Zero Data Loss Recovery Appliance
Administrator's Guide

Release 12.2
E88067-07
February 2019

Zero Data Loss Recovery Appliance Administrator's Guide, Release 12.2

E88067-07

Contributing Authors: Glenn Maxey, Lance Ashdown, Aishwarya Minocha

Primary Author: Terence Buencamino

Contributors: Andrew Babb, Donna Carver, Tim Chien, Sean Connolly, Donna Cooksey, Bill Fischer, Mahesh Girkar, Ray Guzman, Dah-Yoh Lim, Colin McGregor, Kant Patel, Chris Plakyda, Padmaja Potineni, Kathy Rich, Jony Safi, Toru Sasaki, Lawrence To, Randy Urbano, Steven Wertheimer

Contents

2 Recovery Appliance Architecture

8 Replicating Backups with Recovery Appliance

9 Implementing Additional High Availability Strategies

13 Recovery Appliance View Reference

List of Figures

List of Tables

Preface

Welcome to *Zero Data Loss Recovery Appliance Administrator's Guide*.

This preface contains the following topics:

Audience

This document is intended for a computer professional who will configure and administer Zero Data Loss Recovery Appliance, commonly known as Recovery Appliance.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents:

- *Zero Data Loss Recovery Appliance Owner's Guide*
- *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*
- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Secure Backup Administrator's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to Recovery Appliance

The cloud-scale Zero Data Loss Recovery Appliance, commonly known as Recovery Appliance, is an Engineered System designed to dramatically reduce data loss and backup overhead for all Oracle databases in the enterprise. Integrated with Recovery Manager (RMAN), the Recovery Appliance enables a centralized, incremental-forever backup strategy for large numbers of databases, using cloud-scale, fault-tolerant hardware and storage. The Recovery Appliance continuously validates backups for recoverability.

This chapter contains the following topics:

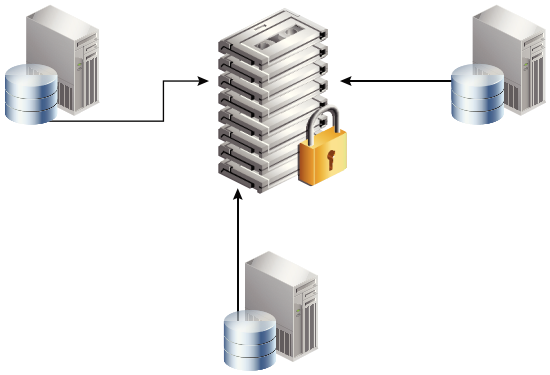
Traditional Database Backup Techniques

All production Oracle databases require data protection. Oracle provides RMAN as its preferred backup solution. Most enterprises have adopted one or more of the database backup strategies described in this section:

Weekly Full and Daily Incremental Backups

One popular approach, shown in Figure 1-1, is to use RMAN to take a weekly full backup, and then daily incremental backups. To improve incremental backup performance, Oracle recommends enabling block change tracking. These backups occur when activity on the database is lowest.

Figure 1-1 Full and Incremental Backups to Tape



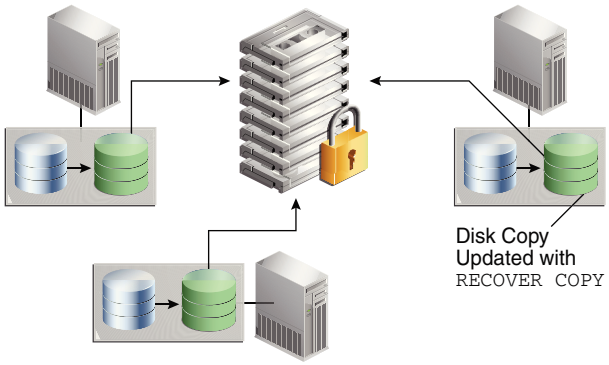
An advantage of this technique is that backup windows, which affect the production server, are relatively brief on the days when incremental backups occur. A disadvantage is that when the database is continuously active, as when serving multiple global time zones, no easily accommodating backup window is available.

One solution is to set up Oracle Data Guard, and then back up the standby database, thereby removing the backup load from the production server. However, protecting all databases with Oracle Data Guard is often impractical.

Incremental Backups and RECOVER COPY

The RMAN technique shown in Figure 1-2 makes daily incremental backups, and then uses the `RECOVER COPY` command to merge the incremental changes into the full database copy. In this way, the database copy on disk is "rolled forward" every day.

Figure 1-2 RECOVER COPY on Disk, and Backup to Tape



This technique has the following advantages:

- Only one initial full backup is required, which reduces the total weekly backup window time.
- An RMAN `SWITCH` command can point the control file to the database copy, which turns the copy into an actual database file, and thus eliminates the `RESTORE` step.

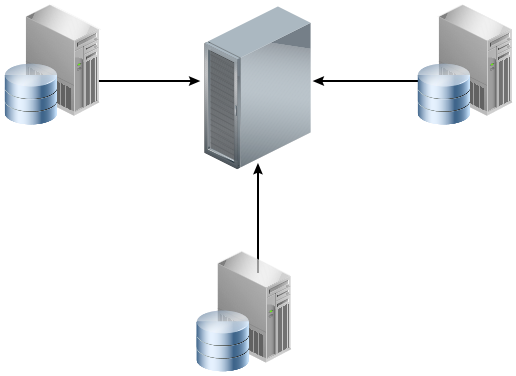
Some disadvantages are as follows:

- You must have sufficient disk space to keep a copy of the whole database on disk, and the archived redo log files required to recover it.
- Only one physical copy of the database exists. You select the point in time at which to keep the copy, so you can recover to subsequent points in time. For example, to restore to any point in time within the past week, your physical copy must be older than `SYSDATE-7`. The disadvantages are:
  - You cannot recover to a time earlier than the time at which you maintain the database copy.
  - The closer your recovery point in time is to the current time, the more incremental backups you must restore and apply to the copy. This technique adds time to the overall recovery time objective.
- The database copy cannot be compressed or encrypted.

Full Backups to a Third-Party Deduplicating Appliance

As an alternative to RMAN incremental backups and tape drives, some customers use third-party deduplicating appliances to process backup streams. Figure 1-3 depicts three databases writing to a centralized third-party appliance.

Figure 1-3 Third-Party Deduplicating Appliance



This technique has the following advantages:

- A central backup location serves all databases in the environment.
- The third-party software searches for patterns at the byte and sub-byte level to eliminate redundant data from backup to backup. For example, if a full database backup is almost identical to the backup taken a week before, then the software can attempt to prune the redundant bits from the incoming backup stream.
- To reduce network load, one optional technique utilizes source-side deduplication so that backup streams are deduplicated on the database host instead of the third-party appliance. Typically, this technique relies on an RMAN SBT plug-in.

Some disadvantages are as follows:

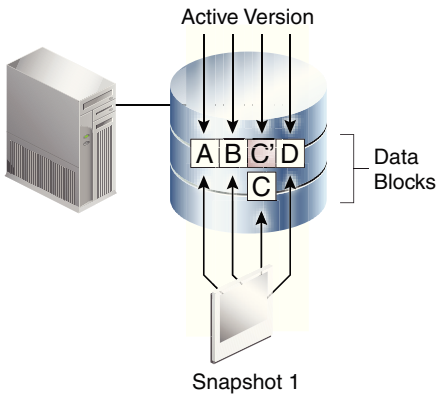
- These third-party appliances do not recognize or validate Oracle Database blocks. From the perspective of the appliance, a database backup is the same as a file system backup: a stream of bytes.

- Deduplication is only effective for full database backups that have a high degree of redundancy. Strategies that use incremental backups often do not achieve good deduplication ratios.
- The third-party appliance dictates which Oracle Database features to use rather than the other way around. Often, adapting to the requirements of the appliance means rewriting existing backup scripts.

Third-Party Storage Snapshots

A third-party storage snapshot is a set of pointers to storage blocks (*not* Oracle blocks) that existed when the snapshot was created. The virtual copies reside on the same storage array as the original data. Figure 1-4 depicts a copy-on-write snapshot, which is a type of third-party snapshot. After a snapshot is taken, when the first change to a storage block occurs, the array copies the before-image block to a new location on disk (`C`) and writes the new block (`C'`) to the original location.

Figure 1-4 Third-Party Copy-on-Write Snapshot



This technique has the following advantages:

- An initial copy of the database is not necessary because snapshots are not stored as physical copies of blocks. Thus, less storage is consumed than in RMAN strategies.
- Snapshots can be extremely fast. You put the database in backup mode (unless storage does *not* meet the requirements for snapshot storage optimization), and then take the snapshot. The snapshot needs to store physical blocks only when the blocks change, so a backup of an unchanged file is a metadata-only operation.
- Snapshots use storage efficiently. A backup of a file with a single changed block requires only one additional version of the block to be stored—either the old version or new version of the block, depending on the snapshot technique.

Some disadvantages are as follows:

- Snapshots have no knowledge of an Oracle Database block structure, and thus cannot validate Oracle blocks.
- Because snapshots reside on the same storage array as the source database, they are vulnerable to storage failures and data corruptions. If the array is inaccessible, or if the storage contains data block corruptions, then the snapshots cannot be used for recovery.

- Restoring a snapshot in place voids all snapshots that were taken after it unless the snapshot is fully restored to an alternate location.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* to learn more about using Storage Snapshot Optimization to take third-party snapshots of the database

Data Protection Challenges in the Modern Enterprise

The role of information technology in the modern business is going through a tremendous transformation. The key drivers for this transformation are:

- Data growth

  Many organizations continue to experience exponential growth, which creates a greater challenge for efficient data management and protection. What works well for dozens of databases may not work well for hundreds or thousands of databases, often running on different platforms and on multiple physical servers.

- Real-time analytics

  Organizations are increasingly dependent on data analysis for critical real-time decisions. This dependency increases the pressure to maintain data integrity and prevent data loss.

- Continuous global availability

  Many databases provide 24/7 access across multiple time zones, which means that databases are continuously active.

The protection strategies described in "Traditional Database Backup Techniques" are not designed to solve the challenges created by this transformation. Enterprises find themselves without a consistent backup and recovery strategy. The following shortcomings are common to most or all of the traditional backup techniques:

- Data loss exposure

  A database is only recoverable to its last valid backup, which may have occurred hours or days ago. In addition, storage snapshots and third-party appliances cannot validate Oracle data blocks, and so cannot detect Oracle block-level corruptions.

- Long backup windows

  As database sizes increase, the lengths of the backup windows also increase, creating additional load on production systems. Critical databases cannot afford to be deprived of resources used for daily backups and related maintenance activities.

- Lack of backup validation

  Because most third-party backup snapshot and Recovery Appliances lack Oracle integrated data block and database backup validation, restore and recovery operations tend to fail. Such failures result in extended downtime and potentially larger data loss.

- Lack of end-to-end visibility

  As the number of databases increases exponentially, so the ease of manageability decreases. Backup scripts proliferate and change. New DBAs may struggle to understand what the legacy scripts do. Questions about the status, backup location, and recovery point objective (RPO) of a particular database become harder to answer.

The traditional techniques fail to provide a comprehensive and efficient Oracle-integrated data protection solution that meets the demands of a large-scale, enterprise Oracle environment. A new approach is required.

Oracle's Recovery Appliance Solution

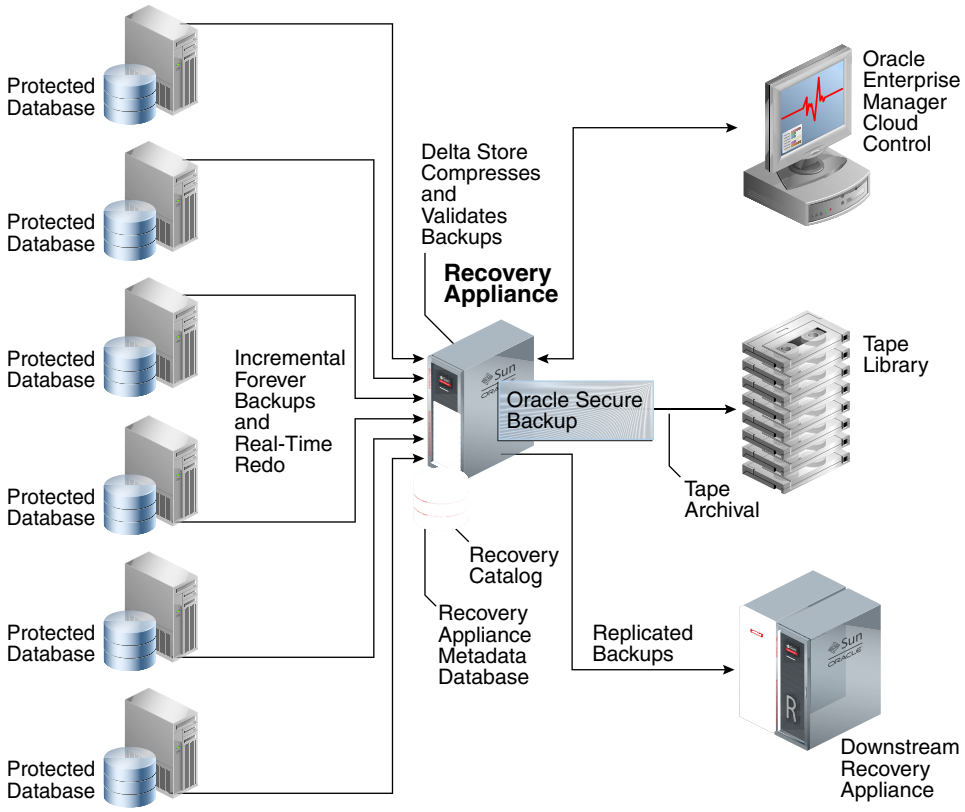
Recovery Appliance is a cloud-scale Engineered System designed to protect all Oracle databases across the enterprise. Most database backup and restore processing is performed by the centralized Recovery Appliance, making storage utilization, performance, and manageability of backups more efficient.

The Recovery Appliance stores and manages backups of multiple Oracle databases in a unified disk pool, using an RMAN incremental-forever strategy. The Recovery Appliance continually compresses, deduplicates, and validates backups at the database block level, while creating virtual full backups on demand.

A virtual full backup is a complete database image as of one distinct point in time, maintained efficiently through Recovery Appliance indexing of incremental backups from protected databases. A virtual full backup can correspond to any incremental backup that was received.

Figure 1-5 shows an overview of a sample Recovery Appliance environment.

Figure 1-5 Recovery Appliance Environment



As shown in Figure 1-5, a protected database is a client database that backs up data to a Recovery Appliance. Each protected database uses the Zero Data Loss Recovery Appliance Backup Module (Recovery Appliance Backup Module) for its backups. This module is an Oracle-supplied SBT library that RMAN uses to transfer backup data over the network to the Recovery Appliance.

The Recovery Appliance metadata database, which resides on each Recovery Appliance, manages metadata stored in the RMAN recovery catalog, and backups located in the Recovery Appliance storage location. The catalog is required to be used by all protected databases that send backups to Recovery Appliance.

> **Note:**
>
> Databases may use Recovery Appliance as their recovery catalog without also using it as a backup repository.

Administrators use Oracle Enterprise Manager Cloud Control (Cloud Control) to manage and monitor the environment. Cloud Control provides a "single pane of glass" view of the entire backup lifecycle for each database, whether backups reside on disk, tape, or another Recovery Appliance.

Recovery Appliance provides the following benefits:

- Elimination of Data Loss
- Minimal Backup Overhead
- Improved End-to-End Data Protection Visibility
- Cloud-Scale Protection

> **See Also:**
>
> "The Recovery Appliance Environment"

Elimination of Data Loss

The Recovery Appliance uses various mechanisms to protect against different types of data loss, including physical block corruption. This section contains the following topics:

- Protection of Ongoing Transactions
- Secure Replication
- Autonomous Tape Archival
- End-to-End Data Validation

Protection of Ongoing Transactions

In traditional backup approaches, if the online redo log is lost, then media recovery loses all changes after the most recent available archived redo log file or incremental backup. A recovery point objective (RPO) of a day or more that might result from a traditional approach may be unacceptable.

Recovery Appliance solves the RPO problem through a continuous transfer of redo changes to the appliance from a protected database. This operation is known as real-time redo transport. Using delta push, the Recovery Appliance is a remote destination for asynchronous redo transport services from Oracle Database 11*g* and Oracle Database 12*c* databases.

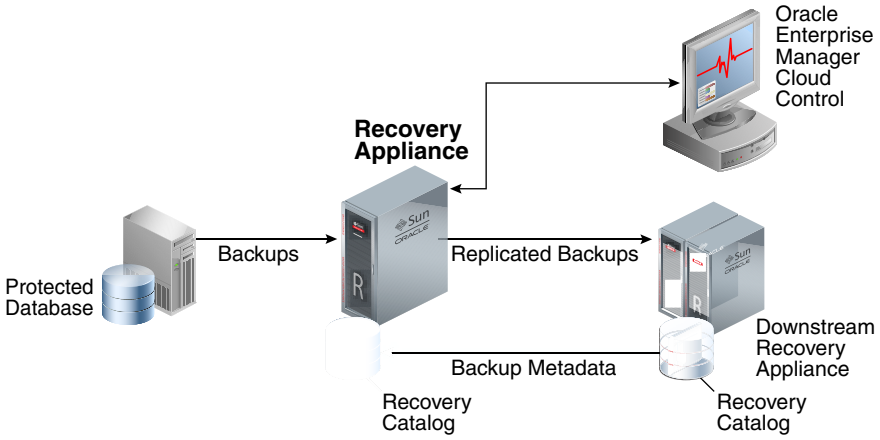
> **Note:**
>
> This technology is based on the real-time redo transport algorithms of Oracle Data Guard. To avoid degrading the performance of the protected database, protected databases transfer redo asynchronously to the Recovery Appliance. If a protected database is lost, zero to subsecond data loss is expected in most cases.

> **See Also:**
>
> - "Real-Time Redo Transport" to learn more about real-time redo transport
> - "Delta Push"
> - *Oracle Data Guard Concepts and Administration* for information about Oracle Data Guard redo transport

Secure Replication

To protect against server or site outage, one Recovery Appliance can replicate backups to a different Recovery Appliance. Figure 1-6 shows the simplest form of replication, called one-way Recovery Appliance replication, in which an upstream Recovery Appliance (backup sender) transfers backups to a downstream Recovery Appliance (backup receiver).

Figure 1-6 One-Way Replication



In Figure 1-6, a protected database sends an incremental backup to the Recovery Appliance, which then queues it for replicating to the downstream Recovery Appliance. When the upstream Recovery Appliance sends the incremental backup to the downstream Recovery Appliance, it creates a virtual full backup as normal. The downstream Recovery Appliance creates backup records in its recovery catalog. When the upstream Recovery Appliance requests the records, the downstream Recovery Appliance propagates the records back.

If the local Recovery Appliance cannot satisfy virtual full backup requests, then it automatically forwards them to the downstream Recovery Appliance, which sends virtual full backups to the protected database. DBAs use RMAN as normal, without needing to understand where or how the backup sets are stored.

> **See Also:**
>
> "Recovery Appliance Replication"

Autonomous Tape Archival

A robust backup strategy protects data against intentional attacks, unintentional user errors (such as file deletions), and software or hardware malfunctions. Tape libraries provide effective protection against these possibilities.

Figure 1-7 show the traditional technique for tape backups, with a media manager installed on each host.

Figure 1-7 Backups to Tape Without Using Recovery Appliance

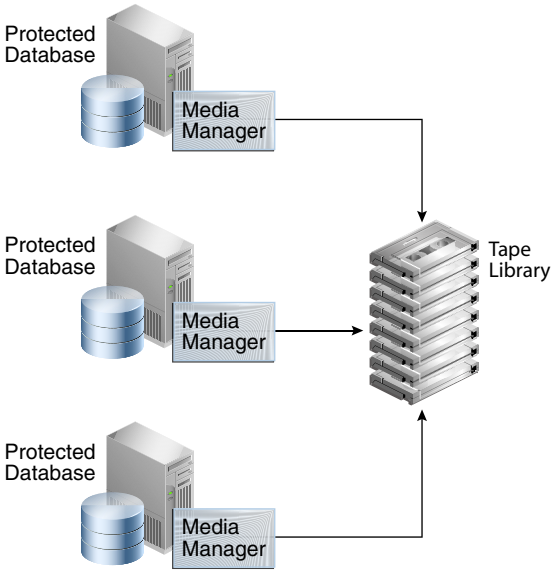
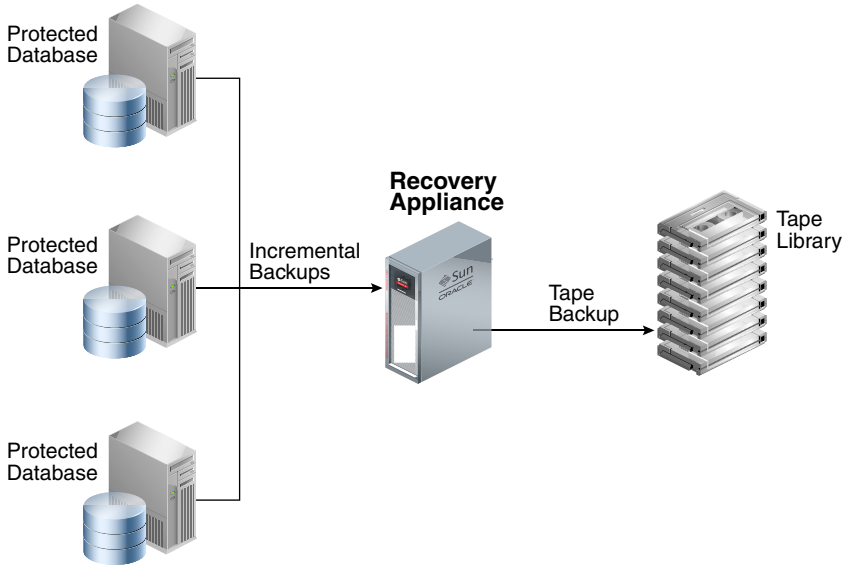


Figure 1-8 shows the Recovery Appliance technique for tape backups. The fundamental difference in the two approaches is that the Recovery Appliance backs up to tape, *not* the protected databases. The Recovery Appliance comes with preinstalled Oracle Secure Backup software, and supports optional Fibre Channel cards. Thus, installation of a media manager is not necessary on the protected database hosts.

Figure 1-8 Backups to Tape Using Recovery Appliance



When Recovery Appliance executes a copy-to-tape job for a virtual full backup, it constructs the physical backup sets, and copies them to tape, and then writes the metadata to the recovery catalog. If desired, the Recovery Appliance can also copy successive incremental backups and archived redo log file backups to tape. Whereas the backup on the Recovery Appliance is virtual, the backup on tape is a non-virtual, full physical backup. The Recovery Appliance automatically handles requests to restore backups from tape, with no need for administrator intervention.

The advantages of the Recovery Appliance tape solution are as follows:

- The Recovery Appliance performs all tape copy operations automatically, with no performance load on the protected database host.
- Tape backups are optimized. Recovery Appliance intelligently gathers the necessary blocks to create a non-virtual, full backup for tape.
- Oracle Secure Backup is preinstalled, eliminating the need for costly third-party media managers.

> **Note:**
>
> You may deploy tape backup agents from third-party vendors on the Recovery Appliance for integration with existing tape backup software and processes. In this configuration, the agents must connect to their specialized media servers, which must be deployed externally to the Recovery Appliance.

- Tape drives and tape libraries function more efficiently because Recovery Appliance is a single large centralized system with complete control over them. In other tape solutions, hundreds or thousands of databases can contend for tape resources in an uncoordinated manner.

See Also:

- Copying Backups to Tape with Recovery Appliance
- *Oracle Secure Backup Administrator's Guide*

End-to-End Data Validation

A basic principle of backup and recovery is to ensure that backups can be restored successfully. To ensure that there are no physical corruptions within the backed-up data blocks, backups require regular validation. Validation typically involves running an RMAN RESTORE VALIDATE job regularly, along with running periodic full restore and recovery operations to a separate machine.

Recovery Appliance provides end-to-end block validation, which occurs in the following stages of the workflow:

- Recovery Appliance validation

  The Recovery Appliance automatically validates the backup stream during the backup ingest phase, before writing the backups to disk. The Recovery Appliance also validates the backup before sending it back to the original or alternate database server during the restore phase. Therefore, no manual `RESTORE VALIDATE` step is required.

  In addition, a background task running on the Recovery Appliance periodically validates the integrity of the virtual full backups in the delta pools (see "Delta Pools"). The goal of this task is to check each block of each virtual full backup of each protected database and to work behind the scenes when minimal activity is occurring. By default, the validation task runs every 14 days following the last completed validation of a database's current set of backups on disk.

  Just as with data file backups, the Recovery Appliance validates the integrity of redo log blocks during every operation, including receiving redo from the protected database, and storing it in compressed archived log backup sets.

- Oracle Automatic Storage Management (ASM)

  Oracle ASM stores the backup and redo data for the Recovery Appliance. Oracle ASM mirrored copies provide redundancy (see "Recovery Appliance Storage Locations").

  If a corrupted block is read on the primary mirror, the Recovery Appliance automatically repairs the block from the mirrored copy. This mechanism resolves most isolated block corruption cases.

- Tape library

  Recovery Appliance validates blocks when it copies them to tape, and also when it restores them from tape (see "Tape Archival").

- Downstream Recovery Appliance in a replication configuration

  If you configure replication, then the downstream Recovery Appliance validates data during the backup ingest and restore phases (see "How a Downstream Recovery Appliance Processes Backups").

None of the preceding backup validation processes occur on the production database hosts, thus freeing production resources for more critical operational workloads.

> **Note:**
>
> Oracle Maximum Availability Architecture best practices recommend that you still perform periodic full database recovery tests to verify operational practices and to detect issues that might occur only during media recovery.

> **See Also:**
>
> - "CONFIG" for information about the `validate_db_days` configuration parameter
> - "RA_DATABASE" for information about the `RA_DATABASE.LAST_VALIDATE` column

Minimal Backup Overhead

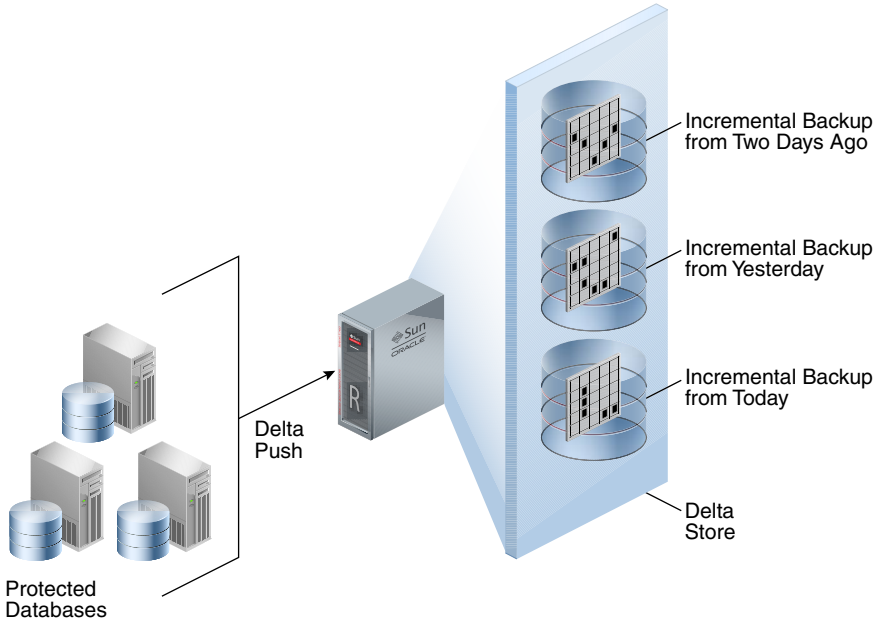
In traditional database backup techniques, the Oracle database host performs the brunt of the processing. Agents for disk backup, tape backup, and deduplication may all be running on the host. Furthermore, all backup operations—compression, validation, deletion, merging, and so on—occur on the database host. This overhead can greatly degrade database performance.

