACK OFF;
SQL> ALTER DATABASE FLASHBACK ON;
```

15. On standby database, restart MRP:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY
DATABASE DISCONNECT;
```

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, the Oracle database can be fully recovered, in the event of media loss on the primary or standby database sites.. 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 a wide range of techniques to effectively respond to media recovery outages in your Data Guard configuration.

REFERENCES

1. Using Recovery Manager with Oracle Data Guard in Oracle9i:

   http://www.oracle.com/technology/deploy/availability/pdf/RMAN_DataGuard_9iR2_wp.pdf

2. Oracle Flashback Technology: Alternative to Point-in-time Recovery, Oracle Backup & Recovery Advanced User's Guide:

   http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10734/rcmflash.htm#1020720

3. RMAN Encrypted Backups, Oracle Database Backup and Recovery Advanced User's Guide:

   http://download-west.oracle.com/docs/cd/B19306_01/backup.102/b14191/rcmbackp.htm#CEGEJABH

4. Starting Up and Shutting Down a Physical Standby Database, Oracle Data Guard Concepts and Administration Guide:

   http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10823/manage_ps.htm#1007684

5. Using Flashback Database After a Failover, Oracle Data Guard Concepts and Administration Guide:

   http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10823/scenarios.htm#1017193

6. Creating a Physical Standby Database, Oracle Data Guard Concepts and Administration Guide:

   http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10823/create_ps.htm#67525

7. Failovers Involving a Physical Standby Database, Oracle Data Guard Concepts and Administration Guide:

http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10823/role_management.htm#1024703

8. Decide Which Recovery Action to Take, Oracle High Availability Architecture and Best Practices:

   http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10726/recover.htm#sthref734

9. Recovering Through the OPEN RESETLOGS Statement, Oracle Data Guard Concepts and Administration Guide:

   http://download-west.oracle.com/docs/cd/B14117_01/server.101/b10823/manage_ps.htm#1026452

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 '<server name>' option when creating backups, RESTORE command must use the FROM TAG '<server name>' option, and the RECOVER command must use FROM TAG '<server name> ARCHIVELOG TAG '<server name>' options.
- Re-instantiation of the standby database must utilize the TAG syntax. The steps to re-instantiate the standby database are as follows:
  1. Start the standby instance in NOMOUNT state using the same parameter files that the standby was operating on earlier. You can use the following SQL to get a current copy of the PFILE from SPFILE:

     ```
     CREATE PFILE='<filename>' FROM SPFILE;
     ```

  2. Create a standby control file on primary instance using the following SQL:

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

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

     ```
     RESTORE DATABASE FROM TAG '<server name used in
     BACKUP command>';

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

  5. Re-start the MRP.

Standby Database Configured as Archived Log Repository

A standby database can be configured as an archived log repository to serve as a remote backup for archived logs. The repository does not contain data files, and can be used by other standby databases to retrieve missing archived logs. For more

information on this configuration, refer to Metalink Note 434164.1 (Data Guard Archived Redo Log Repository Example).

The scripts provided in the section "Backup Procedures" are still valid for backing up archived log repositories. However, omit the RMAN commands that back up data files, since no data files are 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 configuration, 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 that will be used in standby database creation.

Using Recovery Manager with Oracle Data Guard in Oracle Database 10*g*
April 2009
Author: Anand Beldalker, Timothy Chien
Contributing Authors:Steven Wertheimer, Antonio Romero, Ashish Ray, Lawrence To, Douglas Utzig, Tammy Bednar, Joe Meeks, Larry Carpenter

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
oracle.com