Recovery Appliance removes almost the entire load from the protected databases. The only backup operations required on the hosts, which could be primary database or standby database hosts, are sending incremental backups to the Recovery Appliance. The incremental-forever strategy reduces the backup window on the database hosts significantly. Recovery Appliance handles backup processing, tape operations, data integrity checks, and routine maintenance.

> **Note:**
>
> Recovery Appliance only supports backups of Oracle databases, not file system data or non-Oracle databases.

Recovery Appliance optimizes management of database changes using delta push and delta store, shown in Figure 1-9. The net result of delta push and delta store is that the problem of lengthening backup windows is eliminated. The DBA performs only fast incremental backups, and lets the Recovery Appliance manage the backup blocks.

Figure 1-9 Delta Push and Delta Store



> **See Also:**
>
> "Traditional Database Backup Techniques"

Delta Push

This solution consists of two operations that run on each protected database: the incremental-forever backup strategy, and real-time redo transport. Both operations involve protected databases pushing changes to the Recovery Appliance.

In an incremental-forever strategy, only one incremental level 0 backup to the Recovery Appliance is required in the lifetime of each protected file. The initial level 0 backup does not contain committed undo blocks or currently unused blocks.

> **Note:**
>
> The elimination of committed undo and currently unused blocks is only supported for SBT full backups to the Recovery Appliance or Oracle Secure Backup. It is not available for SBT backups to other backup products.

In normal operation, the Recovery Appliance automatically performs the following steps for each incremental level 1 backup:

1. Receives a scheduled incremental level 1 backup from each protected database
2. Validates the incoming backup to protect against physical block corruptions

3. Compresses the backup using specialized block-level algorithms
4. Writes the backup to a delta store in a Recovery Appliance storage location

The incremental-forever strategy greatly reduces the backup window and overhead because no full backups are ever required after the initial incremental level 0 backup. If the strategy includes real-time redo transport, then backup windows are further reduced because traditional archived log backups are not necessary. Also, Recovery Appliance takes on the burden of validation, deduplication, and compression.

> **Note:**
>
> Blocks compressed using table or Hybrid Columnar Compression remain compressed in the RMAN backup and during the Recovery Appliance ingest phase.

> **See Also:**
>
> "Elimination of Data Loss"

Delta Store

The delta store is the key processing engine for Recovery Appliance. A protected database sends only one incremental level 0 backup of each data file to the Recovery Appliance. Following the initial full backup, all backups are highly efficient cumulative incremental backups.

As Recovery Appliance receives incremental backups, it indexes them and stores them in delta pools. Each separate data file backed up to the Recovery Appliance has its own separate delta pool (set of backup blocks). Recovery Appliance automatically manages the delta pools so that it can provide many virtual full backups.

Creation of Virtual Full Backups

To create a virtual full backup, Recovery Appliance converts an incoming incremental level 1 backup into a virtual representation of an incremental level 0 backup. A virtual full backup appears as an incremental level 0 backup in the recovery catalog. From the user's perspective, a virtual full backup is indistinguishable from a non-virtual full backup. Using virtual backups, Recovery Appliance provides the protection of frequent level 0 backups with only the cost of frequent level 1 backups.

> **Note:**
>
> Recovery Appliance provides storage services, but not virtual full backups, for RMAN-encrypted backups (see "Archival and Encrypted Backups"). These backups are stored in their original encrypted format. Recovery Appliance can store, archive, and retrieve them just as it can for unencrypted RMAN backup sets.

Rapid Recovery Using Virtual Full Backups

Recovery Appliance uses virtual full backups to provide rapid recovery to any point in time, regardless of the amount of data being recovered. The on-disk recovery strategy of Recovery Appliance has the advantage that RMAN can recover virtual full backups to any point in time *without* applying incremental backups.

When a database is protected by the Recovery Appliance, RMAN must only restore a single level 0 backup for the day of the RPO, and then recover up to the last second using redo log files sent using the real-time redo transport feature. For example, if the recovery window is 7 days, and if the RPO is 5 days ago, then RMAN can restore a single virtual full (level 0) backup that is current to 5 days ago, and then recover it using redo—not level 1 incremental backups.

See Also:

- "Delta Store"
- *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn more about the incremental-forever backup strategy
- *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn more about recovery strategies
- *Oracle Database Backup and Recovery User's Guide* to learn more about incremental backups

Improved End-to-End Data Protection Visibility

In traditional database backup techniques, management of the database, media server, and tape drives are often separated. For example, a DBA group may manage the databases, while a separate backup administrator group manages the backups, and a storage group manages the disk and tape devices. The overall process lacks visibility, which makes it difficult to manage backups for thousands of databases, each with different recovery requirements.

Cloud Control provides a complete, end-to-end view into the backup lifecycle managed by the Recovery Appliance, from the time the RMAN backup is initiated on the database, to when it is stored on disk, tape, or replicated to a downstream Recovery Appliance. Recovery Appliance monitoring and administration are enabled through installation of the Enterprise Manager for Zero Data Loss Recovery Appliance plug-in (Recovery Appliance plug-in).

Using Cloud Control to manage a Recovery Appliance provides the following benefits:

- Standard metrics such as overall backup performance, and aggregate or per-database space consumption
- Immediate alerts about any backup or Recovery Appliance issues

  For example, Cloud Control may alert the administrator if no backup is available to meet the defined RPO, or if corrupt backups are discovered.
- Status reports, enabled by BI Publisher, are useful for capacity planning and to identify protected databases that are not meeting recovery window goals

For example, Recovery Appliance administrators can receive reports on historical space and network usage to identify backup volume and throughput trends. These trends may necessitate adding storage servers to an existing rack or connecting additional racks.

Although Cloud Control is the recommended user interface for Recovery Appliance administration, Oracle supplies the `DBMS_RA` PL/SQL package as a command-line alternative. Most tasks in this manual provide both Cloud Control and `DBMS_RA` techniques. For command-line monitoring and reporting, you can query the Recovery Appliance catalog views.

See Also:

- "Traditional Database Backup Techniques"
- "Getting Started with Cloud Control for Recovery Appliance "
- "DBMS_RA Package Reference"
- "Recovery Appliance View Reference"

Cloud-Scale Protection

Recovery Appliance scales at a cloud level, supporting tens to hundreds to thousands of databases across a data center. Essentially, Recovery Appliance enables you to create a private data protection cloud within the enterprise. The following technology components within Recovery Appliance make this possible:

- Policy-Based Data Protection Management
- Database-Aware Space Management
- Scalable Architecture

Policy-Based Data Protection Management

Recovery Appliance simplifies management through the protection policy. Benefits include the following:

- A protection policy defines recovery window goals that are enforced for each database for backups to the Recovery Appliance or a tape device.

  Using protection policies, you can group databases by recovery service tier. For example, databases protected by the Platinum policy require backups to be kept for 45 days on the Recovery Appliance and 90 days on tape, which means that backups aged 45 days or less exist on disk *and* tape, but backups older than 45 days are only on tape. Databases protected by the Gold policy require 35 days on the local Recovery Appliance and 90 days on tape. Optionally, you can define a maximum retention time within each policy to limit the space consumed, and to comply with service level agreements dictating that backups cannot be maintained for longer than a specified period.
- Protection policies are means of grouping databases, improving manageability.

  For example, you can configure Recovery Appliance replication or copy-to-tape for a specific protection policy, which means that the configuration applies to all

databases associated with this policy. If you add a database to the policy, then the database automatically inherits the configurations and scheduling of the policy.

> **See Also:**
>
> - "Protection Policies"
> - Managing Protection Policies with Recovery Appliance

Database-Aware Space Management

Using protection policies, the Recovery Appliance manages backup storage space according to the recovery window goal for each protected database. This granular, database-oriented space management approach eliminates the need to manage space at the storage-volume level, as third-party appliances do.

If space is available, then the Recovery Appliance may retain backups older than the recovery window goal, effectively extending the point-in-time recovery period. When space pressure exists, the Recovery Appliance uses predefined thresholds to purge backups. The Recovery Appliance automatically provisions space so that the recovery window goal for each database is met.

> **See Also:**
>
> "How Recovery Appliance Manages Storage Space"

Scalable Architecture

The approaches in "Traditional Database Backup Techniques" are prone to performance bottlenecks and multiplying points of failure. As the number of databases increases, so does the number of media servers, disk arrays, tape devices, and third-party appliances, and thus so does the overall complexity. The "add more devices" approach is not scalable. In contrast, Recovery Appliance can scale to accommodate increases in backup traffic, storage usage, and the number of databases by adding compute and storage resources in a simple, modular fashion.

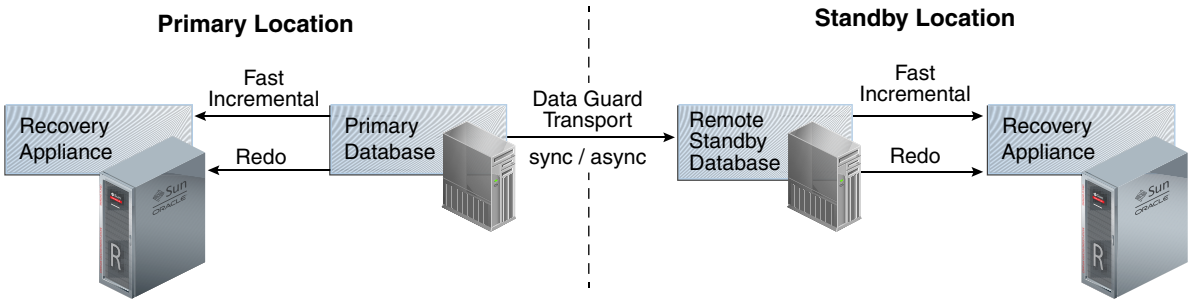
> **See Also:**
>
> *Zero Data Loss Recovery Appliance Owner's Guide* for information about adding storage servers

Maximum Availability: Recovery Appliance with Oracle Data Guard

Oracle Data Guard is a component of a high availability (HA) and disaster recovery solution that can be integrated with Recovery Appliance to provide maximum data protection. Oracle Data Guard minimizes service interruption and resulting data loss

by maintaining a synchronized standby database for the protected database. When the primary system is unavailable, the standby immediately assumes the normal operations of the primary after a Data Guard failover operation, including backups to the local Recovery Appliance. Figure 1-10 shows an example of an environment with Recovery Appliance and Oracle Data Guard.

Figure 1-10 Recovery Appliance with Oracle Data Guard



In Figure 1-10, the primary and standby databases each send incremental backups to their local Recovery Appliance. The primary database sends real-time redo changes to both the local Recovery Appliance and the physical standby, and the standby cascades the redo changes to the remote Recovery Appliance. Each Recovery Appliance has backups and redo information for the same database, therefore either appliance can be used for RMAN restore and recovery operations.

> **See Also:**
>
> - `http://www.oracle.com/technetwork/database/availability/disaster-recovery-2526839.pdf` to learn more about Recovery Appliance with Oracle Data Guard
> - *Oracle Data Guard Concepts and Administration* for information about Oracle Data Guard

What's Next?

To begin using Recovery Appliance, refer to the following topics:

1. Optionally, read Recovery Appliance Architecture to obtain a more in-depth understanding of the principal components of the Recovery Appliance environment.
2. Read Recovery Appliance Workflow to learn about basic tools and tasks. Before you can use Recovery Appliance for data protection, you must perform the tasks described in the following topics:
   a. "Planning for Recovery Appliance"
   b. "Setup and Configuration for Recovery Appliance"

c. "Maintenance Tasks for Recovery Appliance"

2

Recovery Appliance Architecture

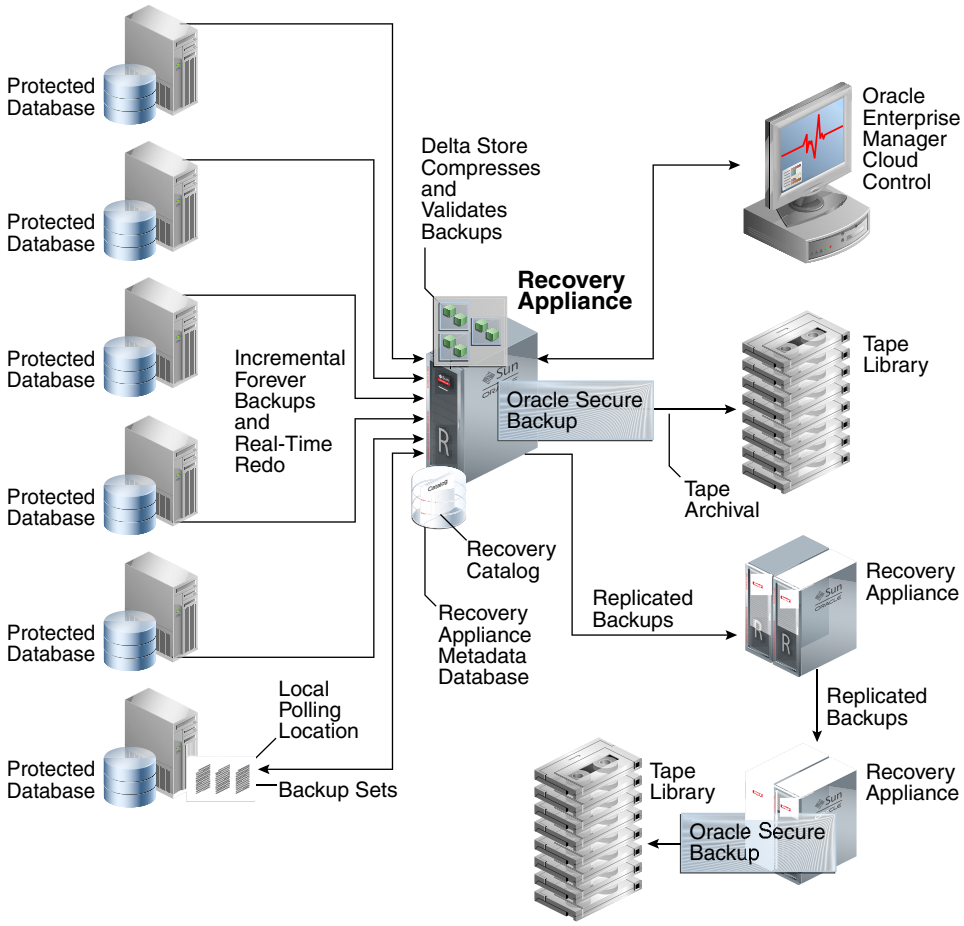
This chapter describes the basic architecture and concepts for Zero Data Loss Recovery Appliance, commonly known as Recovery Appliance.

This chapter contains the following topics:

The Recovery Appliance Environment

At minimum, the Recovery Appliance environment consists of one Recovery Appliance and one protected database. More typical is the sample environment shown in Figure 2-1.

Figure 2-1 Sample Recovery Appliance Environment



This section contains the following topics:

- Main Components of the Recovery Appliance Environment
- User Accounts in the Recovery Appliance Environment
- Lifecycle of a Backup: Scenario

Main Components of the Recovery Appliance Environment

Figure 2-1 shows an example of a typical Recovery Appliance environment, which contains the following components:

- Multiple protected databases

  Each protected database sends backups and real-time redo to the Recovery Appliance. Protected databases can run on different releases of Oracle Database. For example, a mixed environment might include protected databases from Oracle Database 10*g*, Oracle Database 11*g*, and Oracle Database 12*c*.

- Recovery Appliance

  Figure 2-1 shows a central Recovery Appliance, which receives incremental backups and real-time redo from the protected databases. The Recovery

Appliance contains the Recovery Appliance metadata database. This database includes the following components:

  - The RMAN recovery catalog, which is subdivided into multiple virtual recovery catalogs.
  - One or more storage locations. Recovery Appliance storage contains the delta store, which includes multiple delta pools.

  Figure 2-1 also shows the central Recovery Appliance replicating backups to a second Recovery Appliance, which in turn forwards these backups to a third Recovery Appliance.

- Oracle Enterprise Manager Cloud Control (Cloud Control)

  Figure 2-1 shows Cloud Control running on a separate server in the environment. Administrators can use Cloud Control to manage all Recovery Appliances, protected databases, and tape devices in the Recovery Appliance environment.

- `DBMS_RA` PL/SQL package

  This is the command-line interface to Recovery Appliance. This package, which is stored in the Recovery Appliance metadata database, provides the underlying functionality for Cloud Control.

- Oracle Secure Backup

  Figure 2-1 shows the Recovery Appliance using Oracle Secure Backup to archive backups to a tape library. The diagram also shows a downstream Recovery Appliance archiving backups to a separate tape library.

See Also:

- "Recovery Appliance Metadata Database"
- "Recovery Appliance Storage"
- "Recovery Appliance Replication"
- "DBMS_RA Package Reference"

User Accounts in the Recovery Appliance Environment

The central components of a Recovery Appliance environment are the protected databases, Recovery Appliance, and Cloud Control. Table 2-1 summarizes the most important user accounts in the environment.

Table 2-1 User Accounts in the Recovery Appliance Environment

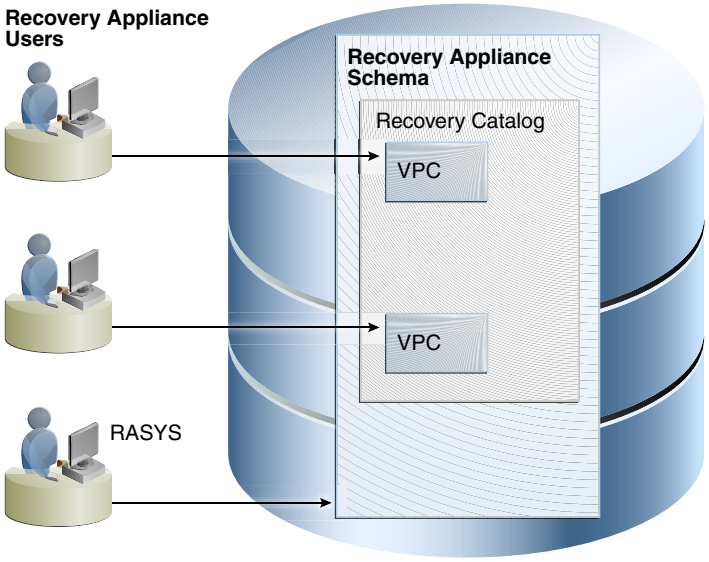
Component	Account Type	User Name	Description
Cloud Control	Cloud Control super-user	`SYSMAN`	This application account exists by default. Its purpose is to administer Cloud Control itself, and is not directly related to managing a Recovery Appliance or protected databases.

Table 2-1 (Cont.) User Accounts in the Recovery Appliance Environment

Component	Account Type	User Name	Description
Cloud Control	Cloud Control administrator	User-Specified	A Cloud Control user account that has been granted the roles and privileges needed to manage a specific protected database or a specific Recovery Appliance. Multiple Cloud Control administrative accounts may exist, depending on your business requirements.
Recovery Appliance	Recovery Appliance metadata database super-user	SYS	SYS can create Recovery Appliance user accounts, but typically is not otherwise used for managing Recovery Appliance.
Recovery Appliance	Recovery Appliance administrator	RASYS	This database account owns the Recovery Appliance schema, which includes the RMAN recovery catalog and the DBMS_RA PL/SQL package (see DBMS_RA Package Reference). The RASYS user name is fixed and cannot be changed. RASYS does not have the privileges required to create database user accounts.
Recovery Appliance	Recovery Appliance user account	User-Specified	This account has authority to send and receive backups for databases registered with the Recovery Appliance, and to manipulate recovery catalog metadata for these databases. This is also the account to use to send redo data from a protected database to the Recovery Appliance. Unlike RASYS, a Recovery Appliance user account has no administrative capabilities in the Recovery Appliance. Typically, a Recovery Appliance metadata database contains multiple Recovery Appliance user accounts. These accounts are created when configuring access for protected databases (see Configuring Recovery Appliance for Protected Database Access). Every Recovery Appliance user account owns a virtual private catalog. The catalog owner can access and modify only those rows in the recovery catalog that pertain to the databases to which it has been granted access. The catalog user name for this is referenced in an RMAN CONNECT CATALOG command.
Protected Database	Protected database backup administrator	User account with SYSBACKUP privileges (or SYSDBA for releases in which SYSBACKUP is not supported)	This account has the privileges to back up, restore, and recover a protected database. This is the database user name that is referenced in an RMAN CONNECT TARGET command.

Figure 2-2 depicts the relationship between RASYS and two Recovery Appliance user accounts. In this example, each Recovery Appliance user account owns a separate virtual private catalog. Note that RASYS, as owner of the Recovery Appliance schema, is also the owner of the RMAN recovery catalog.

Figure 2-2 RASYS and Recovery Appliance User Accounts



> **See Also:**
>
> *Oracle Database Security Guide* to learn how to create database user accounts

Lifecycle of a Backup: Scenario

This section describes the lifecycle of a backup as it flows through the Recovery Appliance environment depicted in Figure 2-1. In this sample scenario, each protected database has already seeded Recovery Appliance with the required initial level 0 incremental backup. The basic data flow is as follows:

1. A protected database, or a standby database protecting this database, sends a level 1 incremental backup to the Recovery Appliance.

   Recovery Appliance distinguishes itself from other backup solutions because only one level 0 backup is ever required for each data file. Level 1 incremental backups are most efficient because data blocks are only backed up when they change.

   Oracle recommends making *cumulative* level 1 incremental backups (see *Oracle Database Backup and Recovery Reference*). Each cumulative level 1 backup uses the most recent *virtual* level 0 backup as its baseline. Typically, this virtual level 0 backup corresponds to the most recent level 1 backup.

> **Note:**
>
> If a level 1 cumulative backup cannot be incorporated into the Recovery Appliance (for example, because of a storage corruption), then the next level 1 backup has the same virtual level 0 backup baseline, enabling the Recovery Appliance to seamlessly incorporate the new level 1 incremental backup. Thus, cumulative backups almost never have greater overhead than differential backups.

2. The Recovery Appliance receives the incremental backup.

   The received backup is available for immediate retrieval, but the Recovery Appliance has not yet indexed it, so the corresponding virtual full backups are not available. If a protected database requires this backup for recovery before the Recovery Appliance can index it, then RMAN automatically restores the previous virtual full backup and applies this incremental backup to it.

3. The Recovery Appliance processes the incremental backup.

   The following operations occur asynchronously:

   - The Recovery Appliance performs backup ingest. The Recovery Appliance processes the backup as follows:
     - Scans the backup that was sent by a protected database
     - Breaks it into smaller groups of blocks, assigning the blocks from each data file to a separate delta pool
     - Writes the groups into the appropriate storage location according to the protection policy for the database
     - Deletes the original backup set after the virtual backup set has been created

     > **Note:**
     >
     > The Recovery Appliance may not delete the original backup at precisely the same time that the virtual backup is created. Thus, it is possible for both the original and virtual backups to coexist briefly in the recovery catalog as two separate copies.

     During backup ingest, the Recovery Appliance also indexes the backup, which involves storing information about the contents and physical location of each data block in the metadata database. Because the Recovery Appliance contains the recovery catalog for the protected database, the newly indexed virtual full backups are now available for use by RMAN, if needed for recovery.

   - If Recovery Appliance replication is configured, then the Recovery Appliance forwards the backup to a downstream Recovery Appliance.

     Many different replication configurations are possible. Figure 2-1 shows a one-to-one configuration in which the central Recovery Appliance, acting as the upstream Recovery Appliance (backup sender), forwards its backups to a separate Recovery Appliance, acting as the downstream Recovery Appliance (backup receiver). Figure 2-1 shows cascaded replication, in which the

downstream Recovery Appliance forwards its backups to a third Recovery Appliance.

- If automated copy-to-tape policies are enabled, then the Recovery Appliance archives the backup to tape.

  In Figure 2-1, the central Recovery Appliance uses Oracle Secure Backup software to communicate with a tape device. Also, the Recovery Appliance furthest downstream in the replication scheme archives its backups to tape. This technique has the following benefits:

  – To create redundancy, identical backups reside on two separate tape devices. In Figure 2-1, the primary Recovery Appliance archives to tape, as does the Recovery Appliance that is furthest downstream.

  – A downstream Recovery Appliance can back up to tape, thus offloading tape archival processing from the upstream Recovery Appliance.

- The Recovery Appliance periodically verifies that backups and redo are valid.

  The Recovery Appliance automatically validates backups on disk, and during inbound and outbound replication. The Recovery Appliance automatically performs crosschecks of tape backups. Just as with data file backups, the Recovery Appliance validates the integrity of redo log blocks during every operation, including receiving redo from protected databases and storing it in compressed archived log backup sets. No manually run RMAN `VALIDATE` commands are required.

- The Recovery Appliance performs automated delta pool space management.

  This phase involves deleting obsolete and expired backups, both on disk and tape, and optimizing the delta pools.

See Also:

- "Recovery Appliance Storage Locations"
- "Replicating Backups with Recovery Appliance "
- "Automated Delta Pool Space Management"
- *Oracle Database Backup and Recovery User's Guide* to learn how to make incremental backups

Protected Databases

A protected database uses a specific Recovery Appliance as a destination for centralized RMAN backup and recovery. In Figure 2-1, multiple protected databases send backups to a single centralized Recovery Appliance. Each database protected by a Recovery Appliance must use the recovery catalog in the Recovery Appliance metadata database.

To send backups to a Recovery Appliance, a protected database must be configured to allow access to the Recovery Appliance. The configuration involves creating the appropriate Recovery Appliance users and permissions, associating each protected database with a protection policy, and distributing Recovery Appliance connection credentials to each database.

This section contains the following topics:

Recovery Appliance Backup Modules

The Zero Data Loss Recovery Appliance Backup Module (Recovery Appliance Backup Module) is an Oracle-supplied SBT library that RMAN uses to transfer backup data over the network to the Recovery Appliance. An SBT library transfers data to and from a backup device type, either a tape device or Recovery Appliance. RMAN performs all backups to the Recovery Appliance, and all restores of complete backup sets, by means of this module.

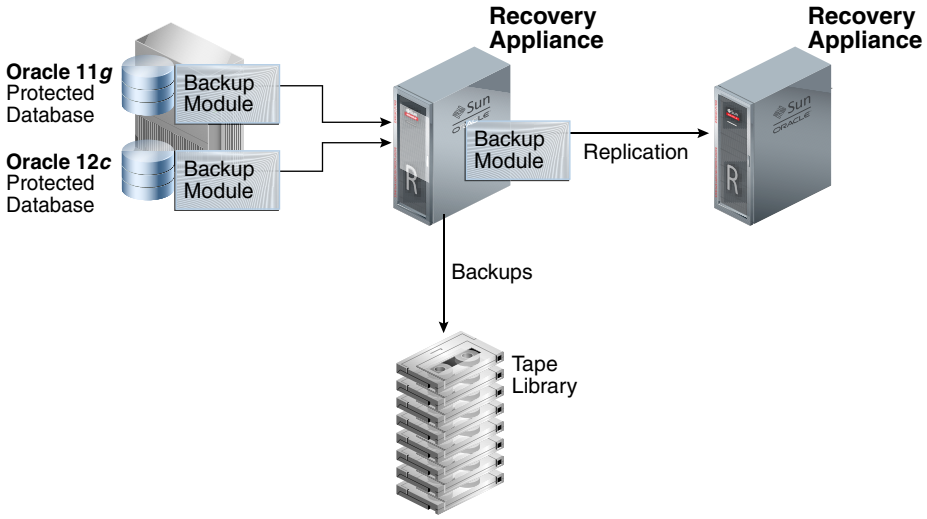
The Recovery Appliance Backup Module must be installed in the following locations:

- In the Oracle home of every protected database that sends backups to a Recovery Appliance

  For example, a single host might have an Oracle Database 11*g* Oracle home, and an Oracle Database 12*c* Oracle home. Each Oracle home might support five protected databases, for a total of ten databases running on the host. In this case, only two Recovery Appliance Backup Modules must be installed: one in each Oracle home.
- For Recovery Appliance replication environments, on every upstream Recovery Appliance that sends backups to downstream Recovery Appliances (see Replicating Backups with Recovery Appliance)

Figure 2-3 depicts an Oracle Database 11*g* and Oracle Database 12*c* protected database running on the same host. The Recovery Appliance Backup Module installed in each Oracle home communicates with the Recovery Appliance, replicates backups to a downstream Recovery Appliance.

Figure 2-3 Recovery Appliance Backup Modules



> **See Also:**
>
> - *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to install the Recovery Appliance Backup Module
> - *Oracle Database Backup and Recovery User's Guide* to learn more about SBT channels and devices

Protection Policies

A protection policy is a named collection of properties that you can assign to multiple protected databases. Using a single policy for multiple databases reduces Recovery Appliance administration time, and enables you to change the properties of multiple protected databases with one operation. To accommodate databases with differing backup and recovery requirements, create as many protection policies as required.

A default installation of Recovery Appliance has the protection policies shown in Table 2-2.

Table 2-2 Default Protection Policies

Service Tier	Recovery Window	Additional Settings
Platinum	45 days on disk, 90 days on tape[1]	Database backups, real-time redo transport, replication, and tape backups. All settings are mandatory.
Gold	35 days on disk, 90 days on tape	Database backups, real-time redo transport, replication, and tape backups (if tape is available).
Silver	10 days on disk, 45 days on tape	Database backups, real-time redo transport, and tape backups (if tape is available).
Bronze	3 days on disk, 30 days on tape	Database backups, and tape backups (if tape is available). There is *no* real-time redo transport.

[1] Backups aged 45 days or less exist on both disk and tape, but backups aged more than 45 days exist only on tape. The Recovery Appliance creates tape backups immediately after disk backups, so the 90 day tape retention period begins at the same time as the 45 day disk retention period.

> **See Also:**
>
> - *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to configure real-time redo transport
> - `http://www.oracle.com/technetwork/database/availability/maa-reference-architectures-2244929.pdf` to learn more about Oracle Maximum Availability Architecture (MAA) service tiers

Protection Policy Attributes

A protection policy, which you create with the `DBMS_RA.CREATE_PROTECTION_POLICY` procedure or with Cloud Control, sets the following attributes for all protected databases assigned to it:

Table 2-3 Protection Policy Attributes

Attribute	Description
`storage_location_name`	A Recovery Appliance storage location for storing backups
`polling_policy_name`	An optional backup polling policy that determines whether Recovery Appliance polls a storage location for backups
`recovery_window_goal`	The disk recovery window goal for the protected database.
`recovery_window_sbt`	The SBT retention period for the protected database
`guaranteed_copy`	The guaranteed copy setting, which determines whether backups protected by this policy must be copied to tape or replicated before being considered for deletion
`max_retention_policy`	The maximum length of time that the Recovery Appliance retains backups for databases that use this retention policy
`unprotected_window`	The maximum acceptable difference between the current time and the latest time that the database can be restored

You can associate an optional replication server configuration with a protection policy. The replication configuration applies to all protected databases associated with the protection policy.

> **See Also:**
>
> - "Creating a Protection Policy Using DBMS_RA"
> - "Protection Policies for Replication"
> - "CREATE_PROTECTION_POLICY"

Recovery Windows

When creating a protection policy, you can define the following two recovery window attributes, expressed as intervals (typically days):

- Disk recovery window goal

  For each database assigned to the policy, Recovery Appliance attempts to support a point-in-time recovery to any time within this interval, counting backward from the current time. For example, if the recovery window goal is 15 days, and if it is noon on April 25, then the goal is the ability to perform point-in-time recovery to any time on or after noon on April 10. At noon on April 26, the goal is the ability to perform point-in-time recovery to any time on or after noon on April 11, and so on.

  For disk, this interval is a goal, and not a guarantee. The Recovery Appliance might purge backups when disk space is low, in which case the goal is not always

met. You can ensure that a minimum number of backups are guaranteed to be available by adjusting the reserved disk space property of each protected database.

- SBT retention period

  For each assigned database, backups are retained long enough on tape to support a point-in-time recovery to any time within this interval, counting backward from the current time. For SBT, this interval is a guarantee.

> **See Also:**
>
> - "Recovery Window Goal"
> - ""Backup Retention on Tape""
> - *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*
> - *Oracle Database Backup and Recovery User's Guide* for a thorough discussion of recovery windows

Backup Polling Policies

A backup polling policy specifies:

- A file system directory on shared storage where Recovery Appliance polls for backups to process (see "Backup Polling Locations")
- The frequency with which Recovery Appliance polls
- Whether backup data is to be deleted after being successfully processed

Assign backup polling policies to protected databases through protection policies. Each protection policy can optionally reference a polling policy.

> **See Also:**
>
> "Creating a Backup Polling Policy (Command-Line Only)"

Supported Oracle Database Releases

See My Oracle Support Note Doc ID 1995866.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=1995866.1`) for information about the Oracle database releases supported by Recovery Appliance, including the features available with each release.

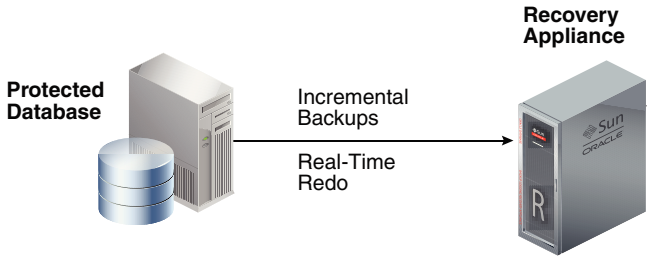
Real-Time Redo Transport

Redo data contains records of all changes made to a database and is therefore critical to minimizing data loss if data failure occurs. By using the real-time redo transport feature of Recovery Appliance, you substantially reduce the window of potential data

loss that exists between successive archived redo log backups. Typical RPO is zero to subsecond when you enable real-time redo transport.

Figure 2-4 shows a protected database sending incremental backups and redo logs to the Recovery Appliance.

Figure 2-4 Redo Log Transmission



With real-time redo transport enabled, a protected database generates redo changes in memory, and then immediately transfers them to the Recovery Appliance, which validates them and writes them to a staging area.

When the protected database performs an online redo log switch, the Recovery Appliance converts and assembles the redo changes into compressed archived redo log file backups. The Recovery Appliance catalog automatically tracks these archived redo log backups in its recovery catalog. RMAN can restore and apply these archived redo log backups as usual. The advantages are:

- If the redo stream terminates unexpectedly, then the Recovery Appliance can close the incoming redo stream and create a partial archived redo log file backup, thereby protecting transactions up to the last change that the appliance received. When the Recovery Appliance detects that the redo stream has restarted, it automatically retrieves all missing archived redo log files from the protected database. In this way, the Recovery Appliance can preserve the recovery window goal.
- Because the Recovery Appliance automatically converts real-time redo into archived redo log files, it is not necessary to back up archived redo log files from the database host to the Recovery Appliance.

The Recovery Appliance does not *apply* the redo that it receives to the backups sent by the protected databases. Thus, to continue providing updated virtual level 0 backups, the Recovery Appliance must incorporate new incremental backups into the delta store. The appliance provides a virtual level 0 backup corresponding to each level 1 incremental backup sent by the protected database. In a recovery scenario, you restore the appropriate level 0 backup, and then use redo log files to roll it forward.

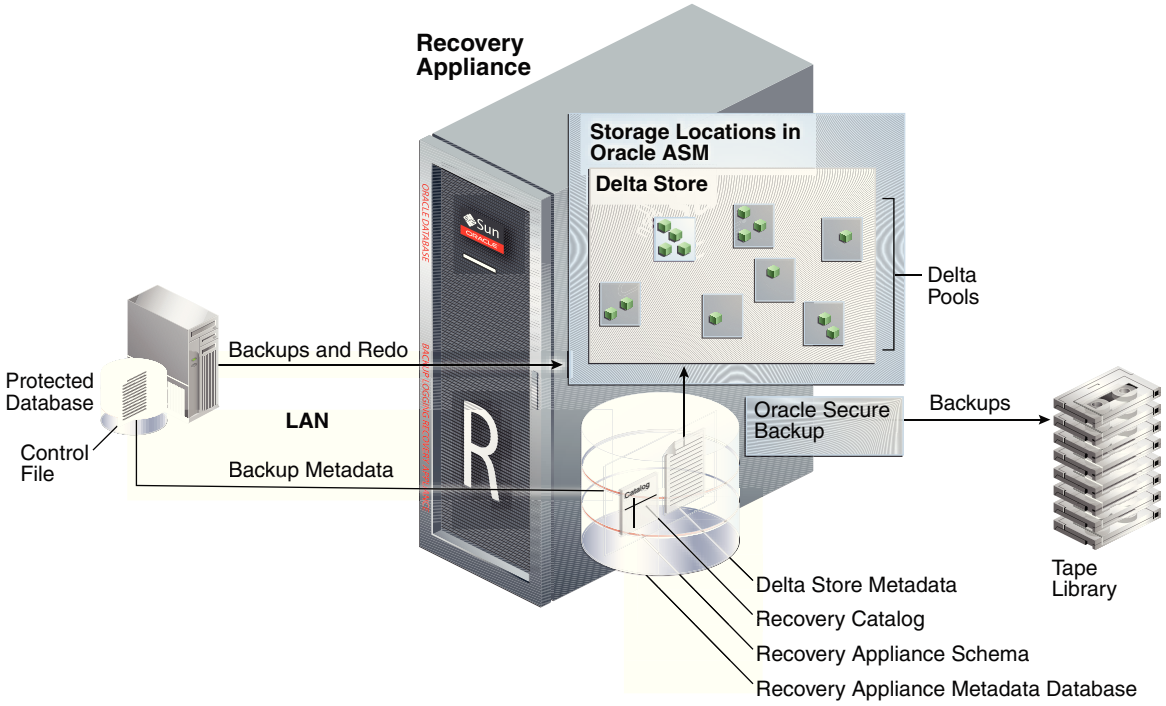
See Also:

Zero Data Loss Recovery Appliance Protected Database Configuration Guide to learn more about real-time redo transport and how to enable it

Recovery Appliance Metadata Database

The key component of the Recovery Appliance is the Recovery Appliance metadata database. This database manages metadata for all backups, and also contains the RMAN recovery catalog. The Recovery Appliance metadata database is preconfigured, pretuned, and managed by the Recovery Appliance. Figure 2-5 depicts a Recovery Appliance metadata database interacting with a protected database.

Figure 2-5 Recovery Appliance Metadata Database



This section contains the following topics:

- Delta Store
- Delta Pools
- Automated Delta Pool Space Management
- Recovery Appliance Schema
- Recovery Appliance Catalog

Delta Store

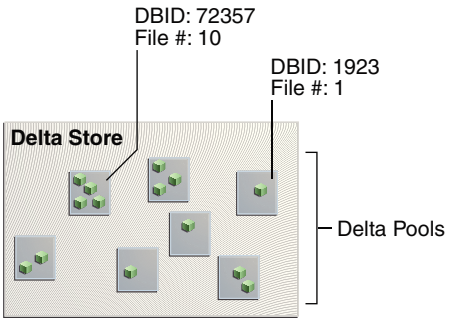
The delta store is the totality of all protected database backup data in Recovery Appliance storage location. All data file and archived redo log backups reside in the delta store. The delta store contains delta pools for all data files in all protected databases.

Delta Pools

The delta store is the collection of delta pools. As the Recovery Appliance receives backups from protected databases, it indexes them and stores them in delta pools. A delta pool is the set of data file blocks from which Recovery Appliance constructs virtual full backups. Recovery Appliance automatically manages delta pools so that it can create a virtual full backup that corresponds to any incremental backup ever received.

Each separate data file whose backups are sent to Recovery Appliance has its own separate delta pool. For example, data file `10` from `prod1` has its own delta pool, data file `1` from database `prod2` has its own delta pool, and so on. As shown in Figure 2-6, the delta store contains all the delta pools for the databases protected by Recovery Appliance.

Figure 2-6 Delta Pools in Delta Store



Automated Delta Pool Space Management

The set of operations by which Recovery Appliance manages backups is called automated delta pool space management. Specifically, space management involves the following automated tasks:

- Deleting backups (both in a Recovery Appliance storage location and on tape) that are obsolete or expired based on the disk recovery window goal and SBT retention policy

  Recovery Appliance periodically determines that some backups no longer need to be stored on disk, so their disk space can be reclaimed. When the Recovery Appliance determines that some backups residing in the delta pools are obsolete, the individual blocks that compose those backups are typically located in physical files alongside non-obsolete blocks. Recovery Appliance rewrites these physical files so that the delta pools can reclaim the space occupied by the obsolete blocks.

- Reorganizing the delta pools periodically to improve performance of restore operations

  The automatic tracking and reorganizing of the delta pools is called delta pool optimization. As old blocks are deleted and new incremental backups arrive for updated data files, the blocks in a backup can become less contiguous. This state can degrade the performance of restore operations. Recovery Appliance runs a

background task that automatically reorganizes virtual full backup blocks to maintain contiguity, thus optimizing read access for restore operations.

> **See Also:**
>
> "How Recovery Appliance Manages Storage Space"

Recovery Appliance Schema

The Recovery Appliance schema contains metadata used internally by the Recovery Appliance to manage backups on behalf of its protected databases. `RASYS` is the Recovery Appliance administrative user who owns the Recovery Appliance schema. The Recovery Appliance schema contains the RMAN recovery catalog.

> **See Also:**
>
> "User Accounts in the Recovery Appliance Environment"

Recovery Appliance Catalog

Updates to the recovery catalog reflect the results of Recovery Appliance indexing and space management collection. These updates do not occur in the control files of the protected databases. For this reason, protected databases that store backups in the Recovery Appliance must use the Recovery Appliance catalog.

> **Note:**
>
> Protected databases may use the recovery catalog in the Recovery Appliance without also using the Recovery Appliance as their backup repository.

RMAN connects to the Recovery Appliance catalog using the same Recovery Appliance account employed for backup and recovery operations. Each Recovery Appliance user account is also a virtual private catalog account. The `DBMS_RA.GRANT_DB_ACCESS` procedure grants Recovery Appliance privileges to a database user account for a specified protected database.

See Also:

- "Granting Database Access to a Recovery Appliance Account Using DBMS_RA"
- "GRANT_DB_ACCESS"
- *Oracle Database Backup and Recovery User's Guide* to learn how to manage a recovery catalog

Recovery Appliance Storage

Recovery Appliance uses the following types of storage:

- Recovery Appliance storage location

  This Oracle ASM location is the main storage for backups on Recovery Appliance disks, serving as the destination for protected database backups.

- Backup polling location

  An optional file system directory on shared storage, outside the Recovery Appliance, that is a destination for backup pieces and archived redo log files from a protected database. Recovery Appliance polls the directory at specified intervals, retrieves any found backups, and then processes and stores them.

- Redo staging area

  For Recovery Appliance installations that enable real-time redo transport, this is the destination for redo streams that protected databases transmit to Recovery Appliance. The staging area resides on Recovery Appliance disks. The Recovery Appliance collects data into archived redo log files, which it then converts to compressed archived redo log backups that it writes to a Recovery Appliance storage location.

See Also:

- "Recovery Appliance Storage Locations"
- "Backup Polling Locations"
- *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to configure real-time redo transport

Recovery Appliance Storage Locations

A Recovery Appliance storage location can be shared among multiple protected databases. The Recovery Appliance administrator decides which clients will use each storage location.

Benefits of Recovery Appliance Storage

The benefits of Recovery Appliance storage locations are:

- More efficient disk usage

  Recovery Appliance uses common storage to absorb spikes from all protected databases, reducing the total amount of over-allocated storage. In traditional RMAN backup and recovery, a fast recovery area stores recovery-related files. Individual fast recovery areas require that each database maintain the amount of storage required to accommodate its largest expected activity spike, which often results in wasted storage.

  > **Note:**
  >
  > The default storage location in the Recovery Appliance also contains a fast recovery area for catalog backups.

  Oracle recommends that protected databases continue to maintain fast recovery areas for storage of local online and archived redo log files, control file autobackups, and flashback logs. In a Recovery Appliance environment, the fast recovery areas have smaller space requirements because RMAN backups are stored in the Recovery Appliance.
- Database-optimized backup deduplication and compression
- Shared disk backup pool distributed based on database protection policy, which defines the disk recovery window goal for each database protected by the policy

Oracle ASM and Recovery Appliance Storage

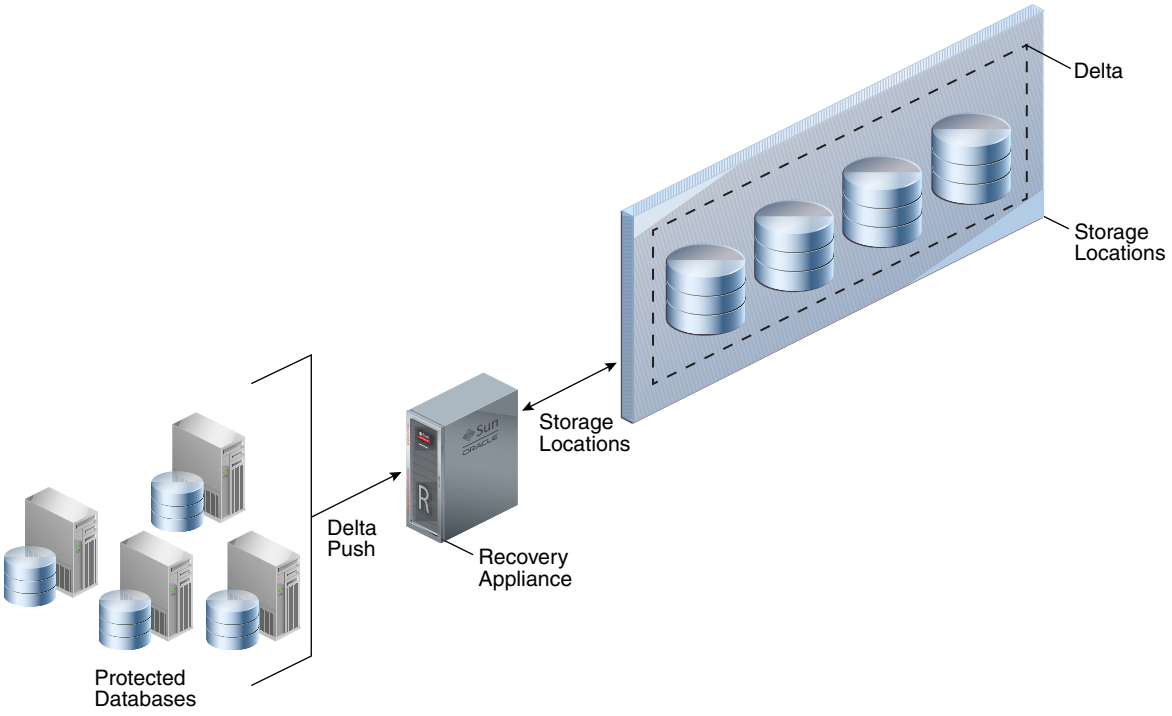
Recovery Appliance storage locations occupy space in Oracle ASM disk groups. By default, the delta pool is stored in normal redundancy Oracle ASM disk groups, which means that the Recovery Appliance maintains two copies of all on-disk backups. Database backups can survive the loss of any one disk or storage server. The Recovery Appliance metadata database, which tracks the files and blocks, is stored in a high redundancy Oracle ASM disk group.

> **See Also:**
>
> - "How Recovery Appliance Manages Storage Space"
> - *Oracle Automatic Storage Management Administrator's Guide*

DELTA Storage Location

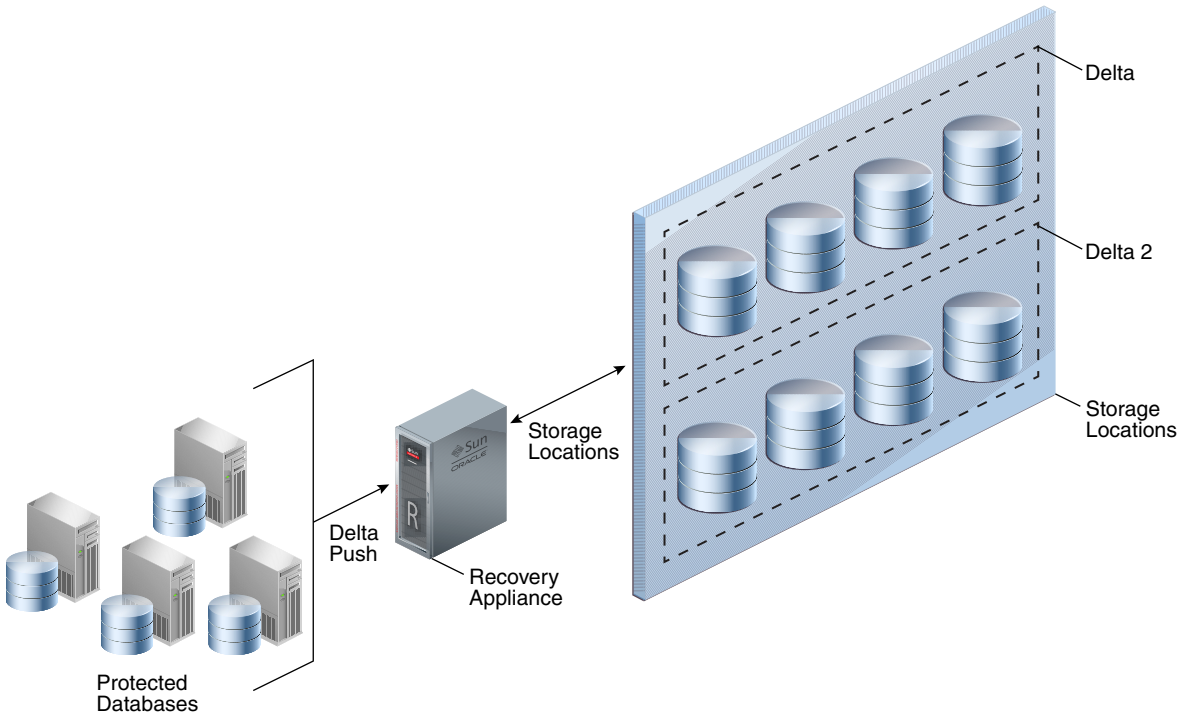
By default, Recovery Appliance is configured with all available disk storage assigned to a single storage location called `DELTA`. As shown in Figure 2-7, all protected databases share this storage location.

Figure 2-7 DELTA Storage Location



A more complex implementation may require multiple storage locations, for reasons such as expansion and space allotment. Figure 2-8 shows a Recovery Appliance configured with two storage locations: `DELTA` (default) and `DELTA2` (nondefault).

Figure 2-8 Multiple Storage Locations



Backup Polling Locations

A backup polling policy defines a file system directory where a protected database places backups without interacting directly with Recovery Appliance. The backup polling directory is an NFS mount point, and is not in a Recovery Appliance storage server.

The polling policy defines the file system path to the storage and how often it will be searched for new backups. Polling policies are optional and do not need to be created if backups are not sent to Recovery Appliance using the polling method.

> **See Also:**
>
> "Backup Polling Policies"

Stages of Backup Polling

Backup polling occurs in the following stages:

1. The protected database writes backups without the involvement of Recovery Appliance, which does not need to be running while backups are created.
2. Recovery Appliance polls for newly arrived backups.

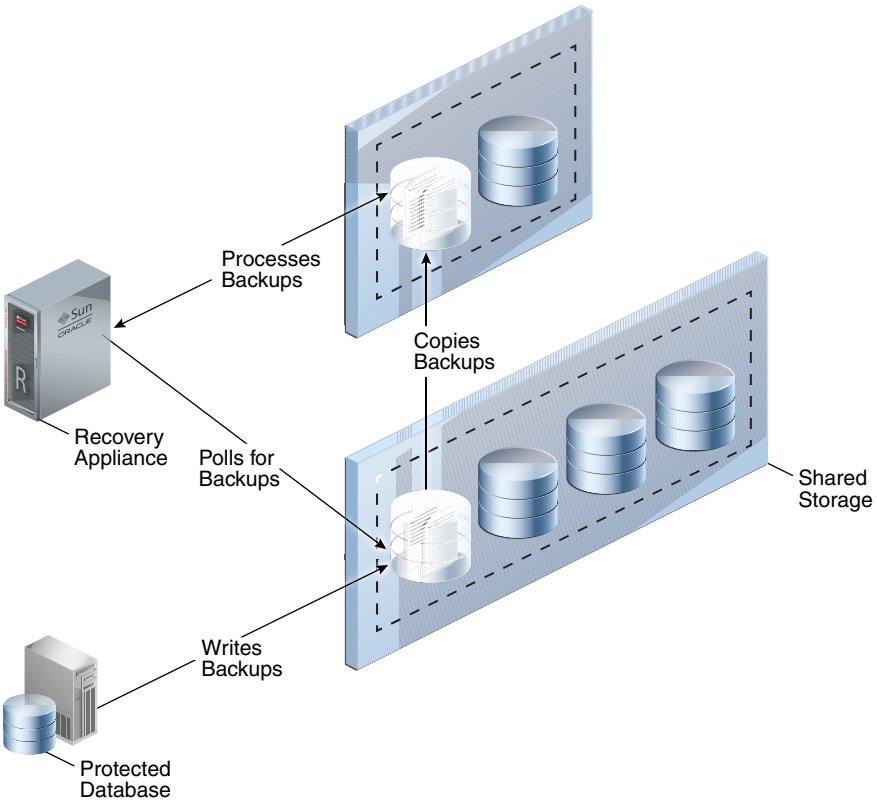
3. When Recovery Appliance discovers a file through polling, the Recovery Appliance examines its contents and tries to associate it with a protected database, and then does either of the following:
   - If the file is associated with a registered protected database, then Recovery Appliance processes the backup.
   - If the file is not associated with any registered protected database, then Recovery Appliance logs a warning message and does not process the file.

How Recovery Appliance Processes Backups in Backup Polling Directories

To set up polling so that backups are copied to Recovery Appliance storage, you must configure a backup polling directory that exists *outside* Recovery Appliance storage locations, but which the Recovery Appliance can access. The protected database writes its backups to the polling directory, which you specify in the polling policy.

The Recovery Appliance checks the polling directory for newly created backups. When backups exist, the Recovery Appliance copies the backups from the polling directory to its internal Recovery Appliance storage location, and then processes them. After enough time has passed for Recovery Appliance to copy the backups, the protected database deletes the backups from the polling directory. Figure 2-9 depicts this configuration.

Figure 2-9 Backup Polling



How Recovery Appliance Manages Storage Space

An important duty of the Recovery Appliance administrator is planning for the proper amount of disk space for a specified retention window and database size. As conditions change, the Recovery Appliance provides space management monitoring and alerting at the storage location and database level. When estimated storage needs are approaching the amount of available storage, alerts and warnings give the administrator time to accommodate the storage demands.

The following attributes, whose settings are accessible through the RA_DATABASE view, determine how Recovery Appliance manages storage space and backup retention:

- Recovery Window Goal
- Reserved Space
- Guaranteed Copy
- Maximum Retention Window

> **See Also:**
>
> "Archival and Encrypted Backups" for special algorithms that apply to RMAN backups that are not part of the incremental-forever strategy

Recovery Window Goal

The `recovery_window_goal` parameter of `DBMS_RA.CREATE_PROTECTION_POLICY` specifies the interval (typically in days) within which point-in-time recovery must be possible, counting backward from the current time. Consider a `recovery_window_goal` setting of 1 day. At midnight on August 7, the goal is recoverability to any time between the current time and midnight on August 6. At midnight on August 8, the goal is recoverability to any time between the current time and midnight on August 7, and so on.

Recovery Appliance attempts to retain sufficient backups to meet the recovery window goal defined for each database. For example, a Recovery Appliance protects databases `STORE01`, `STORE02`, and `STORE03`. The recovery window goal for `STORE01` is 1 day. If at midnight on August 7, `STORE01` needs 624.2 GB for backups to meet its recovery window goal, then the Recovery Appliance attempts to ensure that at least this much space is allocated for `STORE01` backups.

If sufficient space exists in storage, then backups created before a recovery window goal may be available—although they are not guaranteed. If purging previous backups is not necessary, then the Recovery Appliance keeps them, effectively extending the time to which point-in-time recovery is available. For example, on August 7 the space available to `STORE01` might be 700 GB or more, even though only 624.2 GB is required. A similar situation may exist for `STORE02` and `STORE03`.

If sufficient space does *not* exist in storage, then by default (`guaranteed_copy=NO`) the Recovery Appliance may purge backups. When reclaiming space, the Recovery Appliance attempts to respect the recovery window requirement first.

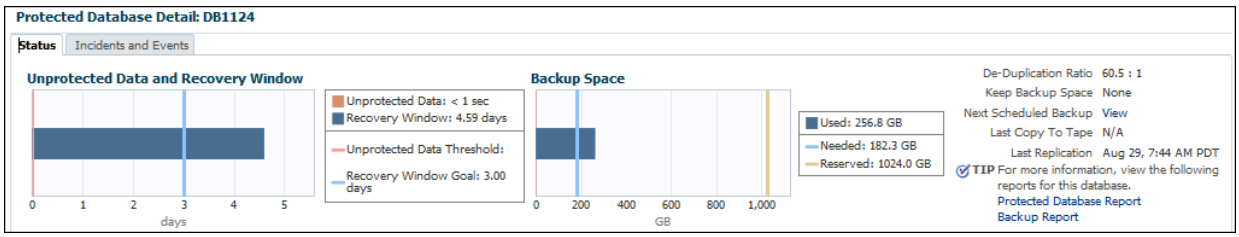
> **See Also:**
>
> - "Creating a Protection Policy"
> - "CREATE_PROTECTION_POLICY"

Reserved Space

Next in importance to the recovery window goal is the `reserved_space` parameter of `DBMS_RA.ADD_DB` and `DBMS_RA.UPDATE_DB`. The reserved space defines the amount of disk space guaranteed to each protected database to meet its recovery window goal.

> **Note:**
>
> This is the only storage parameter that is specified in `ADD_DB` rather than in the protection policy.

Because backups need space, you must estimate how much space you believe is needed to store backups. For example, you might allocate 1024 GB of reserved space to the `DB1124` database, which means the Recovery Appliance guarantees 1024 GB to `DB1124` *if* the database needs this amount to meet its recovery window goal. The following graphic shows a section of a Protected Database Details report:



In the preceding example, the disk recovery window goal for `DB1124` is 3 days, and the actual recovery window (the time to which the Recovery Appliance can currently recover) is 4.59 days. Meeting the recovery window goal requires 182.3 GB of backup data. This amount is less than 20% of the specified reserved space setting of 1024 GB. By default, at any given time, a database may actually have more or less than its specified reserved space available.

> **Note:**
>
> Reserved space is measured in space (GB), whereas the recovery window goal is measured in time.

The Recovery Appliance uses recovery window goals and reserved space settings to allocate storage *dynamically* to meet business requirements. If the Recovery

Appliance has purged as much backup data as possible while still meeting the recovery window goal for each database, and if more space is needed, then the Recovery Appliance evaluates the reserved space setting of each database. Recovery Appliance purges backups for the database whose backups exceed the reserved space by the highest percentage, and logs a message in the RA_INCIDENT_LOG view. Query the RA_PURGING_QUEUE view to determine which database will next have a backup purged.

> **See Also:**
>
> - "Adding Protected Database Metadata Using DBMS_RA"
> - "Accessing the Protected Database Details Report from the Recovery Appliance Reports Page"
> - "CREATE_PROTECTION_POLICY"
> - "ADD_DB"

Guaranteed Copy

A key question in storage management is whether it is more important to ensure that older backups are copied or replicated than it is to accept new backups or redo. The following settings are possible for the `guaranteed_copy` parameter of `DBMS_RA.CREATE_PROTECTION_POLICY`:

- `NO` (default)

  Recovery Appliance can purge backups before they have been copied to tape or replicated when it is necessary to make space for newer backups. In this case, the protected database may have more or less than the reserved space.

- `YES`

  When guaranteed copy is enabled, Recovery Appliance never purges a backup before it has been copied to tape or replicated. Also, the Recovery Appliance never permits the total size of backups that have not been copied or replicated to exceed the `reserved_space` setting for a database.

  After the Recovery Appliance has consumed reserved space with backups that have not been copied to tape or replicated, the Recovery Appliance cannot accept new backups or redo. This setting only changes the behavior of storage management when the tape system or replicated Recovery Appliance is unavailable for an extended period.

  The Recovery Appliance uses a different algorithm for virtual backups that are part of an incremental-forever backup strategy. Nonvirtual backups occupy a specific amount of space and either have or do not have a tape copy. For virtual backups, the tape schedule may write either a level 1 or level 0 version of any virtual backup in the Recovery Appliance. Additionally, space computations for virtual backups are complex because the space includes blocks needed to support the backup, which may differ from the space needed to write the backup to tape. For these reasons, after the Recovery Appliance writes a virtual data file backup to tape, the Recovery Appliance considers all versions of this backup and any older virtual backups of this data file as copied to tape (or replicated).

See Also:

- "Delta Push"
- "Creating a Protection Policy"
- "CREATE_PROTECTION_POLICY"

Maximum Retention Window

The `max_retention_window` parameter of `DBMS_RA.CREATE_PROTECTION_POLICY` specifies the maximum time that the Recovery Appliance retains backups for databases using this policy. Specifying null means that no backup purging occurs unless caused by space pressures within a storage location, or user actions.

The Recovery Appliance only keeps backups longer than the retention window when necessary to preserve the recovery window goal for a database. The effect of this setting is that the Recovery Appliance deletes backups sooner than it might otherwise have chosen to delete them.

See Also:

- "Creating a Protection Policy"
- "CREATE_PROTECTION_POLICY"

Archival and Encrypted Backups

The following types of backups cannot be part of an incremental-forever strategy, or be used to construct virtual full backups:

- RMAN archival backups created using the `BACKUP ... KEEP` command
- RMAN encrypted backups created using `CONFIGURE` or `SET ENCRYPTION`

The Recovery Appliance manages the preceding backups differently from backups in an incremental-forever strategy. Recovery Appliance retains archival backups regardless of the specified recovery window goal. However, encrypted backups do adhere to recovery window settings.

Archival backups are eligible for deletion by the Recovery Appliance only after the `KEEP` time expires. If you intend to store archival backups for an extended time, then note the following guidelines:

- Adjust the reserved space to account for them. Archival backups reduce the space available for achieving your recovery window goal and must be accounted for.
- Because the Recovery Appliance does not automatically copy archival backups to tape, you must manually copy them using the `COPY_BACKUP` procedure. This procedure also enables you to copy archival backups to disk locations that are outside Recovery Appliance storage locations. The `MOVE_BACKUP` procedure

copies an archival backup to disk or tape and then deletes it from the storage area.

> **See Also:**
>
> - "Copying Backups to Tape with Recovery Appliance "
> - "Data Encryption Techniques"
> - *Oracle Database Backup and Recovery User's Guide* to learn more about archival backups
> - My Oracle Support Note Doc ID 2107079.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=2107079.1`) to learn how to create archival backups for long term retention on the Recovery Appliance

Oracle Secure Backup

Oracle Secure Backup is the tape management component of Recovery Appliance. The Recovery Appliance offloads tape backup operations from protected databases to the Recovery Appliance. Thus, protected database hosts do not need the RMAN-integrated media management software module. Instead, a single copy of the Oracle Secure Backup module is installed on Recovery Appliance. The Recovery Appliance automatically manages the copy of backups to tape for all protected databases.

> **See Also:**
>
> Oracle® Database Backup and Recovery Reference.

Tape Archival

Protected databases send backups to the Recovery Appliance, which stores them on disk in the specified storage locations. To reclaim disk space and to create transportable tape backups, business requirements may necessitate archival to tape.

Tape backups are a repetitive task that the Recovery Appliance automates and performs as a background task. The Recovery Appliance administrator configures rules that specify the frequency with which the Recovery Appliance creates tape backups. Because the protection policy is a natural grouping of databases, databases sharing the same protection policy can share the same tape archival requirements.

Recovery Appliance creates tape backups in the compatibility version that matches the greatest compatibility version of any full or incremental backup that contributes blocks to the backup. The database identity information that the Recovery Appliance sends to the media management layer is identical to the information that would be sent if the database were sending the backup. This consistency guarantees that the database for which the backups were created can use them.

> **See Also:**
>
> "Scheduling Tape Backup Jobs with Oracle Scheduler"

Tape Retrieval

Backups are stored on tape as complete backup sets, not virtual backups, so tape backups are usable by RMAN without mediation by the Recovery Appliance. You can restore backups from tape in the following ways:

- Retrieval by Recovery Appliance

  This is the simplest way to restore an SBT backup created by Recovery Appliance. RMAN requests the restore from Recovery Appliance, without needing to be aware that this particular backup has been moved to tape. Recovery Appliance recognizes that the requested backup set is located on tape, open an SBT session to restore the backup from the media manager, and transfer the data over the network to the protected database host.

- Retrieval by protected database

  Because the SBT backups exist in a client-compatible format, RMAN can restore the backups from tape directly to any host, without involving Recovery Appliance. In this case, RMAN must first catalog the backup pieces before it can restore them from tape, and the Oracle Secure Backup library must be installed on the protected database host.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* to learn how to restore backups from tape

Recovery Appliance Replication

In Recovery Appliance replication, one Recovery Appliance (the upstream Recovery Appliance) forwards backups to another Recovery Appliance (the downstream Recovery Appliance). After initial configuration, replication is fully automatic. Each Recovery Appliance in a replication topology manages its own protection and polling policies.

How a Downstream Recovery Appliance Processes Backups

To forward backups to a downstream Recovery Appliance, the upstream Recovery Appliance uses the same Recovery Appliance Backup Module that a protected database uses to send backups. The basic steps for processing backups are as follows:

1. A protected database uses its Recovery Appliance Backup Module to send backups to the Recovery Appliance.
2. The Recovery Appliance receives the backups and processes them as normal.

> **Note:**
>
> The downstream Recovery Appliance does not know that it serves in the downstream role. The logic for receiving and processing backups on the downstream Recovery Appliance is independent of what occurs on the upstream Recovery Appliance.

3. The upstream Recovery Appliance forwards the backups to the downstream Recovery Appliance.

> **Note:**
>
> When real-time redo transport is enabled, incoming redo changes are not replicated in real time by Recovery Appliance. When an archived redo log backup is created, the Recovery Appliance automatically replicates this backup along with the data file backups.

4. As it receives backups from the upstream Recovery Appliance, the downstream Recovery Appliance updates its own metadata database.
5. The upstream Recovery Appliance requests metadata updates from the downstream metadata.

Periodically, the upstream Recovery Appliance requests metadata updates from the Recovery Appliances directly downstream from it. On receiving a metadata request, the downstream Recovery Appliance sends metadata updates to the upstream Recovery Appliance, which updates its own recovery catalog. In Recovery Appliance replication, this process is known as reconciling.

> **See Also:**
>
> "Lifecycle of a Backup: Scenario"

Replication Use Cases

Because a downstream Recovery Appliance processes backups independently from the upstream Recovery Appliance, a downstream Recovery Appliance can have completely different policies for every database whose backups it is storing. This asymmetry allows for a wide variety of use cases to be configured, some of which are shown in Figure 2-10.

Figure 2-10 Recovery Appliance Replication Use Cases

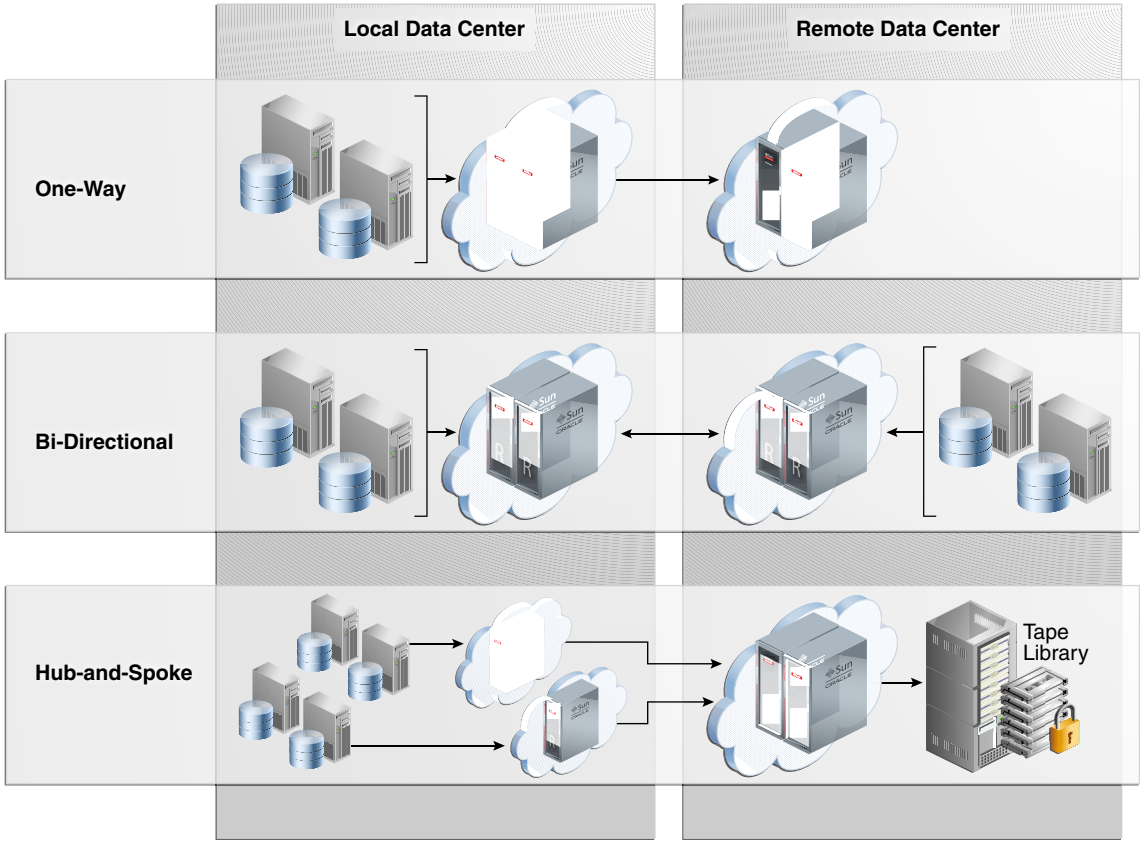


Figure 2-10 depicts the following use cases:

- One-Way

  Backup data flows from local databases to an upstream Recovery Appliance, which forwards it to a remote downstream Recovery Appliance.

- Bi-Directional

  Backups flow from local databases to a local Recovery Appliance, which then forwards them to a remote Recovery Appliance. Conversely, backups flow from remote databases to a remote Recovery Appliance, which then forwards them to the local Recovery Appliance.

  In this case, each Recovery Appliance plays both the upstream *and* downstream roles in the replication topology. Every Recovery Appliance serves as the primary backup location for one set of protected databases, and the secondary backup location for the other set. In this way, every Recovery Appliance is actively utilized while also providing disaster recovery services for the other Recovery Appliance.

- Hub-and-Spoke

  Backups flow from one set of databases to a local Recovery Appliance, and from a different set of databases to a different local Recovery Appliances. The local Recovery Appliances then forward these backups to a single remote Recovery Appliance, which archives the backups to tape.

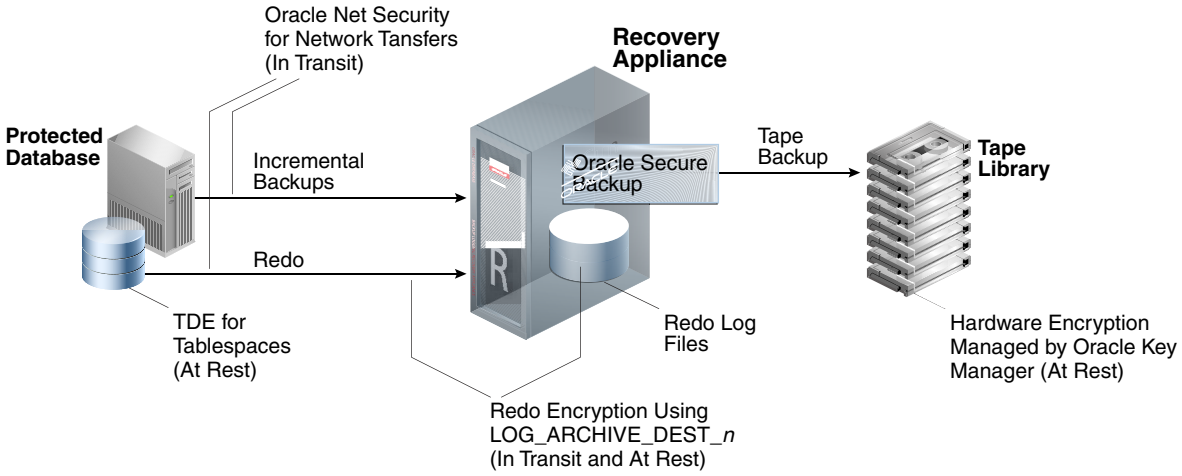
Any of the preceding use cases could be adapted for cascaded replication, in which a downstream Recovery Appliance forwards its backups to a second downstream Recovery Appliance, creating a one-way chain of Recovery Appliances.

> **See Also:**
>
> Replicating Backups with Recovery Appliance

Data Encryption Techniques

Various encryption options are available for backups and redo sent to the Recovery Appliance, as shown in Figure 2-11. The Recovery Appliance does not provide server-side encryption, which means that the appliance does not itself encrypt and decrypt data.

Figure 2-11 Data Encryption Techniques



The following types of encryption are supported:

- Transparent Data Encryption (TDE) on Production Database Tablespaces
- Oracle Net Security for Network Transfers to the Recovery Appliance
- Redo Encryption Using LOG_ARCHIVE_DEST_n
- Tape Drive-Based Hardware Encryption

Transparent Data Encryption (TDE) on Production Database Tablespaces

Oracle recommends that you enable TDE on tablespaces in the database, and then take incremental backups as usual. TDE requires the Advanced Security Option. The benefits of TDE are as follows:

- TDE is transparent to applications.
- Backups of encrypted tablespaces, and redo describing changes to these tablespaces, are encrypted. The TDE-encrypted data blocks are secured on the protected database, Recovery Appliance storage, tape devices, and replicated appliances, and also when transferred through any network connections.
- TDE on the source database reduces overhead on downstream servers.
- This technique supports an incremental-forever strategy and virtual full backups.

> **Note:**
>
> Oracle does not recommend encrypting backups using the RMAN `SET` or `CONFIGURE ENCRYPTION` command. See "Archival and Encrypted Backups" for more information

The following table shows the support for incremental forever when RMAN encryption and/or RMAN compression are used for the protected database backups:

Table 2-4 Support for Incremental Forever with RMAN Encryption and RMAN Compression

Data in the Database	No RMAN Encryption and No RMAN Compression	RMAN Encryption	RMAN Compression	RMAN Encryption and RMAN Compression
Not Encrypted	Yes	No	Yes	No
TDE Tablespace Encryption	Yes	Yes	No	No

> **See Also:**
>
> - *Oracle Database Advanced Security Guide* to learn about TDE
> - *Oracle Database Backup and Recovery User's Guide* to learn about configuring backup encryption
> - *Oracle Database Backup and Recovery User's Guide* to learn about making compressed backups

Oracle Net Security for Network Transfers to the Recovery Appliance

Oracle Net security includes an LDAP-compliant directory server, digital certificates, and Secure Sockets Layer (SSL). Oracle Net security encrypts the transfer of backups and redo to the Recovery Appliance. This technique provides in-transit network encryption that is independent of TDE encryption of the database, redo encryption using `LOG_ARCHIVE_DEST_`*n*, or RMAN backup encryption.

> **See Also:**
>
> - *Oracle Database Security Guide* to learn how to configure SSL
> - *Oracle Database Advanced Security Guide* to learn about Oracle Net security

Redo Encryption Using LOG_ARCHIVE_DEST_n

When enabled, the `ENCRYPTION` attribute of `LOG_ARCHIVE_DEST_`*n* encrypts redo both at rest on the Recovery Appliance and during the network transfer to the appliance. The basic process is as follows:

1. The protected database encrypts the redo in memory, using the private key contained in the Oracle Wallet on the protected database.
2. The protected database transfers the redo to the Recovery Appliance over the network.

> **Note:**
>
> If Oracle Net security is also enabled, then the redo is double encrypted during network transfer.

3. The Recovery Appliance writes the encrypted redo to archived redo log files, which exist in encrypted form only on the Recovery Appliance.

In a recovery scenario, RMAN restores and decrypts the encrypted redo log files on the protected database, using the encryption key stored in the Oracle wallet on the protected database host (not on the Recovery Appliance). RMAN never applies encrypted redo log files during media recovery.

> **See Also:**
>
> - My Oracle Support Note Doc ID 1995866.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=1995866.1`) for versions of Oracle Database that support encrypted redo
> - *Oracle Data Guard Concepts and Administration* to learn about redo encryption using `LOG_ARCHIVE_DEST_`*n*

Tape Drive-Based Hardware Encryption

The Recovery Appliance supports tape drive-based hardware encryption. In this case, the tape drive encrypts the data, not the software.

> **Note:**
>
> Oracle Secure Backup can encrypt backup pieces before Recovery Appliance copies them to tape. However, Oracle does not recommend software-based encryption because of its possible negative effect on performance.

For key management, Oracle recommends Oracle Key Manager, which centrally authorizes, secures, and manages all encryption keys. Oracle Key Manager does not consume CPU on the Recovery Appliance when encrypting and decrypting data.

> **See Also:**
>
> *Oracle Secure Backup Administrator's Guide* to learn about hardware encryption

Part I
Managing Recovery Appliance

Part I contains the following chapters:

3
Recovery Appliance Workflow

This chapter explains the basic workflow for managing a Zero Data Loss Recovery Appliance environment. Where appropriate, this chapter refers to *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*. The chapter contains the following topics:

- Separation of Duties in Recovery Appliance Administration
- Prerequisites for Recovery Appliance Administration
- Tools for Recovery Appliance Administration
- Planning for Recovery Appliance
- Setup and Configuration for Recovery Appliance
- Maintenance Tasks for Recovery Appliance

Separation of Duties in Recovery Appliance Administration

A typical Recovery Appliance environment includes personnel with the following roles:

- Cloud Control administrator

  The application administrator with this role administers Oracle Enterprise Manager Cloud Control (Cloud Control). Duties may include:

  - Discovering targets, including the Recovery Appliance
  - Managing one or more protected databases
  - Managing one or more Recovery Appliances
- Recovery Appliance administrator

  This administrator manages Recovery Appliance. Typical duties include:

  - Creating the protection policies
  - Assigning protected databases to protection policies
  - Managing space on Recovery Appliance
  - Configuring tape and replication operations
  - Creating the Recovery Appliance user accounts that own virtual private catalogs
  - Monitoring Recovery Appliance, and generating reports
- Protected database administrator

  This administrator is responsible for configuring backups to the Recovery Appliance using the virtual private catalog account assigned by the Recovery Appliance administrator.

See Also:

- "User Accounts in the Recovery Appliance Environment"
- Cloud Control online help to learn how to create an Enterprise Manager Administrator account
- *Oracle Database Security Guide* to learn how to create database user accounts

Prerequisites for Recovery Appliance Administration

You must work with the Oracle field engineers to install and set up Recovery Appliance.

Tools for Recovery Appliance Administration

Use the following tools to complete administrative tasks for Recovery Appliance:

- Cloud Control

  Cloud Control is a system management tool with a graphical user interface that enables you to manage and monitor Recovery Appliance and its protected databases. This is the preferred UI for Recovery Appliance tasks.

  See the Cloud Control online help for more information.

- SQL*Plus

  SQL*Plus is a command-line tool that enables you to run `DBMS_RA` program units, and query recovery catalog views. You use SQL statements and Oracle-supplied PL/SQL packages to complete these tasks in SQL*Plus.

  See *SQL*Plus User's Guide and Reference*.

Planning for Recovery Appliance

You must complete the following general tasks:

- Task 1: Group protected databases into tiers
- Task 2: Determine the recovery requirements for each database tier
- Task 3: Determine the recovery requirements for each protected database
- Task 4: Determine access requirements for Recovery Appliance
- Task 5: Create a backup migration plan to Recovery Appliance
- Task 6: Review Cloud Control reporting and monitoring tools

Task 1: Group protected databases into tiers

Group databases based on their recovery requirements. By default, Recovery Appliance includes the protection policies Platinum, Gold, Silver, and Bronze. Each policy corresponds to a level of protection. For example, Gold provides databases in this tier with real-time redo transport protection, whereas Bronze does not.

Task 2: Determine the recovery requirements for each database tier

For each database tier, make decisions about the following:

- The maximum amount of time for potential data loss exposure
- The disk recovery window goal
- The recovery window for tape
- The schedule for database tiers that back up to tape, and any tape vaulting or encryption requirements
- Whether to configure Recovery Appliance replication
- Directories for backup polling, if you intend to enable a backup polling policy
- Whether existing recovery catalogs will be imported into the Recovery Appliance catalog
- Whether to enable the guaranteed copy feature, which requires that backups on Recovery Appliance be copied to tape or replicated before being considered for deletion to reclaim space
- The maximum retention time of backups on disk

> **See Also:**
>
> "About Protection Policies"

Task 3: Determine the recovery requirements for each protected database

For example, perform the following tasks:

- Calculate the reserved space, which is based on the protected database size, change rate, and recovery window goal
- Decide whether to implement real-time redo transport

See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* for additional planning considerations for protected databases.

Task 4: Determine access requirements for Recovery Appliance

Decide which persons have access to the Recovery Appliance in the data center. For example, database administrators, storage administrators, system administrators, and backup administrators may have different access requirements. In some data centers, a single person may play all roles.

Task 5: Create a backup migration plan to Recovery Appliance

In this stage, decide how your legacy RMAN backups fit into your Recovery Appliance backup strategy. After setting up Recovery Appliance, you may choose either of the following strategies:

- Continue to run old backups to disk and tape concurrently with new backups to Recovery Appliance for a specified time, until you are ready to back up to Recovery Appliance exclusively.

- Back up protected databases exclusively to Recovery Appliance, and then manage legacy backups on legacy media separately.

In either case, to simplify overall catalog management, Oracle recommends that you first import legacy RMAN recovery catalogs into the Recovery Appliance catalog.

> **See Also:**
>
> *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to import metadata into the Recovery Appliance catalog

Task 6: Review Cloud Control reporting and monitoring tools

Cloud Control is the preferred interface for Recovery Appliance. Before configuring Recovery Appliance, become familiar with the main Cloud Control pages, as described in Getting Started with Cloud Control for Recovery Appliance . Database administrators can also review backup-related pages such as Backup Settings, Schedule Backup, and Backup Reports.

> **See Also:**
>
> - Monitoring the Recovery Appliance
> - Accessing Recovery Appliance Reports

Setup and Configuration for Recovery Appliance

You must complete the following general tasks:

- Task 1: Create Cloud Control user accounts
- Task 2: Create a protection policy for each database tier
- Task 3: Configure access on Recovery Appliance for protected databases
- Task 4: Configure protected databases (for DBAs)
- Task 5: Migrate legacy backups to Recovery Appliance (for DBAs)
- Task 6: Create copy-to-tape schedules to meet recovery requirements
- Task 7: Configure Recovery Appliance replication

Task 1: Create Cloud Control user accounts

As explained in "Separation of Duties in Recovery Appliance Administration", a Recovery Appliance environment may require multiple administrative accounts. In this step, create the Cloud Control user accounts necessary for your environment.

> **Note:**
>
> These are application-level user accounts, not database user accounts.

> **See Also:**
>
> Cloud Control help to learn how to create Enterprise Manager user accounts

Task 2: Create a protection policy for each database tier

For each tier of protected databases, create a separate protection policy. "Basic Tasks for Managing Protection Policies" describes these tasks.

1. Optionally, if your Recovery Appliance has access to a backup polling location, then create a backup polling policy.

   > **Note:**
   >
   > If you are using Cloud Control, then this step is included in the protection policy configuration. When using `DBMS_RA`, you must run a separate procedure (`CREATE_POLLING_POLICY`).

   "Creating a Backup Polling Policy (Command-Line Only)" describes this task.
2. Create a protection policy for a specific database tier.

   "Creating a Protection Policy" describes this task.

Task 3: Configure access on Recovery Appliance for protected databases

Create a virtual private catalog owner in the Recovery Appliance metadata database, add protected database metadata, and grant the catalog owner access to protected databases. Perform all of these steps on the Recovery Appliance, as explained in "Basic Tasks for Configuring Protected Database Access".

Task 4: Configure protected databases (for DBAs)

Protected database administrators perform this task, which does not involve running `DBMS_RA` procedures on Recovery Appliance. Client-side configuration includes the following subtasks:

1. Configuring backup and recovery settings, including real-time redo transport
2. Enabling access to the Recovery Appliance, which involves installing the Recovery Appliance Backup Module and authenticating the Recovery Appliance user account
3. Testing backup and restore operations

See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to configure protected databases.

Task 5: Migrate legacy backups to Recovery Appliance (for DBAs)

DBAs for protected databases perform this task, which does not involve running `DBMS_RA` procedures on Recovery Appliance. Migration includes importing legacy recovery catalogs into the Recovery Appliance catalog, and enabling the Recovery Appliance to access physical backups on disk or tape.

See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to migrate legacy backups.

Task 6: Create copy-to-tape schedules to meet recovery requirements

If you employ tape devices in your environment, then you must create SBT attribute sets, schedule tape jobs, monitor tape backup status, and so on. You perform all of these steps on the Recovery Appliance, as explained in "Basic Tasks for Copying Backups to Tape with Recovery Appliance".

Task 7: Configure Recovery Appliance replication

This task involves configuring both the upstream Recovery Appliance and the downstream Recovery Appliance, and performing some steps on the protected database hosts. See "Basic Tasks for Configuring Recovery Appliance Replication".

Maintenance Tasks for Recovery Appliance

Typically, you must perform the following tasks:

- Task 1: Monitor activity on Recovery Appliance
- Task 2: Monitor backup jobs (for DBA)
- Task 3: Generate and review reports on Recovery Appliance
- Task 4: Restart the Recovery Appliance

Task 1: Monitor activity on Recovery Appliance

Using Cloud Control, monitor Recovery Appliance to ensure that business requirements are being met. For example, do the following:

- Review any alerts or warnings
- Verify that available space can meet all recovery windows
- Verify that backup throughput meets performance requirements

See "Basic Tasks for Monitoring the Recovery Appliance".

Task 2: Monitor backup jobs (for DBA)

Protected database administrators must periodically monitor backup job reports for errors.

See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*.

Task 3: Generate and review reports on Recovery Appliance

Using Cloud Control, generate and review BI Publisher reports for storage usage and capacity planning.

See "Basic Tasks for Accessing Recovery Appliance Reports".

Task 4: Restart the Recovery Appliance

If necessary, shut down and start up the Recovery Appliance using operating system utilities and `DBMS_RA` procedures. See *Zero Data Loss Recovery Appliance Owner's Guide* to learn how to restart the Recovery Appliance.

4

Getting Started with Cloud Control for Recovery Appliance

This chapter explains how to access the principal pages in Oracle Enterprise Manager Cloud Control (Cloud Control) for Recovery Appliance, and contains the following sections:

- Displaying All Recovery Appliances in the Enterprise
- Accessing the Recovery Appliance Home Page
- Accessing the Recovery Appliance Storage Locations Page

Displaying All Recovery Appliances in the Enterprise

Cloud Control lists every Recovery Appliance in the enterprise. From this page, you can access the individual home page of any Recovery Appliance.

To display all Recovery Appliances in the enterprise:

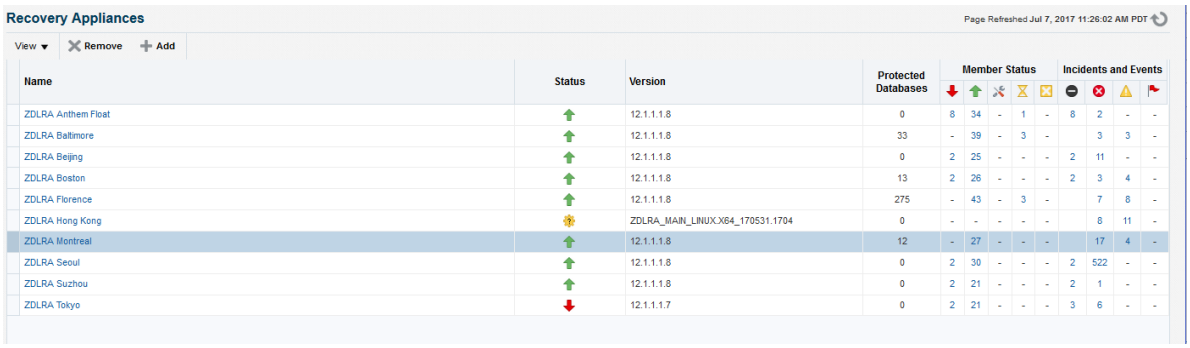
1. On the Cloud Control Login page, enter your `SYSMAN` user name and password.

   The Welcome to Enterprise Manager Cloud Control 12*c* page appears.

> **See Also:**
>
> "User Accounts in the Recovery Appliance Environment" for more information on user accounts in the Recovery Appliance environment

2. Select **Targets**, and then **Recovery Appliances**.

   The Recovery Appliances page appears. The following graphic shows a section of a sample page:



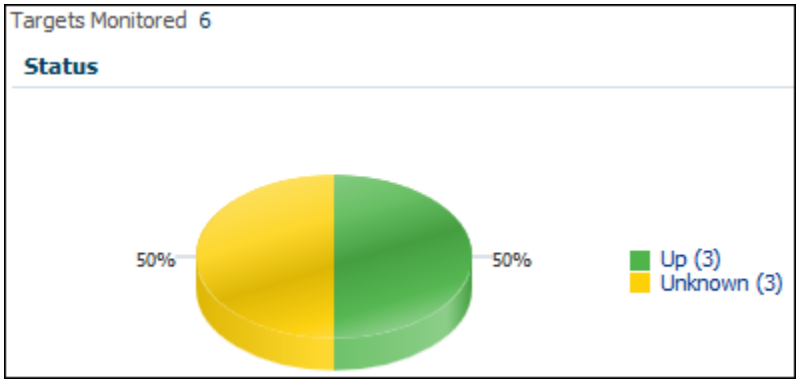
This page provides an overview of all Recovery Appliances in your environment. You can click the links in some columns (for example, Member Status or Incidents and Events) to go to pages with more information.

3. To see a pie chart showing the availability of the Recovery Appliances, select **Enterprise**, and then **Summary**.

   The Enterprise Summary page appears.

4. In the Overview section, in the **View** menu, select **Recovery Appliance**.

   The pie chart in the Status section indicates the percentage of Recovery Appliances that are available. The following sample graphic shows that half the six monitored Recovery Appliances are currently available:



See Also:

Cloud Control online help for more information about these pages

Accessing the Recovery Appliance Home Page

The Recovery Appliance Home page is a command center that centralizes management of the Recovery Appliance environment. From this page, you can manage Recovery Appliance storage and performance, and view recent activity and issues that may need attention. The Recovery Appliance Home page is divided into the following sections:

- Summary

  This section shows the number of protected databases, and summarizes their health status, current activity, and activity within the last 24 hours. For more information, click the links in the Operation column: Backup, Copy-to-Tape, Replication, and Restore.

- Protected Database Issues

  This section highlights any issues relating to backup and recovery status for protected databases. The View menu filters the information on key categories.

- Data Sent/Received (Daily)

  This section displays daily throughput over the past week.

- Performance

  This section charts performance statistics for Data Rate and Queued Data. The statistics are filterable by day, week, or month.

- Media Managers

  This section displays the configured media manager for copy-to-tape operations.

- Storage Locations

  This section summarizes total available space and usage by indicating how much has been consumed to meet the disk recovery window goal for all databases, and what percentage of total space is reserved space for databases backing up to the specified storage location (see "How Recovery Appliance Manages Storage Space").

- Replication

  This section lists the downstream Recovery Appliances to which this Recovery Appliance is replicating, and also the upstream Recovery Appliances from which this Recovery Appliance is receiving backups (see "About Recovery Appliance Replication").

- Incidents and Events

  This section summarizes all warnings or alerts that have been generated by Cloud Control monitoring of all targets associated with the Recovery Appliance. From this section, drill down for further detail on the issues.

To access the Recovery Appliance Home page:

1. On the Cloud Control Login page, enter your `SYSMAN` user name and password.

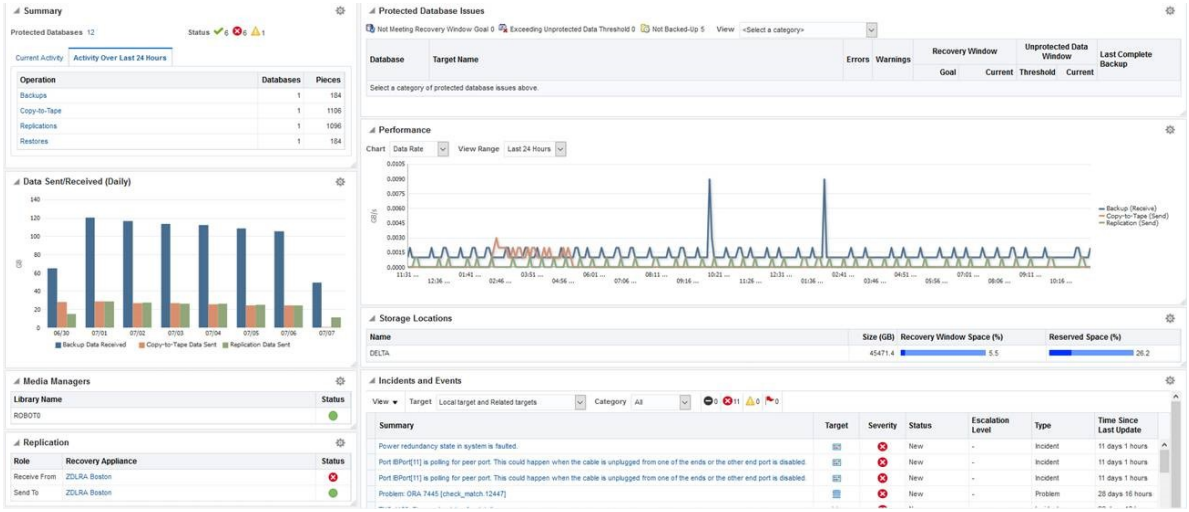
   > **See Also:**
   >
   > "User Accounts in the Recovery Appliance Environment" for more information on user accounts in the Recovery Appliance environment

2. From any Cloud Control page, select **Targets**, and then **Recovery Appliances**.

   The Recovery Appliances page appears.

3. In the Name column, click the name of a Recovery Appliance.

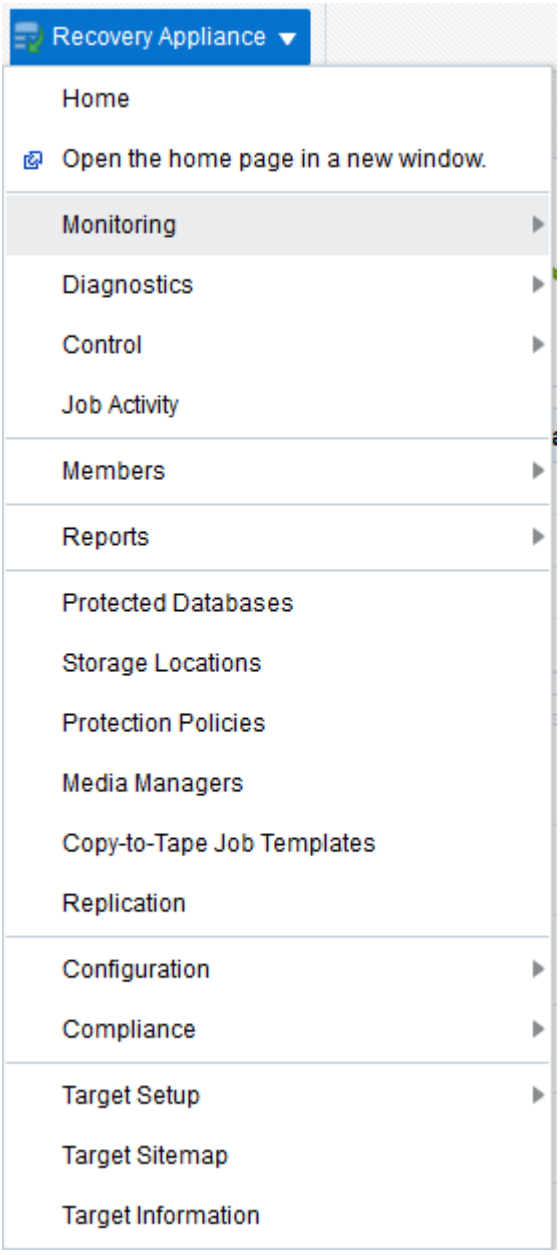
   The Home page for the selected Recovery Appliance page appears. The following graphic shows part of a sample Home page:



From this page you can see a snapshot of the entire Recovery Appliance, and also click links to obtain more information about a particular area.

4. Optionally, to access the main menu, click **Recovery Appliance**.

   The menu appears. The following graphic shows the menu options:



From the preceding menu you can go to all pages relating to management, monitoring, and reporting for this Recovery Appliance.

See Also:

Cloud Control online help for more information about these pages

Accessing the Recovery Appliance Storage Locations Page

The Storage Locations page expands on the information provided in the Storage Locations section on the Recovery Appliance Home page. This page provides the following storage-related information:

- Oracle ASM disk groups in the Recovery Appliance storage location
- Number of protection policies using the storage location
- Total space (in GB) needed to meet disk recovery window goals for all databases protected by this Recovery Appliance
- Total reserved space for all databases protected by this Recovery Appliance

Besides using this page to monitor existing storage, you can use this page add storage to an existing storage location, or to create a new storage location.

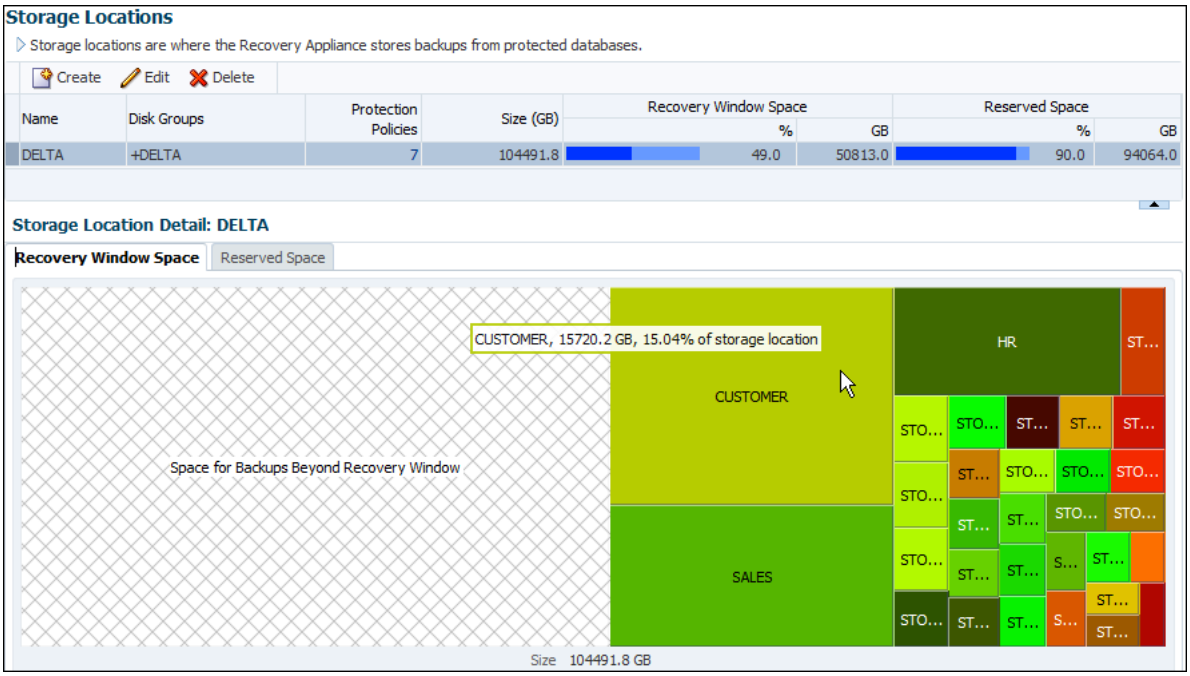
To access the Storage Locations page:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. From the **Recovery Appliance** menu, select **Storage Location**.

   The Recovery Appliance login page appears when you first log in to the Recovery Appliance pages, or when the browser has been inactive for an extended time.
3. If prompted, enter your login credentials, and then click **Login**.

   The Storage Locations page appears, as shown in Figure 4-1.

Figure 4-1 Storage Locations Page



This page provides a useful graphical representation of how much of the storage location is reserved and unreserved. In the preceding sample page, the DELTA storage location, which is the default, is the only storage location configured.

4. In the Storage Location Detail: DELTA section, click **Recovery Window Space** (if it is not already selected).

   The Recovery Window Space subpage appears.

   In the sample shown in Figure 4-1, the size of the DELTA storage location is 104491.8 GB and the total size of the needed recovery window space is 50813.0 GB. When the cursor hovers over the name of a protected database, as for the CUSTOMER database, a message indicates the amount of storage space needed by this database to meet its recovery window, and the percentage of the total storage space required.

5. In the Storage Location Detail: DELTA section, click **Reserved Space**.

   The Reserved Space subpage appears.

In the preceding graphic, the total size of the reserved space is 94064.0 GB. When the cursor hovers over the name of a protected database, as for the CUSTOMER database, a message indicates the amount of reserved space needed by this database, and the percentage of the total storage space required.

> **See Also:**
>
> Cloud Control online help for more information about these pages

5

Managing Protection Policies with Recovery Appliance

This chapter explains how to manage protection policies and polling policies, which are part of "Setup and Configuration for Recovery Appliance".

This chapter contains the following topics:

- About Protection Policies
- Creating a Backup Polling Policy (Command-Line Only)
- Creating a Protection Policy
- Updating a Protection Policy
- Deleting a Protection Policy

About Protection Policies

A protection policy is the central mechanism for controlling management of backup storage space, based on pre-defined recovery window goals. From the perspective of a DBA, the most important elements of a protection policy are the disk and tape recovery windows.

This section contains the following topics:

- Purpose of Protection Policies
- Overview of Protection Policies
- User Interfaces for Protection Policies
- Basic Tasks for Managing Protection Policies

> **See Also:**
>
> "Protection Policies" for an architectural overview

Purpose of Protection Policies

For every database associated with it, a protection policy specifies:

- The recovery window goal for disk backups
- The recovery window for tape backups
- Whether Recovery Appliance must replicate backups or copy them to tape before deleting them
- Which Recovery Appliance storage location is used for backups

- An optional backup polling policy

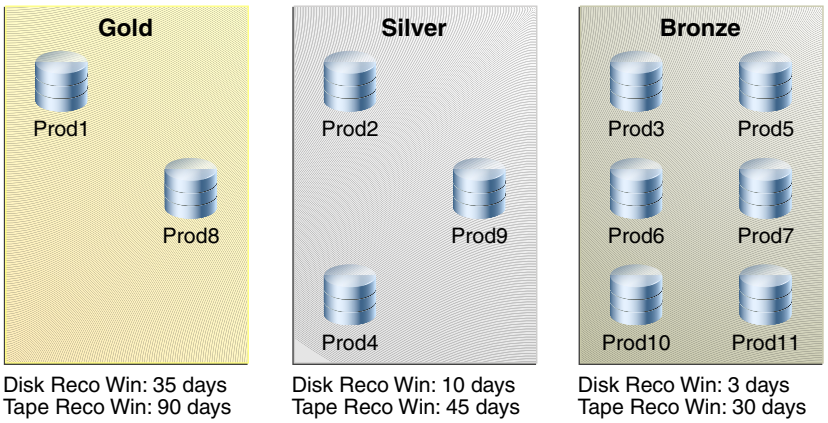
You can attach multiple protected databases to a single protection policy. A Recovery Appliance may have a variety of protection policies to support different data protection support levels. For example, protection policies can be generic service levels such as gold, silver, and bronze. Alternatively, policies can be specific to the requirements of protected databases and applications.

Overview of Protection Policies

A protection policy is a named, logical object recorded in the Recovery Appliance metadata database. To be added to a Recovery Appliance, a protected database must be associated with a specific protection policy. The default protection polices are Platinum, Gold, Silver, and Bronze.

Each protection policy specifies different values for the disk and tape recovery windows. These values apply to every database protected by the policy. For example, Figure 5-1 shows three of the default protection policies, with different protected databases assigned to each policy. In the example, databases `prod3` and `prod11` are in the same policy, and so both have the same disk recovery window goal of 3 days.

Figure 5-1 Protection Policies



> **See Also:**
>
> "Protection Policies"

User Interfaces for Protection Policies

This section contains the following topics:

- Accessing the Create Protection Policy Page in Cloud Control
- DBMS_RA Procedures Relating to Protection Policies
- Recovery Catalog Views for Protection Policies

Accessing the Create Protection Policy Page in Cloud Control

The Create Protection Policy page in Oracle Enterprise Manager Cloud Control (Cloud Control) is the recommended interface for creating protection policies.

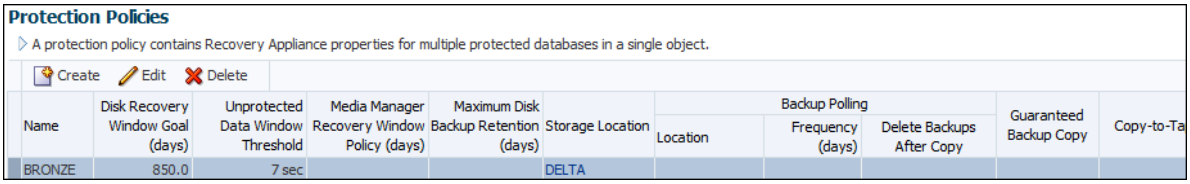
To access the Create Protection Policy page:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. From the **Recovery Appliance** menu, select **Protection Policies**.

   The Recovery Appliance Login page appears.
3. Enter your login credentials, and then click **Login**.

   The Protection Policies page appears, as shown in the example in Figure 5-2.

Figure 5-2 Protection Policies Page



> **See Also:**
>
> Cloud Control online help for more information about the Protection Policies page

DBMS_RA Procedures Relating to Protection Policies

You can use the `DBMS_RA` package to create and manage protection policies. Table 5-1 describes the principal program units relating to protection policies.

Table 5-1 DBMS_RA Protection Policy Procedures

Program Unit	Description
CREATE_POLLING_POLICY	Creates a backup polling policy.
CREATE_PROTECTION_POLICY	Creates a protection policy.
DELETE_PROTECTION_POLICY	Deletes a protection policy.

> **See Also:**
>
> DBMS_RA Package Reference

Recovery Catalog Views for Protection Policies

You can monitor protection policies using the Recovery Appliance catalog views. Table 5-2 summarizes the views that are most relevant for protection policies.

Table 5-2 Recovery Catalog Views for Protection Policies

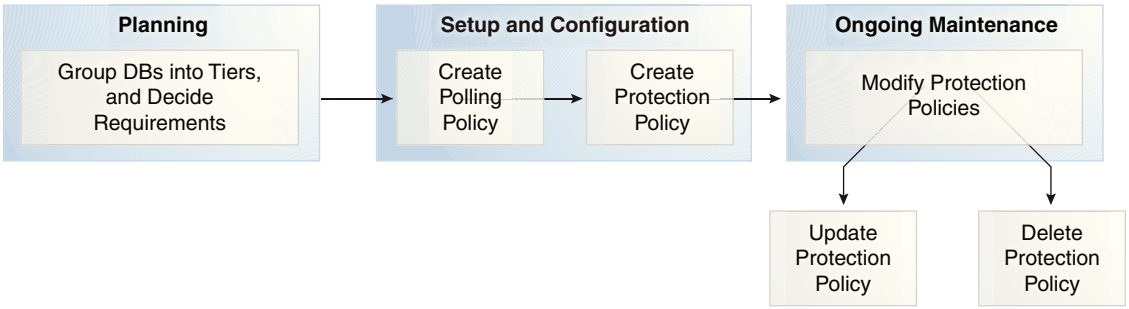
View	Description
RA_PROTECTION_POLICY	This view describes the defined protection policies.
RA_POLLING_POLICY	This view describes the defined backup polling policies.
RA_DATABASE	The POLICY_NAME column of this view lists the protection policy used by this protected database.
RA_REPLICATION_SERVER	The PROTECTION_POLICY column of this view lists the protection policy for a particular Recovery Appliance used for replication.

> **See Also:**
>
> Recovery Appliance View Reference

Basic Tasks for Managing Protection Policies

This section explains the basic tasks involved in managing protection policies. Figure 5-3 shows the overall workflow described in Recovery Appliance Workflow , with the protection policy tasks highlighted.

Figure 5-3 Protection Policy Tasks in Recovery Appliance Workflow



Typically, you perform protection policy tasks in the following sequence:

1. During the planning phase, group the databases into tiers, and decide the recovery requirements for each tier.

   "Planning for Recovery Appliance" describes these tasks.

2. During the configuration phase (see "Setup and Configuration for Recovery Appliance"), create one protection policy for each database tier.

a. Optionally, if your Recovery Appliance has access to a backup polling location, and if you are performing configuration using command-line tools, then create a backup polling policy.

   "Creating a Backup Polling Policy (Command-Line Only)" describes this task.

   > **Note:**
   >
   > Cloud Control enables you to configure the polling policy and the protection policy in the same page.

b. Create a protection policy for a specific database tier.

   "Creating a Protection Policy" describes this task.

3. During the ongoing maintenance phase (see "Maintenance Tasks for Recovery Appliance"), modify protection policies as needed. Typical modification tasks include:
   - Update the attributes of a protection policy.

     "Updating a Protection Policy" describes this task.
   - Delete a protection policy.

     "Deleting a Protection Policy" describes this task.

Creating a Backup Polling Policy (Command-Line Only)

An optional backup polling policy defines a directory where a protected database places backup sets without interacting directly with a Recovery Appliance. The backup polling directory must be created on an external file system and made accessible to a Recovery Appliance as an NFS mount point. The polling policy defines the file system path to the storage and how often the Recovery Appliance checks it for new backup sets (not image copies). Specify the polling policy name within a protection policy.

> **Note:**
>
> The separate step of creating a backup polling policy is not necessary in Cloud Control, which includes it in the Create Protection Policy page.

Backup polling policies are useful for the following reasons:

- If a Recovery Appliance is offline, then protected databases can continue to send backups to backup polling locations. When a Recovery Appliance comes online, it can poll these locations for backups.
- If the storage network is faster than your Ethernet, and if you configure the polling location in network storage, then protected database backups to the polling location may be faster.
- You can use a polling location when migrating legacy backups to a Recovery Appliance.

Protected databases that use backup polling store backup pieces and archived redo log files in shared storage. The Recovery Appliance periodically retrieves and processes backups from the shared storage.

Prerequisites

You must log in to the metadata database as RASYS.

Assumptions

Assume that you want to create a polling policy that meets the following requirements:

- The Recovery Appliance must poll the /u03/shared/polling1 directory, which is a shared directory accessible by the Recovery Appliance and all protected databases.
- You want the Recovery Appliance to poll the shared directory every 4 hours.
- You want the Recovery Appliance to delete backups from the shared directory after it has processed them.

To create a backup polling policy:

1. Start SQL*Plus or SQL Developer, and then log in to the metadata database as RASYS.
2. Run the DBMS_RA.CREATE_POLLING_POLICY procedure.

   For example, execute the following PL/SQL anonymous block:

```
BEGIN
  DBMS_RA.CREATE_POLLING_POLICY (
    polling_policy_name   => 'nas_polling1',
    polling_location      => '/u03/shared/polling1',
    polling_frequency     =>  INTERVAL '4' HOUR,
    delete_input          =>  TRUE);
END;
```

3. Optionally, query the recovery catalog to confirm creation of the policy.

   For example, query RA_POLLING_POLICY as follows (sample output included):

```
COL POLLING_NAME FORMAT a15
COL DEST FORMAT a40
SELECT POLLING_NAME, DEST, DELETE_INPUT,
       TO_CHAR(EXTRACT(DAY FROM FREQUENCY),'fm00')||':'||
       TO_CHAR(EXTRACT(HOUR FROM FREQUENCY),'fm00')||':'||
       TO_CHAR(EXTRACT(MINUTE FROM FREQUENCY),'fm00')||':'||
       TO_CHAR(EXTRACT(SECOND FROM FREQUENCY),'fm00')
         AS "DD:HH:MM:SS"
FROM   RA_POLLING_POLICY;

POLLING_NAME    DEST                                    DELET DD:HH:MM:SS
--------------- --------------------------------------- ----- --------------
NAS_POLLING1    /u03/shared/polling1/                   TRUE  00:04:00:00
```

> **See Also:**
>
> - "Backup Polling Policies" for more information about polling
> - "CREATE_POLLING_POLICY" for descriptions of procedure arguments

Creating a Protection Policy

This section explains how to create a protection policy using either Cloud Control (recommended) or the `DBMS_RA.CREATE_PROTECTION_POLICY` procedure. The best practice is to create a separate protection policy for each tier of databases (see "Task 1: Group protected databases into tiers").

Creating a Protection Policy Using Cloud Control

This section describes how to create a protection policy from the Protection Policy page in Cloud Control.

Prerequisites

You must meet the following prerequisites:

- You must be logged in to the Recovery Appliance as `RASYS`.
- The protection policy must not be associated with any protected database. To delete a policy that is associated with one or more databases, you must associate those databases with different policies before you can delete the desired policy.

Assumptions

Assume that you want to create a protection policy called `bronze_dev` for a tier of databases in a development environment. You have the following requirements for all databases protected by this policy:

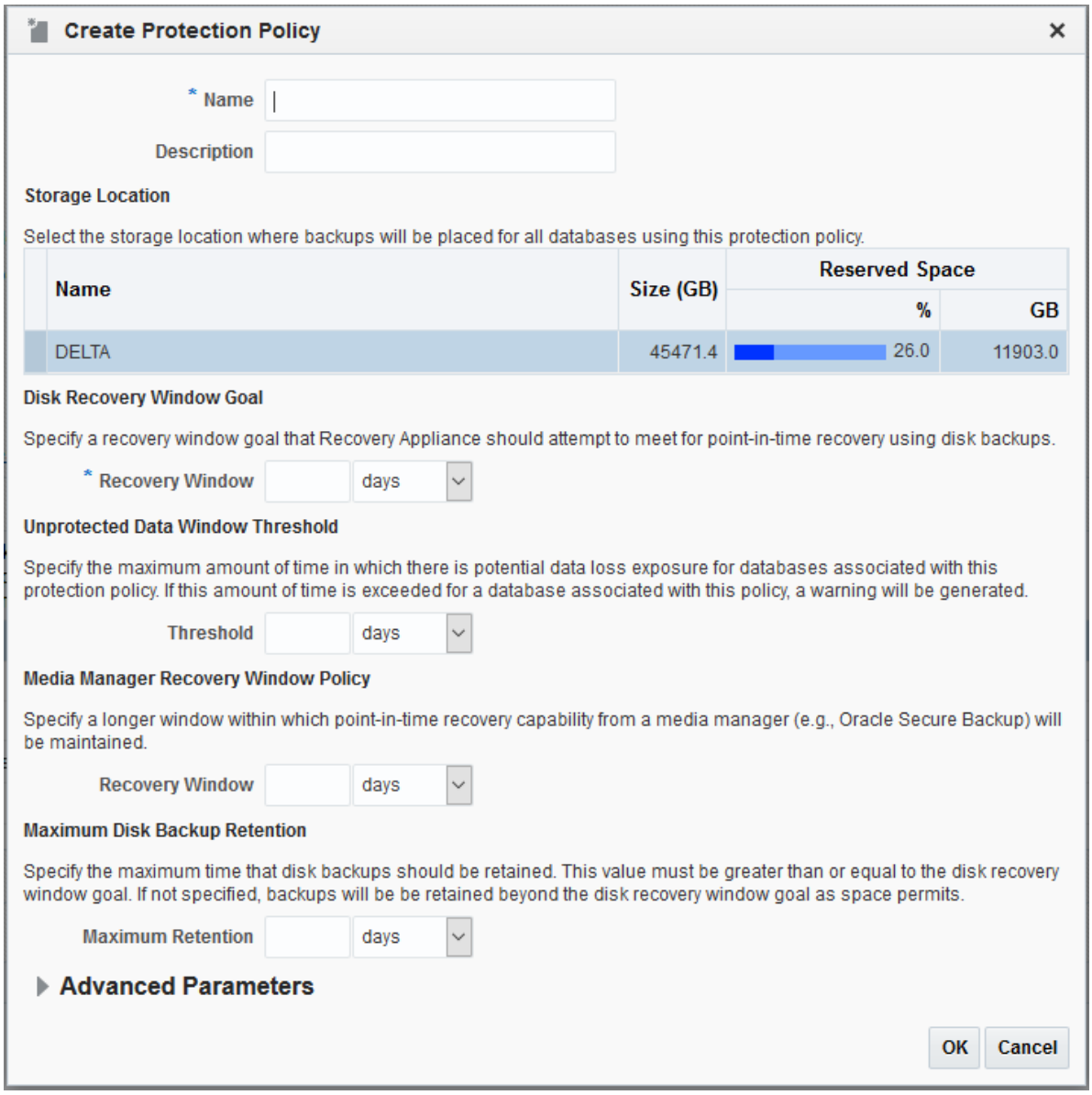
- The disk recovery window goal is 3 days, which means that every database must be recoverable using disk backups to any time within the last 3 days, starting from the current time.
- You do not want to archive backups to tape.
- You want the Recovery Appliance to receive new backups even if old backups must be deleted because available storage space is low.
- No backup polling policy is enabled.

To create a protection policy:

1. Access the Protection Policies page, as described in "Accessing the Create Protection Policy Page in Cloud Control".
2. Click **Create**.

   The Create Protection Policy page appears.

Figure 5-4 Create Protection Policy Page



In this page, the default Recovery Appliance storage location `DELTA` is already selected.

3. Enter values as follows:
   - In the **Name** field, enter the name of the new protection policy.

     For example, enter `bronze_dev`.
   - In the **Description** field, enter a description for the new policy.

     For example enter, `Policy with disk recovery window of 3 days and no tape backup`.
   - In the **Recovery Window** field of the Disk Recovery Window Goal section, specify a recovery window goal that the Recovery Appliance should attempt to meet for point-in-time recovery using disk backups, and then select the units.

For example, enter 3 and then select **days**.

- In the Threshold field of the Unprotected Data Window Threshold section, enter the maximum tolerable interval for data loss.

  For example, enter 5 and then select **minutes**.

- In the **Recovery Window** field of the Media Manager Recovery Window Policy section, optionally specify a larger window within which point-in-time recovery from a media manager will be maintained.

  For example, if no tape backup is desired, then leave this field blank.

- In the **Maximum Retention** field of the Maximum Disk Backup Retention section, enter the maximum time that the Recovery Appliance must retain disk backups.

  For example, leave this field blank, which means that the Recovery Appliance does not purge backups unless you explicitly purge them or space pressures exist within a storage location.

- Optionally, in the Backup Polling Location section, define a backup polling policy.
  - In the **Location** field, specify a directory accessible by the Recovery Appliance.
  - In the **Frequency** field, specify a time interval, and then select the time units.
  - To specify that the Recovery Appliance must delete the backups from the polling location after copying them, select **Delete Backups After Copy**.

  For example, to specify that backup polling is disabled, leave the fields blank.

- In the Backup Copy Policy section, specify whether the Recovery Appliance must replicate backups or copy backups to tape before deleting them.

  For example, select **Always accept new backups even if it requires purging existing backups not yet copied to tape or replicated**.

4. Click **OK**.

   The Protection Policies page appears, with the newly created protection policy listed.

See Also:

- "How Recovery Appliance Manages Storage Space"
- "Backup Polling Policies"
- "Backup Polling Locations"
- Cloud Control online help for more information about the Create Protection Policy page

Creating a Protection Policy Using DBMS_RA

To create a protection policy, execute the `DBMS_RA.CREATE_PROTECTION_POLICY` procedure.

Prerequisites

You must log in to the metadata database as `RASYS`.

Assumptions

Assume that you want to create a protection policy named `bronze_dev` for a tier of databases in a development environment. You have the following requirements for all databases protected by this policy:

- The disk recovery window goal is 3 days, which means that every database must be recoverable using disk backups to any time within the last 3 days, starting from the current time.
- You do not want to archive backups to tape.
- You want the Recovery Appliance to receive new backups even if old backups must be deleted because available storage space is low.
- No backup polling policy is enabled.

You also want to create policies for `gold_dev`, with a disk recovery window goal of 35 days, and `silver_dev`, with a disk recovery window goal of 10 days. Additionally, you create one policy named `test_dev` with a disk recovery window goal of 12 hours.

To create protection policies:

1. Start SQL*Plus or SQL Developer, and then log in to the metadata database as `RASYS`.
2. Run the `DBMS_RA.CREATE_PROTECTION_POLICY` procedure.

   For example, execute the following PL/SQL anonymous block:

```
BEGIN
 DBMS_RA.CREATE_PROTECTION_POLICY (
  protection_policy_name => 'bronze_dev',
  description            => 'For protected dbs in bronze tier',
  storage_location_name  => 'delta',
  recovery_window_goal   => INTERVAL '3' DAY,
  guaranteed_copy        => 'NO');
 DBMS_RA.CREATE_PROTECTION_POLICY (
  protection_policy_name => 'silver_dev',
  description            => 'For protected dbs in silver tier',
  storage_location_name  => 'delta',
  recovery_window_goal   => INTERVAL '10' DAY,
  guaranteed_copy        => 'NO');
 DBMS_RA.CREATE_PROTECTION_POLICY (
  protection_policy_name => 'gold_dev',
  description            => 'For protected dbs in gold tier',
  storage_location_name  => 'delta',
  recovery_window_goal   => INTERVAL '35' DAY,
  guaranteed_copy        => 'NO');
 DBMS_RA.CREATE_PROTECTION_POLICY (
  protection_policy_name => 'test_dev',
```

```
  description           => 'Test policy',
  storage_location_name => 'delta',
  recovery_window_goal  => INTERVAL '12' HOUR,
  guaranteed_copy       => 'NO');
END;
```

3. Optionally, query the recovery catalog to confirm creation of the policy.

   For example, query `RA_PROTECTION_POLICY` as follows (sample output included):

```
COL POLICY_NAME FORMAT a11
COL DESCRIPTION FORMAT a36
SELECT POLICY_NAME, DESCRIPTION,
       TO_CHAR(EXTRACT(DAY FROM RECOVERY_WINDOW_GOAL),'fm00')||':'||
       TO_CHAR(EXTRACT(HOUR FROM RECOVERY_WINDOW_GOAL),'fm00')||':'||
       TO_CHAR(EXTRACT(MINUTE FROM RECOVERY_WINDOW_GOAL),'fm00')||':'||
       TO_CHAR(EXTRACT(SECOND FROM RECOVERY_WINDOW_GOAL),'fm00')
         AS "DD:HH:MM:SS"
FROM   RA_PROTECTION_POLICY
WHERE  POLICY_NAME LIKE '%DEV'
ORDER BY POLICY_NAME;

POLICY_NAME DESCRIPTION                          DD:HH:MM:SS
----------- ------------------------------------ --------------
BRONZE_DEV  For protected dbs in bronze_dev tier 03:00:00:00
GOLD_DEV    For protected dbs in gold_dev tier   35:00:00:00
SILVER_DEV  For protected dbs in silver_dev tier 10:00:00:00
TEST_DEV    Test policy                          00:12:00:00
```

See Also:

- "Guaranteed Copy"
- "Backup Polling Policies"

Updating a Protection Policy

This section explains how to update protection policies using either Cloud Control (recommended) or the `DBMS_RA` PL/SQL package.

Updating a Protection Policy Using Cloud Control

This section describes how to update a protection policy from the Protection Policy page in Cloud Control.

Prerequisites

You must be logged in to the Recovery Appliance as `RASYS`.

Assumptions

Assume that you created the `bronze_dev` policy as described in "Creating a Protection Policy Using Cloud Control". You want to update the disk recovery window goal from 3 days to 6 days.

To update a protection policy:

1. Access the Protection Policies page, as described in "Accessing the Create Protection Policy Page in Cloud Control".
2. In the Protection Policies table, select the protection policy that you want to edit.

   For example, select the `BRONZE_DEV` row.
3. Click **Edit**.

   The Edit Protection Policy page appears.
4. Change the desired values, and then click **OK**.

   For example, in the **Recovery Window** field of the Disk Recovery Window Goal section, enter `6`.

   The Protection Policies page appears, with the newly updated protection policy listed.

> **See Also:**
>
> Cloud Control online help for more information about the Protection Policies page

Updating a Protection Policy Using DBMS_RA

To update a protection policy, execute the `DBMS_RA.UPDATE_PROTECTION_POLICY` procedure. Parameters that are null retain their existing values. For example, if `guaranteed_copy` is currently `NO` for a protection policy, and if you specify null for this parameter in `DBMS_RA.UPDATE_PROTECTION_POLICY`, then the value remains `NO`.

Prerequisites

You must log in to the metadata database as `RASYS`. The protection policy `bronze_dev` that you created in "Creating a Protection Policy Using DBMS_RA" must exist.

Assumptions

Assume that you want to change the disk recovery window goal of `bronze_dev` from 3 days to 6 days.

To update the attributes of an existing protection policy:

1. Start SQL*Plus or SQL Developer, and then log in to the metadata database as `RASYS`.
2. Run the `DBMS_RA.UPDATE_PROTECTION_POLICY` procedure.

   For example, execute the following PL/SQL anonymous block:

```
BEGIN
 DBMS_RA.UPDATE_PROTECTION_POLICY(
  protection_policy_name => 'bronze_dev',
  recovery_window_goal   => INTERVAL '6' DAY);
END;
```

3. Optionally, query the recovery catalog to confirm the update of the policy.

   For example, query RA_PROTECTION_POLICY as follows (sample output included):

```
COL POLICY_NAME FORMAT a11
COL DESCRIPTION FORMAT a36
SELECT POLICY_NAME, DESCRIPTION,
       TO_CHAR(EXTRACT(DAY FROM RECOVERY_WINDOW_GOAL),'fm00')    ||':'||
       TO_CHAR(EXTRACT(HOUR FROM RECOVERY_WINDOW_GOAL),'fm00')   ||':'||
       TO_CHAR(EXTRACT(MINUTE FROM RECOVERY_WINDOW_GOAL),'fm00') ||':'||
       TO_CHAR(EXTRACT(SECOND FROM RECOVERY_WINDOW_GOAL),'fm00')
         AS "DD:HH:MM:SS"
FROM   RA_PROTECTION_POLICY
WHERE  POLICY_NAME='BRONZE_DEV';

POLICY_NAME DESCRIPTION                          DD:HH:MM:SS
----------- ------------------------------------ --------------
BRONZE_DEV  For protected dbs in bronze tier     06:00:00:00
```

Deleting a Protection Policy

This section explains how to delete protection policies using either Cloud Control (recommended) or the DBMS_RA PL/SQL package.

Deleting a Protection Policy Using Cloud Control

This section describes how to delete a protection policy from the Protection Policy page in Cloud Control.

Prerequisites

You must be logged in to the Recovery Appliance as RASYS.

Assumptions

Assume that you created the bronze_dev policy as described in "Creating a Protection Policy Using Cloud Control". You want to delete this policy.

To delete a protection policy:

1. Access the Protection Policies page, as described in "Accessing the Create Protection Policy Page in Cloud Control".
2. In the Protection Policies table, select the protection policy that you want to delete.

   For example, select the BRONZE_DEV row.
3. Click **Delete**.

   A confirmation window appears.
4. Click **Yes**.

   The Protection Policies page appears, with the deleted protection policy no longer listed.

> **See Also:**
>
> Cloud Control online help for more information about the Protection Policies page

Deleting a Protection Policy Using DBMS_RA

To delete a protection policy, execute the `DBMS_RA.DELETE_PROTECTION_POLICY` procedure.

Prerequisites

You must meet the following prerequisites:

- You must log in to the metadata database as `RASYS`.
- The protection policy must not be associated with any protected database. To delete a policy that is associated with one or more databases, you must associate those databases with different policies before you can delete the desired policy.

Assumptions

Assume that you want to delete the protection policy named `test_dev` that you created in "Creating a Protection Policy".

To delete a protection policy:

1. Start SQL*Plus or SQL Developer, and then log in to the metadata database as `RASYS`.
2. Confirm that the protection policy that you intend to delete is not currently associated with any protected databases.

   For example, query all protection policies not associated with databases:

   ```
   SELECT POLICY_NAME AS "Currently unused policy"
   FROM   RA_PROTECTION_POLICY
   WHERE  POLICY_NAME NOT IN (SELECT POLICY_NAME FROM RA_DATABASE)
   ORDER BY POLICY_NAME;

   Currently unused policy
   ----------------------
   TEST_DEV
   ```

3. Delete the policy.

   For example, execute the following PL/SQL anonymous block to delete the protection policy named `test_dev`:

   ```
   BEGIN
     DBMS_RA.DELETE_PROTECTION_POLICY(
       protection_policy_name => 'test_dev');
   END;
   ```

4. Optionally, confirm the deletion.

   For example, count the rows for policies named `test_dev` (sample output included):

```
SELECT COUNT(*)
FROM   RA_PROTECTION_POLICY
WHERE  POLICY_NAME = 'TEST_DEV';

  COUNT(*)
----------
         0
```

6

Configuring Recovery Appliance for Protected Database Access

This chapter contains the following topics:

About Protected Database Access

This section contains the following topics:

> **See Also:**
>
> Recovery Appliance Architecture

Purpose of Protected Database Access

A database is not protected by a Recovery Appliance until it can access the database backups.

Overview of Protected Database Access

Performing necessary configuration so that a protected database can send backups to Recovery Appliance is called enrolling a database. Enrolling is a one-time task that must be performed the first time you set up a protected database to use Recovery Appliance. This task requires configuration on both the Recovery Appliance and the protected database.

The basic enrollment steps are as follows:

1. Adding the database

   The process of adding a database to a Recovery Appliance adds metadata for the database to the Recovery Appliance metadata database, and assigns this

database to the specified protection policy. The result of running `DBMS_RA.ADD_DB` is that a non-protected database attains the status of a protected database.

2. Granting access to the database to a Recovery Appliance user account

   After you create a virtual private catalog account (the Recovery Appliance user) in the metadata database, run `DBMS_RA.GRANT_DB_ACCESS` on the Recovery Appliance to associate this account with the protected database.

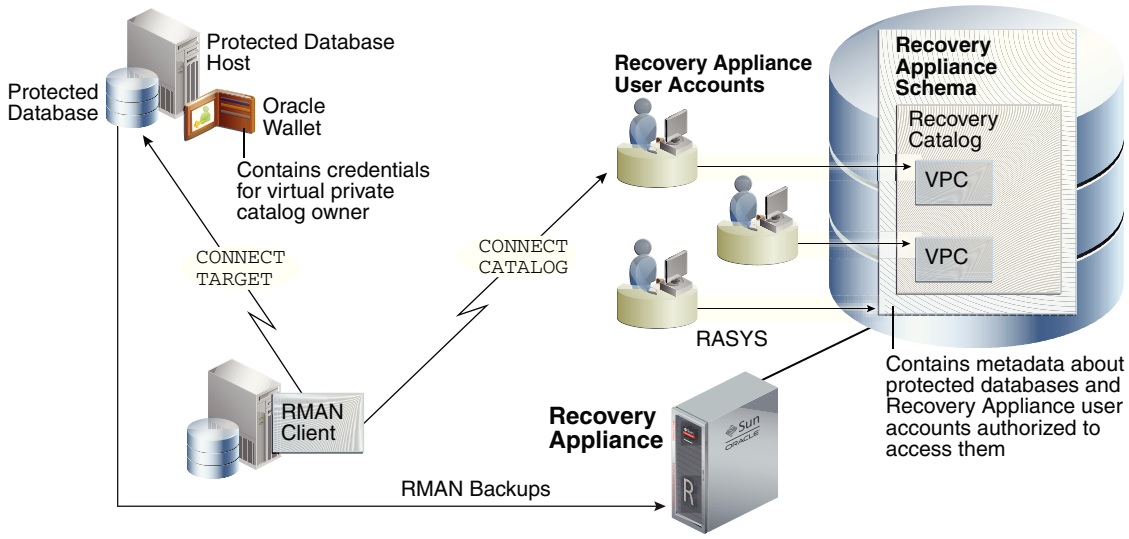
3. Registering the database with the virtual private catalog

   On the protected database host, create an Oracle wallet, and then add the credentials of the virtual private catalog account. Register the protected database with the recovery catalog using the RMAN `REGISTER DATABASE` command.

> **Note:**
>
> If you choose to configure real-time redo transport, then you must execute several SQL statements on the protected database (see *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*).

Figure 6-1 shows an RMAN client connecting to a protected database (`CONNECT TARGET`) and to the virtual private catalog (`CONNECT CATALOG`). For backup and restore operations to be possible, the credentials for the virtual private catalog owner must exist in the Oracle wallet on the protected database host.

Figure 6-1 Protected Database Access



It is possible for a database to store metadata in the Recovery Appliance catalog *without* backing up files to Recovery Appliance. In this case, the databases do not have the status of protected databases, and thus are not enrolled with Recovery Appliance. Future enrolling of such databases is simplified because the virtual private catalog owner already exists, and thus does not need to be created.

User Interfaces for Configuring Protected Database Access

This section contains the following topics:

- Accessing the Protected Databases Page in Cloud Control
- DBMS_RA Procedures Relating to Protected Database Access
- Recovery Catalog Views for Protected Database Access

Accessing the Protected Databases Page in Cloud Control

The Protected Databases page in Oracle Enterprise Manager Cloud Control (Cloud Control) is the recommended interface for starting the database enrollment process.

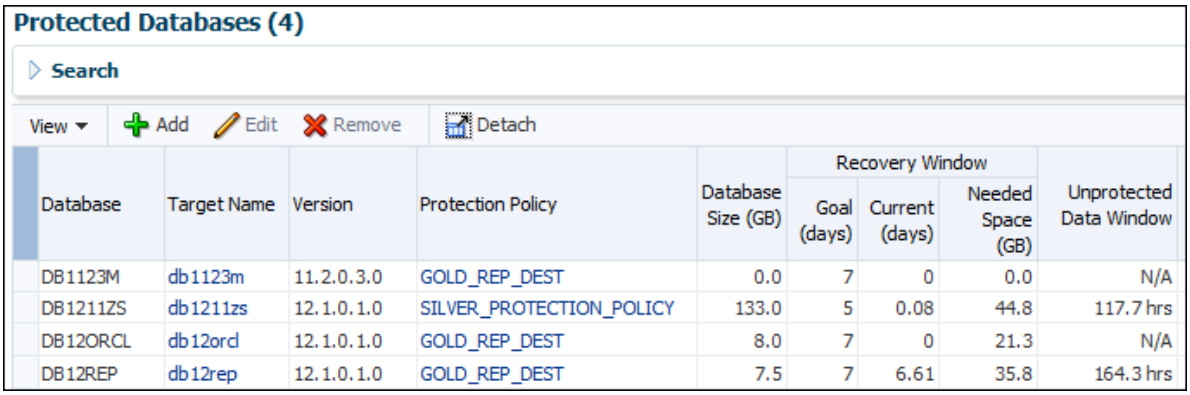
The Protected Databases page lists all databases under the management of this Recovery Appliance, whether they back up directly to the Recovery Appliance or are configured for downstream Recovery Appliance replication. From this page, you can add protected databases by selecting an individual database, selecting multiple databases, or selecting a previously defined Enterprise Manager group.

To access the Protected Databases page:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. From the **Recovery Appliance** menu, select **Protected Databases**.

   The Protected Databases page appears, as shown in Figure 6-2.

Figure 6-2 Protected Databases Page



Database	Target Name	Version	Protection Policy	Database Size (GB)	Recovery Window			Unprotected Data Window
					Goal (days)	Current (days)	Needed Space (GB)	
DB1123M	db1123m	11.2.0.3.0	GOLD_REP_DEST	0.0	7	0	0.0	N/A
DB1211ZS	db1211zs	12.1.0.1.0	SILVER_PROTECTION_POLICY	133.0	5	0.08	44.8	117.7 hrs
DB12ORCL	db12ord	12.1.0.1.0	GOLD_REP_DEST	8.0	7	0	21.3	N/A
DB12REP	db12rep	12.1.0.1.0	GOLD_REP_DEST	7.5	7	6.61	35.8	164.3 hrs

See Also:

- "About Recovery Appliance Replication"
- Cloud Control online help for more information about the Protected Databases page

DBMS_RA Procedures Relating to Protected Database Access

You can use the `DBMS_RA` package to configure protected database access. Table 6-1 describes the principal program units relating to protected databases.

Table 6-1 DBMS_RA Protected Database Access Procedures

Program Unit	Description
ADD_DB	Adds metadata for the specified database to Recovery Appliance, and assigns a protection policy to the database. Note that you must set the `reserved_space` parameter.
DELETE_DB	Removes metadata for the specified database from Recovery Appliance. All metadata and backups of this database are deleted, from both disk and SBT.
GRANT_DB_ACCESS	Grants Recovery Appliance privileges to a user for a specified database.
REVOKE_DB_ACCESS	Revokes Recovery Appliance privileges from a user for a specified database.
UPDATE_PROTECTION_POLICY	Modifies the parameters for an existing protection policy.

> **See Also:**
>
> DBMS_RA Package Reference

Recovery Catalog Views for Protected Database Access

You can monitor database access using the Recovery Appliance catalog views. Table 6-2 summarizes the most relevant views.

Table 6-2 Recovery Catalog Views for Protected Database Access

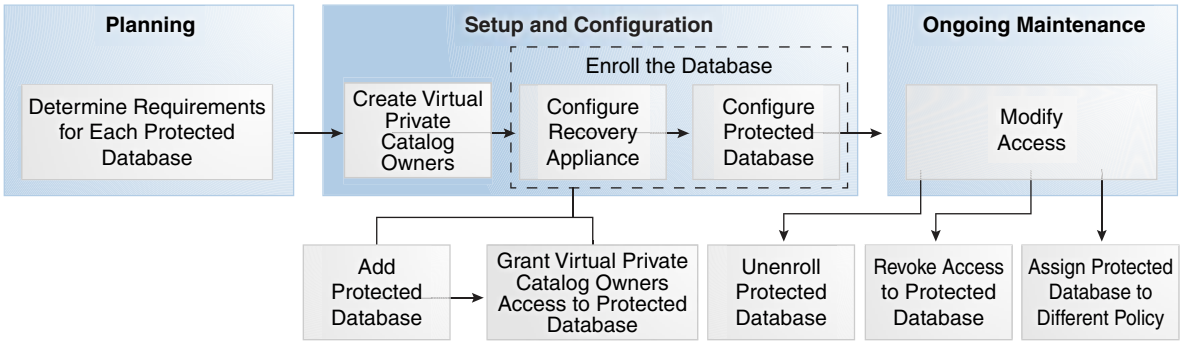
View	Description
RA_DATABASE	This view describes databases protected by this Recovery Appliance.
RA_DB_ACCESS	This view describes the user account that can access specific protected databases.

> **See Also:**
>
> Recovery Appliance View Reference

Basic Tasks for Configuring Protected Database Access

This section explains the basic tasks involved in configuring protected database access. Figure 6-3 shows the overall workflow described in Recovery Appliance Workflow , with the configuration tasks on the Recovery Appliance highlighted.

Figure 6-3 Database Access Configuration Tasks in the Recovery Appliance Workflow



Typically, you configure protected database access in the following sequence:

1. During the planning phase, decide which databases will be protected by the Recovery Appliance.

   "Task 4: Determine access requirements for Recovery Appliance" describes this task.

2. During the configuration phase (see "Setup and Configuration for Recovery Appliance"), do the following:

   a. Create virtual private catalog accounts.

      "Creating Virtual Private Catalog Accounts" describes this task.

   b. Enroll the protected database with the Recovery Appliance.

      > **Note:**
      >
      > With Cloud Control, you can perform all enrollment steps in a single page *except* registering the database in the recovery catalog.

      "Enrolling Protected Databases" describes this task.

3. During the ongoing maintenance phase (see "Maintenance Tasks for Recovery Appliance"), you can do the following:

   - Update the properties of an existing protected database using `DBMS_RA.UPDATE_DB` (see "Updating Protected Database Properties")
   - Remove metadata for protected databases from the Recovery Appliance using `DBMS_RA.DELETE_DB`

- Revoke access to a specific protected database from a specific virtual private catalog owner by using `DBMS_RA.REVOKE_DB_ACCESS`

Creating Virtual Private Catalog Accounts

RMAN must connect to the Recovery Appliance catalog when backing up to a Recovery Appliance. In this step, you create a virtual private catalog user for a specific protected database or set of protected databases.

Prerequisites

Log in to the metadata database as `SYSTEM`.

Assumptions

Assume that you are a Recovery Appliance administrator with the following requirements:

- You want to enroll database `orcld` with a Recovery Appliance.
- You want to create a virtual private catalog account named `ravpc1`. When backing up `orcld`, you plan to run `CONNECT CATALOG` with the `ravpc1` credentials.

To create a virtual private catalog account:

1. Log in to the Recovery Appliance as `root`.
2. Change to the `bin` directory:

   ```
   # cd /opt/oracle.RecoveryAppliance/bin
   ```

3. Run the command to add the new virtual private catalog account.

   The following command adds a virtual private catalog account named `ravpc1`:

   ```
   # ./racli add vpc_user -username=ravpc1
   ```

   When prompted, enter the password for the `ravpc1` user.

> **See Also:**
>
> - *Oracle Database Security Guide* to learn how to create database user accounts
> - *Oracle Database 2 Day + Security Guide* to learn how to create database user accounts using Cloud Control
> - *Oracle Database Backup and Recovery User's Guide* to learn about virtual private catalogs

Enrolling Protected Databases

This section explains how to enroll a protected database using either Cloud Control (recommended) or the `DBMS_RA` command-line interface.

> **See Also:**
>
> My Oracle Support Note Doc ID 1995866.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=1995866.1`) for main prerequisites for enrolling a database with Recovery Appliance

Enrolling Protected Databases Using Cloud Control

This section describes how to start the database enrollment process from the Protected Databases page in Cloud Control.

Prerequisites

The databases to be enrolled with Recovery Appliance must already be discovered as Database Instance targets by Cloud Control.

Assumptions

Assume that you have the following business requirements:

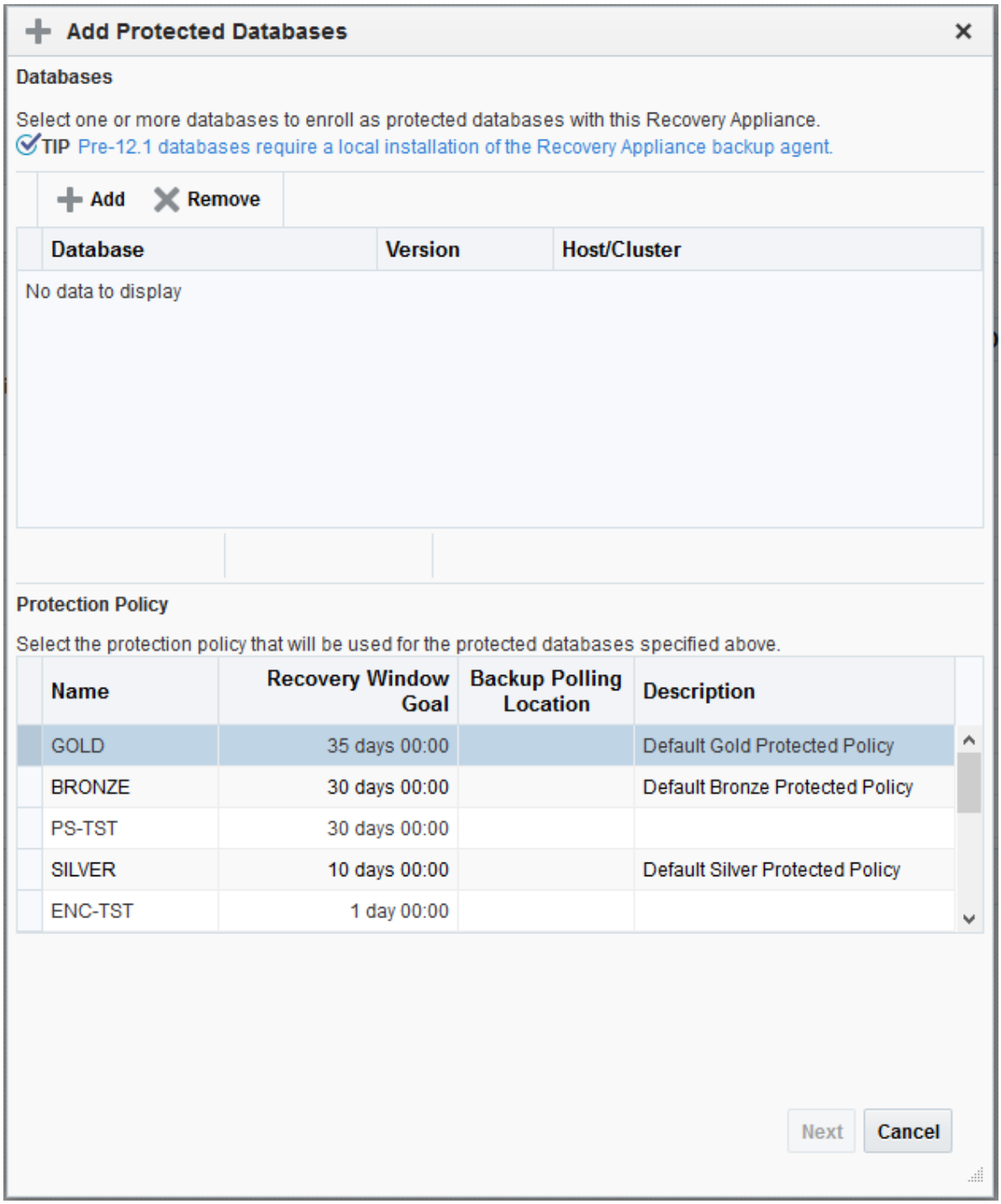
- You want to enroll databases `ORCL11` and `ORCL12`.
- You want to assign these databases to the protection policy named `GOLD`.
- You want each of the newly enrolled databases to have 6355 GB of reserved space (the amount of disk space guaranteed to each protected database).

To enroll protected databases:

1. Access the Protected Databases page, as described in "Accessing the Protected Databases Page in Cloud Control".
2. Click **Add**.

   The Add Protected Databases page appears.

Figure 6-4 Add Protected Databases Page



3. Click **Add**

   The Select Targets page appears.

4. In **Target Type**, select **Database Instance**.

   The page refreshes to list only database instances.

5. Optionally, narrow the database instances by entering values in the **Target Name** and **On Host** fields.

In this example, leave the fields blank so that you can multi-select databases in the next step.

6. In the table of targets, click the desired databases while pressing the Ctrl key.

   For example, from the target list, select `ORCL11` and `ORCL12`.

7. Click **Select**.

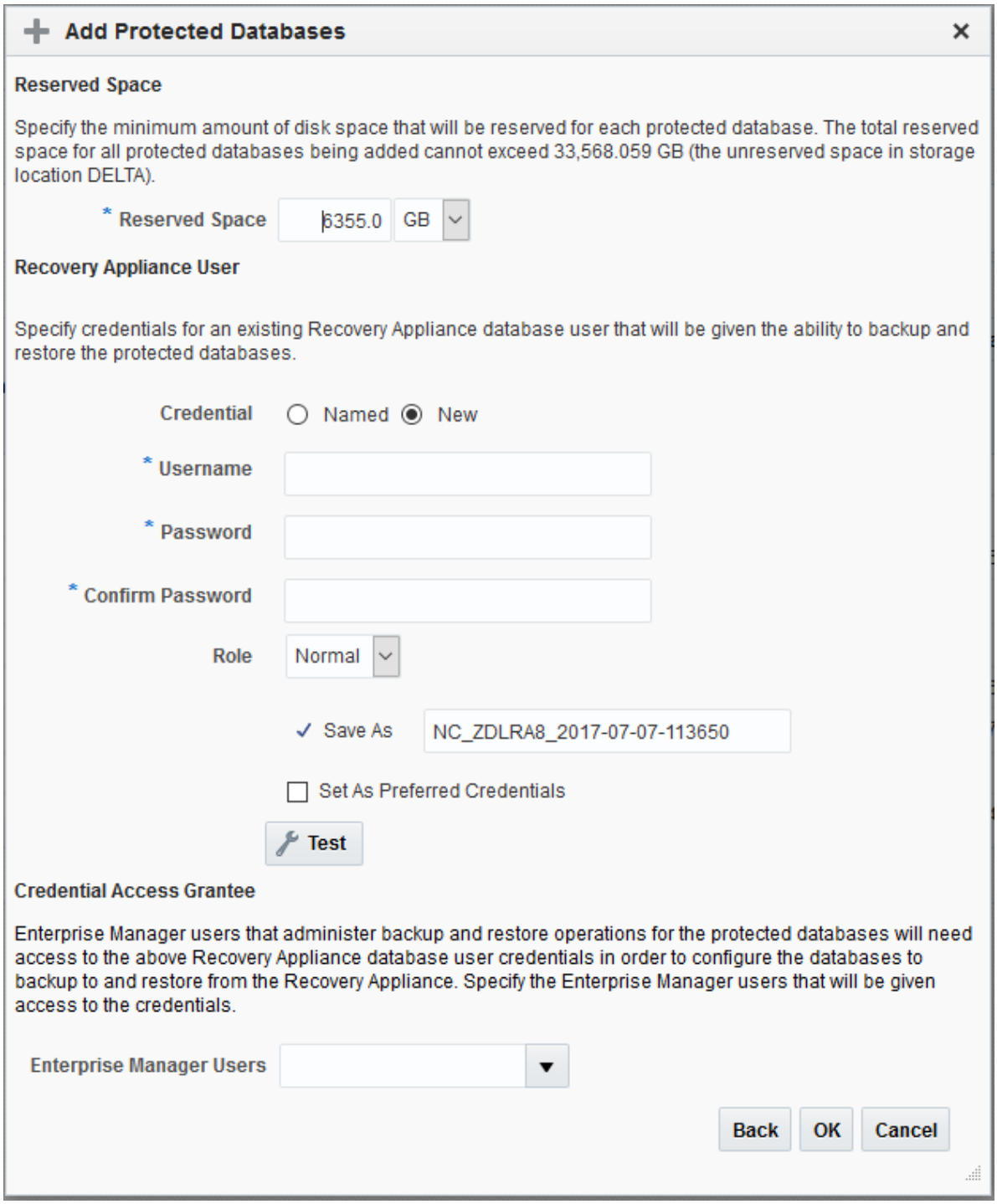
   The Add Protected Database page appears, listing the databases to be enrolled.

8. In the Protection Policy section, click the policy to which you want to add the databases, and then click **Next**.

   For example, click `GOLD`, and then click **Next**.

   The Add Protected Databases page appears.

Figure 6-5 Add Protected Databases Page



9. Set required attributes of the protected database:
   - In the **Reserved Space** field, enter the minimum amount of disk space to be reserved for each protected database.

> **Note:**
>
> When you add a database to a Recovery Appliance using Cloud Control, the Recovery Appliance allocates a default reserved space of 2.5X the database size. You can accept or change this amount.

For example, enter `6355`, and then select **GB** for the units.

- In the Recovery Appliance User section, enter the credentials for the appropriate virtual private catalog account.
- In the Credential Access Grantee section, in **Enterprise Manager Users**, select the Enterprise Manager user accounts that need access to the Recovery Appliance user credentials.

  For example, select **All**.

10. Click **OK**.

    A confirmation window appears.

11. Click **Close** to return to the Protected Databases page.

    The newly added databases appear in the table of protected databases.

    At this stage, the databases have been added and granted access, but not yet registered in the virtual private catalog.

12. See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn how to complete the database enrollment.

> **See Also:**
>
> Cloud Control online help for more information about the Add Protected Databases page

Enrolling Protected Databases Using the Command Line

When enrolling databases using the `DBMS_RA` command-line interface, you must perform the following tasks:

1. "Adding Protected Database Metadata Using DBMS_RA"
2. "Granting Database Access to a Recovery Appliance Account Using DBMS_RA"
3. Configuring the protected database for access (see *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*)

Adding Protected Database Metadata Using DBMS_RA

For a database to be protected, you must add metadata for this database to the Recovery Appliance using `DBMS_RA.ADD_DB`. This procedure requires you to specify an existing protection policy and the amount of reserved space for the database.

Prerequisites

You must log in to the Recovery Appliance with the RASYS account.

Assumptions

Assume that you are a Recovery Appliance administrator with the following requirements:

- You want to make orcld a protected database.
- You want to add this database to the existing bronze protection policy, and provide it with 200 GB of reserved space.

To add metadata for a protected database to the Recovery Appliance:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as RASYS.
2. Use the ADD_DB procedure to add database metadata to the Recovery Appliance and assign a protection policy.

   For example, the following anonymous block adds database orcld:

   ```
   BEGIN
     DBMS_RA.ADD_DB (
       db_unique_name         => 'orcld',
       protection_policy_name => 'bronze',
       reserved_space         => '200G');
   END;
   ```

3. Optionally, query the recovery catalog to see information about the newly added database.

   For example, execute the following query to show details about orcld (sample output included):

   ```
   COLUMN PROT_DB FORMAT a10
   COLUMN POLICY_NAME FORMAT a11
   SELECT DB_UNIQUE_NAME AS PROT_DB, DB_KEY, DBID, POLICY_NAME
   FROM   RA_DATABASE
   WHERE  DB_UNIQUE_NAME = 'ORCLD';

   PROT_DB        DB_KEY       DBID POLICY_NAME
   ---------- ---------- ---------- -----------
   ORCLD             301 3210984255 BRONZE
   ```

> **Note:**
>
> In an Oracle Data Guard environment, add the db_unique_name of whichever database (primary or standby) that you registered with the Recovery Appliance catalog.

> **See Also:**
>
> "ADD_DB"

Granting Database Access to a Recovery Appliance Account Using DBMS_RA

You must grant the necessary privileges to a Recovery Appliance user account—which is also a virtual private catalog account—so that protected databases that authenticate with this account can perform backup and restore operations. The `DBMS_RA.GRANT_DB_ACCESS` procedure associates a protected database with a virtual private catalog.

Prerequisites

This task has the following prerequisites:

- You must log in to the Recovery Appliance with the `RASYS` account.
- The the Recovery Appliance user account specified in `DBMS_RA.GRANT_DB_ACCESS` must exist.
- You must have already added the protected database named `orcld`.

Assumptions

Assume that you want to enable RMAN to `CONNECT CATALOG` as `ravpc1` when backing up protected database `orcld`.

To grant access to a virtual private catalog account to a protected database:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance database as `RASYS`.
2. Run the `GRANT_DB_ACCESS` procedure to grant backup and restore privileges on the database for the user.

   The following PL/SQL anonymous block grants access to protected database `orcld` to virtual private catalog account `ravpc1`:

   ```
   BEGIN
      DBMS_RA.GRANT_DB_ACCESS (
        db_unique_name =>  'orcld',
        username       =>  'ravpc1');
   END;
   ```

3. Optionally, query the recovery catalog to see information about the database access.

   For example, execute the following query to show details about `orcld` and catalog owner `ravpc1` (sample output included):

   ```
   COLUMN PROT_DB FORMAT a10
   COLUMN POLICY_NAME FORMAT a11
   COLUMN USERNAME FORMAT a15
   COLUMN DB_KEY FORMAT 999999
   SELECT d.DB_UNIQUE_NAME AS PROT_DB, d.DB_KEY,
          d.DBID, d.POLICY_NAME, a.USERNAME
   FROM   RA_DATABASE d, RA_DB_ACCESS a
   ```

```
WHERE  d.DB_UNIQUE_NAME = 'ORCLD'
AND    a.DB_KEY = d.DB_KEY;

PROT_DB     DB_KEY       DBID POLICY_NAME USERNAME
---------- ------- ---------- ----------- ---------------
ORCLD          301 3210984255 BRONZE      RAVPC1
```

4. Send the virtual private catalog user name and password to the DBA for each protected database that must authenticate using this account.
5. To complete the enrollment procedure, see *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*.

> **See Also:**
>
> "GRANT_DB_ACCESS"

Updating Protected Database Properties

This section explains how to update protected database properties using either Cloud Control (recommended) or the `DBMS_RA` command-line interface.

Updating Protected Database Properties Using Cloud Control

This section describes how to edit a database from the Protected Databases page in Cloud Control.

Assumptions

Assume that you have the following business requirements:

- You want to change the protection policy for protected database `ORCL11` from `GOLD` to `BRONZE`.
- You want change the reserved space from 6355 GB to 7000 GB.
- You want to change the Recovery Appliance user account associated with this protected database from `rauser11` to `rauser12`.

To update the properties of a protected database:

1. Access the Protected Databases page, as described in "Accessing the Protected Databases Page in Cloud Control".
2. Click **Edit**.

   The Edit Protected Databases page appears.
3. Change the desired attributes of the protected database, and then click **OK**:
   - In the Protection Policy section, select the row for the policy named `BRONZE`.

     For example, select **All**.
   - In the **Reserved Space** field, enter the new minimum amount of disk space to be reserved for this protected database.

     For example, enter `7000`, and then select **GB** for the units.

- In the Recovery Appliance User section, enter the credentials for the database user `rauser12`.

The newly updated database appears in the table of protected databases.

> **See Also:**
>
> Cloud Control online help for more information about the Edit Protected Databases page

Assigning a Database to a Different Protection Policy Using DBMS_RA

To update the properties of a protected database, use the `DBMS_RA.UPDATE_DB` procedure. Unspecified parameters retain their existing values. This section shows how to update a protected database to use a different protection policy.

Prerequisites

You must log in to the metadata database as `RASYS`.

Assumptions

This tutorial assumes that the existence of the protection policy named `bronze` that you created in "Creating a Protection Policy Using DBMS_RA". Your goal is to change the protection policy for database `zdlrac` from `silver` to `bronze`.

To assign a database to a different protection policy:

1. Start SQL*Plus or SQL Developer, and then log in to the metadata database as `RASYS`.
2. Query the existing protection policies.

   For example, execute the following query (sample output included):

```
COL POLICY_NAME FORMAT a11
COL DESCRIPTION FORMAT a35
SELECT POLICY_NAME, DESCRIPTION,
       TO_CHAR(EXTRACT(DAY FROM RECOVERY_WINDOW_GOAL),'fm00')||':'||
       TO_CHAR(EXTRACT(HOUR FROM RECOVERY_WINDOW_GOAL),'fm00')||':'||
       TO_CHAR(EXTRACT(MINUTE FROM RECOVERY_WINDOW_GOAL),'fm00')||':'||
       TO_CHAR(EXTRACT(SECOND FROM RECOVERY_WINDOW_GOAL),'fm00')
         AS "DD:HH:MM:SS"
FROM   RA_PROTECTION_POLICY;

POLICY_NAME DESCRIPTION                         DD:HH:MM:SS
----------- ----------------------------------- --------------
BRONZE      For protected dbs in bronze tier    01:00:00:00
SILVER      For protected dbs in silver tier    07:00:00:00
GOLD        For protected dbs in gold tier      14:00:00:00
```

3. Determine which protected databases are associated with which protection policies.

   For example, execute the following query (sample output included):

```
SELECT d.DB_UNIQUE_NAME, d.POLICY_NAME
FROM   RA_PROTECTION_POLICY p, RA_DATABASE d
WHERE  p.policy_name=d.policy_name
ORDER BY d.DB_UNIQUE_NAME;

DB_UNIQUE_NAME                 POLICY_NAME
------------------------------ -----------
ZDLRA                          BRONZE
ZDLRAC                         SILVER
.
.
.
```

4. Run the `DBMS_RA.UPDATE_DB` procedure to associate a database with a new policy.

   For example, execute the following PL/SQL anonymous block to associate the database named `zdlrac`, which has `silver` as its current policy, with the protection policy named `bronze`:

```
BEGIN
 DBMS_RA.UPDATE_DB(
   db_unique_name            => 'zdlrac',
   protection_policy_name    => 'bronze');
END;
```

5. Optionally, confirm that the database is associated with the correct policy.

   For example, execute the following query (sample output included):

```
SELECT d.DB_UNIQUE_NAME, d.POLICY_NAME
FROM   RA_PROTECTION_POLICY p, RA_DATABASE d
WHERE  p.POLICY_NAME=d.POLICY_NAME
ORDER BY d.DB_UNIQUE_NAME;

DB_UNIQUE_NAME                 POLICY_NAME
------------------------------ -----------
ZDLRA                          BRONZE
ZDLRAC                         BRONZE
.
.
.
```

> **See Also:**
>
> "UPDATE_DB"

7

Copying Backups to Tape with Recovery Appliance

This chapter explains how to copy completed backups to tape to ensure optimal utilization of space and storage resources on Zero Data Loss Recovery Appliance.

This chapter contains the following topics:

About Copying Backups to Tape with Recovery Appliance

This section contains the following topics:

Purpose of Copying Backups to Tape with Recovery Appliance

A robust backup strategy protects data against intentional attacks, unintentional user errors (such as file deletions), and software or hardware malfunctions. Tape libraries provide effective protection against these possibilities.

The advantages of the Recovery Appliance tape solution are as follows:

- Tape backups are ideal for long-term storage. Tapes are portable and easy to store for lengthy periods of time.
- All tape backup operations are performed by the Recovery Appliance, with no performance load on the protected database host.
- Tape backups are optimized. Recovery Appliance intelligently gathers the necessary blocks to create a virtual, full backup or incremental backup for tape. Although Recovery Appliance backups are incremental forever, you can create a flexible backup strategy to tape, such as weekly full and daily incremental or just daily full backups.

- Oracle Secure Backup is pre-installed, eliminating the need for third-party media managers.
- Tape drives and tape libraries function more efficiently because Recovery Appliance is a single large centralized system with complete control over them. In other tape solutions, hundreds or thousands of databases can compete for tape resources in an uncoordinated manner.

> **See Also:**
>
> "Autonomous Tape Archival"

Overview of Copying Backups to Tape with Recovery Appliance

This section contains the following topics:

- About Tape Operations on Recovery Appliance
- Recovery Appliance Components for Managing Tape Operations
- Backup Retention on Tape
- About Pausing and Resuming Tape Backup Operations

About Tape Operations on Recovery Appliance

All backups that Recovery Appliance receives from protected databases are always first stored on disk. The Recovery Appliance can then optionally copy these backups to tape. All copying to tape is automated, policy-driven, and scheduled.

A protection policy defines desired recovery windows for backups stored on tape. Recovery windows are expressed as time intervals, such as 30 days. Backups are retained on tape long enough for a recovery to be possible at any time within this interval, counting backward from the current time.

You can copy Recovery Appliance backups from disk to tape. To perform this task, you must create a tape backup job that defines the properties of the copy operation, such as the media manager library and attribute set that will manage this job, the protection policy or the database for which the backups need to be copied, and so on. After you have defined the job properties, you must schedule this job to run.

Note:

- Only backups that have not already been copied to tape are processed in a tape backup operation for each tape backup job template with which the backup is associated. Thus, a tape backup operation on the same backup after the initial tape copy has no effect. In addition, only the most recent backup is copied to tape when the tape backup operation runs.

  If you require more than one copy of the same backup, such as to a different media family on tape, use the `COPIES` parameter of the template or create a separate tape backup job template for the additional copy.

- Long-term archival backups that were created with the `KEEP` option of the `BACKUP` command are never automatically copied to tape. You must manually copy them using the COPY_BACKUP or MOVE_BACKUP procedure.

  See My Oracle Support Note Doc ID 2107079.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=2107079.1`) to learn how to create archival backups for long term retention on the Recovery Appliance

During a restore, Recovery Appliance transparently retrieves the backup from tape.

Recovery Appliance writes backups to tape in formats supported by RMAN. If a protected database has the required media management software (for example, Oracle Secure Backup), then it can directly restore backups written to tape by the Recovery Appliance.

Recovery Appliance Components for Managing Tape Operations

Every database in a Recovery Appliance setup is associated with a protection policy that specifies the parameters for backup storage and recovery window goal. To manage and control tape operations, you must create a job that uses the properties defined by the selected protection policy, media manager library, and attribute set to copy backups to tape. Oracle Secure Backup and its components (media manager library and attribute sets) are preconfigured with Recovery Appliance.

Table 7-1 summarizes the roles of the Recovery Appliance objects for managing tape operations.

Table 7-1 Recovery Appliance Objects for Copying Backups to Tape

Cloud Control Object Name	Command-line Object Name	Description
Protection policy	Protection policy	Among other attributes, defines the recovery window. This recovery window is applied to all protected databases assigned to the protection policy.
Media manager library	SBT library	Describes a media management software library installed on Recovery Appliance.

Table 7-1 (Cont.) Recovery Appliance Objects for Copying Backups to Tape

Cloud Control Object Name	Command-line Object Name	Description
Media manager attribute set	SBT attribute set	Contains a collection of attributes that control the copy operation. One attribute specifies the library to be used in the copy operation. Other attributes are optional and can include channel parameters, media management software library–specific commands, and a media pool identifier. You can define multiple attribute sets, but only one attribute set is associated with a given copy job.
Copy-to-tape job template	SBT job template	Defines the properties of backups to be copied to tape and specifies an attribute set to control the copy operation. Typically, multiple job templates are defined as described in "Example: Creating a Tape Backup Job Using Cloud Control".

> **Note:**
>
> The Oracle Enterprise Manager Cloud Control (Cloud Control) object name and the command-line name in Table 7-1 refer to the same tape backup objects, with their respective interface terms.

> **See Also:**
>
> "Basic Tasks for Copying Backups to Tape with Recovery Appliance" for more information on how to create a tape backup job using these components

Backup Retention on Tape

You can control the length of time that backup copies are retained on tape by specifying a recovery window. A recovery window defines how long the Recovery Appliance maintains tape backups in its catalog for recovery purposes. The recovery window is expressed as an interval, in values of hours, days, weeks, or months. Backups are retained long enough to guarantee that a recovery is possible to any point in time within this interval, counting backward from the current time.

> **Note:**
>
> Recovery windows directly apply only to full or level 0 data file and control file backups.

Recovery Appliance does not purge tape backups. Instead, it informs the media manager which pieces are no longer needed for RMAN retention. With Oracle Secure Backup as the media manager, it does not explicitly delete these files, it updates its

catalog. After all files on a given tape are no longer needed, Oracle Secure Backup considers the tape for reuse.

You set the recovery window for a backup by providing a value for it in the protection policy. If this attribute is `NULL`, then Recovery Appliance never purges the backup from tape.

> **See Also:**
>
> - "CREATE_SBT_JOB_TEMPLATE"
> - "Creating a Protection Policy Using Cloud Control" for more information on setting recovery window goals using Cloud Control
> - "Creating a Protection Policy Using DBMS_RA" for more information on setting an SBT recovery window using command-line
> - *Oracle Database Backup and Recovery User's Guide* for a thorough discussion of recovery windows

About Pausing and Resuming Tape Backup Operations

You might want to pause the copying of backups to tape for these reasons:

- To investigate previous backup copy failures
- To perform maintenance operations on tape devices

You pause tape backup operations for a specific media management software library by pausing its corresponding media manager library.

When you pause a media manager library, in-progress copies of backup pieces are allowed to complete, while backup pieces that were queued for copy but not yet copied are held until you resume the library. Pausing a library suspends future scheduled runs of tape backup jobs that reference the media manager library.

> **Note:**
>
> Tape backup jobs reference media manager libraries indirectly through their assigned media manager attribute sets.

> **See Also:**
>
> - "Pausing and Resuming Media Manager Library Operations Using Cloud Control"
> - "Pausing and Resuming the SBT Library Using DBMS_RA"

About Using Oracle Secure Backup with Recovery Appliance

Oracle Secure Backup is a media manager that provides reliable, centralized tape management by protecting file-system data and Oracle Database files for multiple environments. Oracle Secure Backup is the tape management component for Recovery Appliance. It is installed with its components on the Recovery Appliance during its configuration.

Preconfigured Oracle Secure Backup components include the following:

Media Manager Library
During its installation, while Recovery Appliance is being configured, Oracle Secure Backup creates a media manager library with default parameters, such as the following:

- Library name (`ROBOT0`)
- Maximum number of accessible tape drives
- Number of restore drives
- Media manager location

Apart from the name, other media manager library parameters can be modified. This library manages the tape backup operations associated with it, based on the parameters set.

Media Manager Attribute Sets
Along with a media manager library, Oracle Secure Backup also comes installed with attribute sets for all tape drives that the default media manager library accesses. These attribute sets have default values for parameters like the media pool number and streams required for the copy operation. These and the media manager vendor parameters and commands can be modified. The default attribute sets are named `DRIVE_COUNT_1`, `DRIVE_COUNT_2`, `DRIVE_COUNT_3`, and so on for the number of tape drives accessed by the media manager library.

> **See Also:**
>
> - "Accessing the Oracle Secure Backup Domain Using Cloud Control" for more information on how to access the Oracle Secure Backup domain using Cloud Control
> - "Creating Tape Backup Job Components" for more information on how to create a media manager library and attribute sets for third-party media managers
> - "Managing Tape Backup Job Components" for more information on how to edit and control existing media manager components

User Interfaces for Recovery Appliance

You can manage and perform tape backup job operations by using either Cloud Control or the Recovery Appliance command-line options.

This section contains the following topics:

- Accessing Recovery Appliance in Cloud Control
- Accessing Recovery Appliance Using DBMS_RA

Accessing Recovery Appliance in Cloud Control

To access Recovery Appliance using Cloud Control, complete the steps listed in "Accessing the Recovery Appliance Home Page".

Accessing Recovery Appliance Using DBMS_RA

This section contains the following topics:

- DBMS_RA Procedures for Tape Backup Operations
- Recovery Catalog Views for Tape Operations

DBMS_RA Procedures for Tape Backup Operations

Table 7-2 lists the Recovery Appliance `DBMS_RA` procedures that are associated with SBT job operations:

Table 7-2 DBMS_RA Procedures Associated with Tape Backup Operations

SBT Object	Procedures
SBT Job	• CREATE_SBT_JOB_TEMPLATE • UPDATE_SBT_JOB_TEMPLATE • DELETE_SBT_JOB_TEMPLATE
SBT Library	• CREATE_SBT_LIBRARY • UPDATE_SBT_LIBRARY • PAUSE_SBT_LIBRARY • RESUME_SBT_LIBRARY • DELETE_SBT_LIBRARY
SBT Attribute Set	• CREATE_SBT_ATTRIBUTE_SET • UPDATE_SBT_ATTRIBUTE_SET • DELETE_SBT_ATTRIBUTE_SET
Protection Policy	• CREATE_PROTECTION_POLICY • UPDATE_PROTECTION_POLICY • DELETE_PROTECTION_POLICY
Backup	• QUEUE_SBT_BACKUP_TASK • COPY_BACKUP • MOVE_BACKUP

See Also:

"DBMS_RA Package Reference" for more information on other Recovery Appliance `DBMS_RA` procedures

Recovery Catalog Views for Tape Operations

This section lists the Recovery Appliance recovery catalog views that are associated with SBT job operations:

- RA_SBT_JOB
- RA_SBT_LIBRARY
- RA_SBT_ATTRIBUTE_SET
- RA_PROTECTION_POLICY
- RA_EM_SBT_JOB_TEMPLATE

> **See Also:**
>
> "Recovery Appliance View Reference" for more information on other Recovery Appliance data dictionary views

Basic Tasks for Copying Backups to Tape with Recovery Appliance

This section lists the high level essential steps to copy database backups to tape using Recovery Appliance.

> **See Also:**
>
> "About Copying Backups to Tape with Recovery Appliance"

To copy backups to tape using Recovery Appliance:

1. Create a media manager library for your media management software to manage all your tape backup jobs by adding parameters that apply to a set of jobs.

   Recovery Appliance uses Oracle Secure Backup as its media management software. During its setup, Recovery Appliance installs Oracle Secure Backup with a preconfigured media manager library and attribute sets.

   > **See Also:**
   >
   > "Creating a Media Manager Library" for more information on how to create additional media manager libraries

2. Create a media manager attribute set that helps you control your tape backup jobs further by adding more job-specific parameters and commands for your media manager software.

   Tape backup jobs use a combination of parameters specified at the media manager library level and the attribute set level while performing the copy

operation. Media manager libraries define parameters that apply to a set of jobs while attribute sets help further define tape backup settings for specified jobs.

Oracle Secure Backup also configures default attribute sets for all drives that are a part of the default media manager library.

> **See Also:**
>
> "Creating an Attribute Set" for more information on creating additional attribute sets

3. Create a tape backup job.

   The job definition includes job properties such as the media manager library and attribute set associated with this job, the type of backups that need to be copied to tape, the run-time window for this job, and so on.

   You can also schedule this job to run at a specified time according to your task requirements.

   > **See Also:**
   >
   > "Creating a Tape Backup Job" for more information on creating tape backup jobs

4. (Optional) If required, pause or resume a media manager library or a tape backup job.

   > **See Also:**
   >
   > "Pausing and Resuming Tape Backup Operations "

5. View the status of all media manager libraries and tape backup operations to check for errors.

   > **See Also:**
   >
   > "Viewing the Status of Tape Backup Operations"

You may want to create additional media families for refining backup properties or to schedule vaulting to manage your tape. Use the Recovery Appliance Oracle Secure Backup domain to complete these tasks.

See Also:

- "Accessing the Oracle Secure Backup Domain Using Cloud Control"
- *Oracle Secure Backup Administrator's Guide* for more information about configuring media families
- *Oracle Secure Backup Administrator's Guide* for more information on vaulting

Accessing the Oracle Secure Backup Domain Using Cloud Control

Use Cloud Control to access the Oracle Secure Backup domain. You can use this domain to manage (if required) the existing Oracle Secure Backup configurations set for the selected Recovery Appliance environment.

To access the Oracle Secure Backup domain using Cloud Control:

1. From the Cloud Control Home page, select **Targets**.
2. From the Targets Menu, select **All Targets**.

   The All Targets page appears.
3. On the All Targets page, under the Refine Search Menu, select **Databases** as the Target Type.
4. Under the Databases section, select **Oracle Secure Backup Domain**.

   A list of all Oracle Secure Backup Domains for all existing Recovery Appliance targets appears.
5. From the list of targets, click the target for which you want to access the Oracle Secure Backup domain.

   The Oracle Secure Backup domain for the selected Recovery Appliance appears.

See Also:

"About Using Oracle Secure Backup with Recovery Appliance"

Creating Tape Backup Job Components

To successfully create a tape backup job, you must first ensure that its components exist. A media manager library and its attribute sets are essential for a tape backup operation. These components define a combination of parameters for a tape backup job and help categorize these jobs when storing them on tape.

> **See Also:**
>
> "Recovery Appliance Components for Managing Tape Operations" for more information on the role of a media manager library and its attribute sets

This section contains the following topics:

- Creating a Media Manager Library
- Creating an Attribute Set

Creating a Media Manager Library

A media manager library sets the properties for a tape backup job by defining parameters like the number of tape drives it can access. Optional advanced parameters include specifying the number of required restore drives and media manager parameters.

Recovery Appliance comes with Oracle Secure Backup as its preconfigured media manager. During the Recovery Appliance configuration, a media manager library is also configured for Oracle Secure Backup, typically named `ROBOT0`. It is recommended that you only use the preconfigured media manager objects. The media manager (Oracle Secure Backup, in this case) must have only a single media manager library as more than one library object will result in a conflict between the tape backup jobs created and the media manager resources handling these jobs. Currently, you cannot install Oracle Secure Backup in the Recovery Appliance in `client only` .

If you are using third-party media management software, you must install its backup agent on the Recovery Appliance compute servers. To schedule a tape backup job using the third-party product, you must create a new media manager library and add RMAN parameters applicable for that media manager for backups over LAN to tape devices attached to the backup application's media servers. In this scenario, you will not be able to use the media manager components preconfigured for Oracle Secure Backup and cannot directly attach tape devices to the Recovery Appliance.

This section describes the steps to create an additional media manager library for a third-party media manager. It contains the following topics:

- Creating a Media Manager Library Using Cloud Control
- Creating an SBT Library Using DBMS_RA

Creating a Media Manager Library Using Cloud Control

A media manager library sets and manages the parameters for the media management software through which backups are copied to tape.

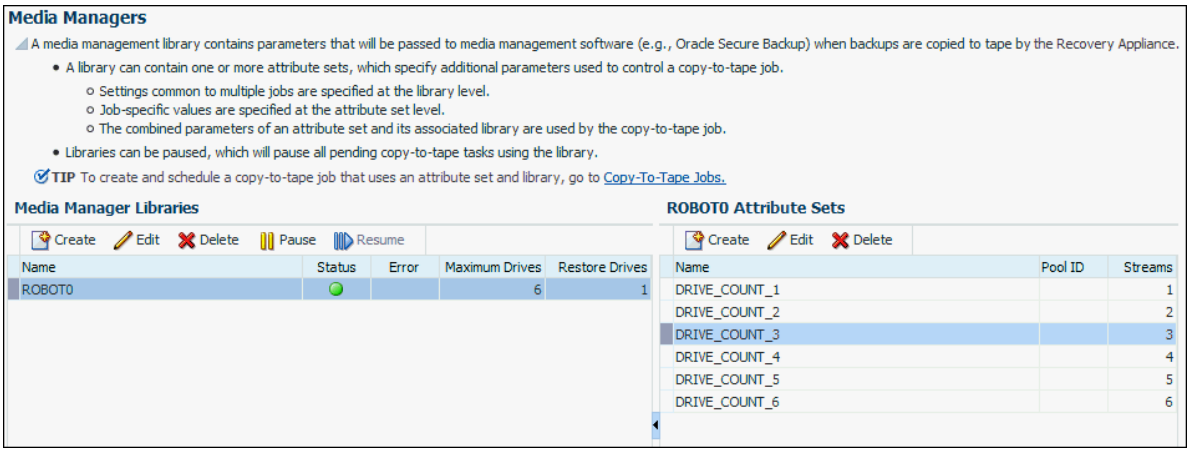
To create a media manager library:

1. Complete the steps in "Accessing the Recovery Appliance Home Page".

   On the Recovery Appliance Home page, select **Media Managers** from the Recovery Appliance Menu.

   Figure 7-1 displays the Media Managers screen with the default Oracle Secure Backup library and its corresponding attribute sets.

Figure 7-1 Media Managers Page



2. On the Media Managers page, click **Create** to configure a new media manager library.

   The Create Media Manager Library and Initial Attribute Set dialogue box appears.

3. In the **Media Manager Library** section, enter a name for this library.

4. In the **Maximum Drives** field, select the maximum number of tape drives that this media manager library can access.

5. Optionally, you can choose to enter **Advanced Parameters** for this media manager library.

   a. In the **Restore Drive** advanced parameter field, specify the number of drives that you want to use solely for restore operations. If you do not enter any restore drive value, then the current restore operation uses the first free drive that is available once all the backup operations are complete.

   b. In the **Media Management Vendor Parameters**, you can choose to add additional parameters to define your media manager library.

      For example, a media manager vendor parameter for Oracle Secure Backup contains the `SBT_LIBRARY` parameter by default, which specifies the path of the media manager library.

   If you are using a third party product as your media manager, create a new media library and use product-specific parameters for the specified media manager, especially the `SBT_LIBRARY` location parameter.

6. To add the initial attribute set for this library, complete the steps in the section "Creating an Attribute Set Using Cloud Control".

   If you do not enter any values for the attribute set, default values are applied.

7. Click **OK**.

> **See Also:**
>
> "Recovery Appliance Components for Managing Tape Operations"

Creating an SBT Library Using DBMS_RA

The SBT library object describes a media management software library installed on the Recovery Appliance. It includes parameters to pass to the media management software.

To create an SBT library:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the `DBMS_RA.CREATE_SBT_LIBRARY` procedure.

```
BEGIN
   DBMS_RA.CREATE_SBT_LIBRARY(
     lib_name       => 'osbsbt',
     drives         => 12,
     restore_drives => 2,
     parms          => 'SBT_LIBRARY=libobk.so');
END;
```

In this example, the media management software is Oracle Secure Backup. The drives argument specifies the maximum number of tape drives that this SBT library can access. The `restore_drives` argument sets the number of tape drives that will be reserved for restore operations. The `parms` argument has the same purpose and format as the `PARMS` clause of an RMAN `ALLOCATE CHANNEL` command. It typically includes at least the `SBT_LIBRARY` parameter. In this case it designates the shared library for the Oracle Secure Backup media family.

> **See Also:**
>
> "CREATE_SBT_LIBRARY" for descriptions of procedure arguments

Creating an Attribute Set

A SBT attribute set helps you to further customize and categorize your backups while copying them to tape. An attribute set is created for each tape drive associated with its media manager library. It helps classify backups while storing them on tape by specifying parameters such as the media pool number, streams, and media manager commands needed to perform the tape backup operation.

Recovery Appliance comes with Oracle Secure Backup preconfigured as its media manager. Oracle Secure Backup components which include a media manager library and attribute sets for each of its tape drives, are also preconfigured. The preconfigured attribute sets are typically named `DRIVE_COUNT_1`, `DRIVE_COUNT_2`, and so on for the number of existing tape drives.

This section describes how to create additional attribute sets for third-party media managers. It contains the following topics:

- Creating an Attribute Set Using Cloud Control
- Creating an SBT Attribute Set Using DBMS_RA

Creating an Attribute Set Using Cloud Control

You create the initial attribute set for a media manager library while creating the media manager library itself. If you leave the initial attribute set fields empty while configuring the media manager library, the default values are used.

> **See Also:**
>
> "Creating a Media Manager Library Using Cloud Control"

You can also use the following steps to create additional attribute sets for a third-party media manager library.

To create an attribute set:

1. Complete the steps in "Accessing Recovery Appliance in Cloud Control".

   From the Recovery Appliance menu, select **Media Managers**.
2. Under the Attribute Sets section, click **Create**.

   The Create Attribute Set box appears.
3. In the **Name** field, enter a name for this attribute set.
4. Optionally, in the **Pool ID** field, enter the media pool number that will be used to store backup copies.
5. Optionally, in the **Streams** field, specify the maximum number of streams that will be used to perform the tape backup operation.

   If you do not enter any value, then all available streams will be used.
6. Optionally, use the **Media Management Vendor Parameters** field to specify additional parameters to define your tape backup job. The Recovery Appliance uses a combination of these parameters and the media manager library parameters to complete the tape backup job.
7. Optionally, in the **Media Management Vendor Commands** field, enter vendor-specific commands for your media manager software to control your tape backup job.
8. Click **OK**.

Creating an SBT Attribute Set Using DBMS_RA

An SBT attribute set is referenced by an SBT job and is a collection of attributes that controls a tape backup operation. The attribute set specifies the SBT library to use for the copy operation. It also specifies SBT channel parameters and parameters to pass to the media management software library. These parameters are merged with the parameters specified in the SBT library object.

> **Note:**
>
> If all SBT attribute sets share the same parameter value, then you can specify that parameter in the SBT library object instead of in each SBT attribute set.

To create an SBT attribute set:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.
2. Run the `DBMS_RA.CREATE_SBT_ATTRIBUTE_SET` procedure for each SBT attribute set that you want to create.

```
BEGIN
  DBMS_RA.CREATE_SBT_ATTRIBUTE_SET(
    lib_name             => 'osbsbt',
    attribute_set_name   => 'wholedb',
    streams              => 10,
    parms                => 'ENV=(OB_MEDIA_FAMILY=wholedb_mf)');
END;
```

Again in this example, the media management software is Oracle Secure Backup (OSB). The streams argument sets the maximum number of concurrent streams that can be used for automated backups. The `parms` argument has the same purpose and format as the `PARMS` clause of an RMAN `ALLOCATE CHANNEL` command. In this case it designates the `wholedb_mf` Oracle Secure Backup media family as the destination of the copy operation.

> **See Also:**
>
> - "CREATE_SBT_ATTRIBUTE_SET" for descriptions of procedure arguments
> - "Backup Retention on Tape" for a discussion of the `sbt_retention_policy` argument

Managing Tape Backup Job Components

After you have created your tape backup job components, you may need to modify their properties periodically based on job requirements or delete them when no longer required.

This section contains the following topics:

- Managing a Media Manager Library Using Cloud Control
- Managing an SBT Library Using DBMS_RA
- Managing an Attribute Set Using Cloud Control
- Managing an Attribute Set Using DBMS_RA

> **See Also:**
>
> "Creating Tape Backup Job Components"

Managing a Media Manager Library Using Cloud Control

This section contains the following topics:

- Editing a Media Manager Library
- Deleting a Media Manager Library

Editing a Media Manager Library

Edit the existing properties of a media manager library to update parameters based on your tape backup job requirements.

To edit a media manager library:

1. Complete the steps in "Accessing the Recovery Appliance Home Page".

   On the Recovery Appliance Home page, select **Media Managers** from the Recovery Appliance menu.

2. On the Media Managers page, select the Media Manager library that you want to edit and click **Edit**.

   The Edit Media Manager Library screen appears with the existing library parameters.

Figure 7-2 Edit Media Manager Library Screen

3. Change the number of **Maximum Drives**.

   Optionally, edit the Advanced Parameters by changing the number of **Restore Drives** and values in **Media Management Vendor Parameters**.

   If you are using a third party media manager, then edit the required parameters for that specific media manager.

4. Click **OK**.

Deleting a Media Manager Library

You can delete an existing media manager library after all tape backup operations associated with this media manager are complete and you no longer require the parameters specified as a part of this library.

To delete a media manager library:

1. Complete the steps in "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Media Managers**.
3. From the list of existing media manager libraries, select the Media Manager Library you want to delete.
4. Click **Delete**.

   A message asks you confirm the deletion of this library.
5. Click **Yes**.

Managing an SBT Library Using DBMS_RA

This section contains the following topics:

- Editing an SBT Library
- Deleting an SBT Library

Editing an SBT Library

You can modify one or more attributes of an SBT library by calling the given procedure in the `DBMS_RA` PL/SQL package.

To edit an SBT library:

1. Using SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the `UPDATE_SBT_LIBRARY` procedure, providing the name of the SBT library to modify and the new values for its attributes.

   Attributes omitted from the procedure call are left unchanged.

> **See Also:**
>
> DBMS_RA Package Reference

Deleting an SBT Library

You can delete an SBT library by calling the given procedure in the `DBMS_RA` PL/SQL package.

To delete an SBT library:

1. Using SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the DELETE_SBT_LIBRARY procedure, providing the name of the SBT object to delete.

Example 7-1 Deleting an SBT Library

```
BEGIN
  DBMS_RA.DELETE_SBT_LIBRARY(
    lib_name => 'OSBSBT');
END;
```

> **See Also:**
>
> DBMS_RA Package Reference

Managing an Attribute Set Using Cloud Control

This section contains the following topics:

- Editing an Attribute Set
- Deleting an Attribute Set

Editing an Attribute Set

You can edit the existing properties of a media manager attribute set to modify your tape backup job settings at a job-specific level.

To edit an attribute set:

1. Complete the steps "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Media Managers**.
3. From the list of attribute sets, select one attribute set that you need to edit.
4. Make the required changes to the **Pool ID**, **Media Management Vendor Parameters**, and **Media Management Vendor Command** values.
5. Click **OK**.

Deleting an Attribute Set

You can delete an existing attribute set after all associated tape backup jobs are complete and you no longer require the job parameters specified in the attribute set.

To delete an attribute set:

1. Complete the steps in "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Media Managers**.
3. In the Attribute Sets section, select the attribute list that you want to delete.

4. Click **Delete**.

   A message asks you to confirm deletion of this attribute set.

5. Click **Yes**.

Managing an Attribute Set Using DBMS_RA

This section contains the following topics:

- Editing an SBT Attribute Set
- Deleting an SBT Attribute Set

Editing an SBT Attribute Set

You can modify one or more attributes of an SBT attribute set by calling the given procedure in the `DBMS_RA` PL/SQL package.

To delete an SBT attribute set:

1. Using SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.
2. Run the `UPDATE_SBT_ATTRIBUTE_SET` procedure, providing the name of the SBT attribute set to modify and the new values for its attributes.

   Attributes omitted from the procedure call are left unchanged.

> **See Also:**
>
> DBMS_RA Package Reference

Deleting an SBT Attribute Set

You can delete an SBT attribute set by calling the given procedure in the `DBMS_RA` PL/SQL package.

To delete an SBT attribute set:

1. Using SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.
2. Run the `DELETE_SBT_ATTRIBUTE_SET` procedure, providing the name of the SBT object to delete.

> **See Also:**
>
> DBMS_RA Package Reference

Creating a Tape Backup Job

This section describes how to create tape backup jobs for a selected database or databases associated with a protection policy. The tape backup job templates define the properties for backups that need to be stored on tape. Media manager libraries and their attribute sets manage these job settings.

Recovery Appliance gives you the option to copy backups from multiple protected databases to tape. To perform this task, specify the protection policy that contains the protected databases for which you want to copy backups to tape. Alternatively, you can also copy backups to tape for a single specified database.

> **See Also:**
>
> - "Creating a Media Manager Library" for more information on configuring media manager libraries
> - "Creating an Attribute Set" for more information on configuring attribute sets

You can create a tape backup job using Cloud Control or command-line options. This section contains the following topics:

- Creating a Tape Backup Job Using Cloud Control
- Example: Creating a Tape Backup Job Using Cloud Control
- Creating a Tape Backup Job Using DBMS_RA
- Example: Creating a Tape Backup Job Using DBMS_RA

Creating a Tape Backup Job Using Cloud Control

Cloud Control is the graphical user interface that you can use to create a tape backup job.

To create a tape backup job:

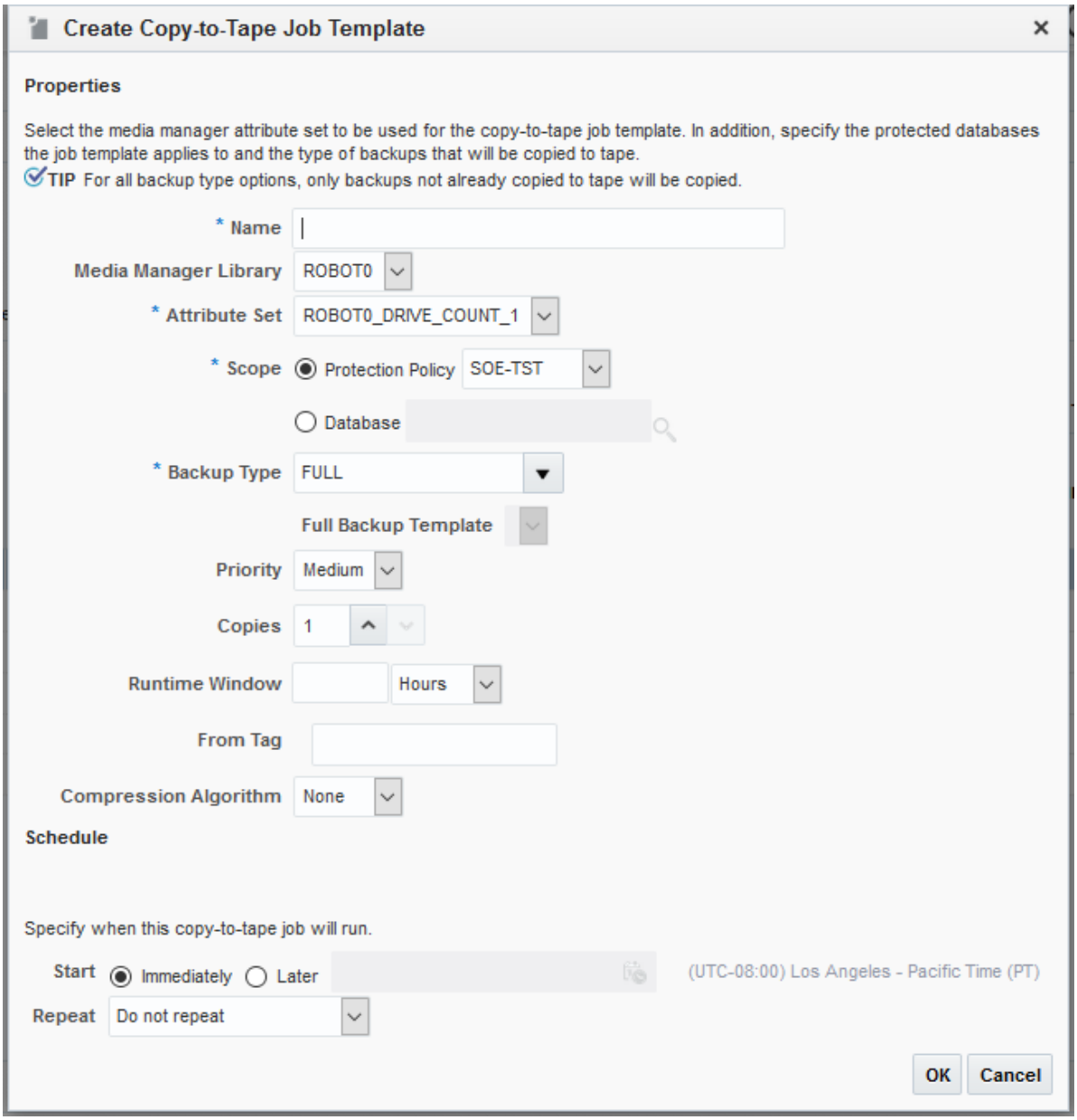
1. Complete the steps in "Accessing the Recovery Appliance Home Page"
2. On the Recovery Appliance Home page, from the Recovery Appliance menu, select **Copy-To-Tape Job Templates**.

   The Copy-to-Tape Job Templates page appears.
3. Click **Create** to create a new tape backup job.

   The Create Copy-to-Tape Job Template page appears as shown in Figure 7-3.

   This page displays the job properties and schedule settings.

Figure 7-3 Recovery Appliance Create Copy-to-Tape Job Template Page



4. In the **Name** field, enter a name for the job.
5. Optionally, from the **Media Manager Library** drop-down list, select a media manager library that will manage this job.
6. From the **Attribute Set** drop-down list, select an attribute set that you want to use for this job. This attribute set will define the settings for your tape backup job.
7. In the **Scope** field, add one of the following:
   - Select the protection policy from the **Protection Policy** drop-down list that includes all the databases for which you want to copy the backups to tape.
   - Search for and select a single database for which you want to copy the backups to tape.

8. In the **Backup Type** field, select the type of backup to be copied. Options include: Full backup, Incremental Backup, and Archived Log.
9. (Optional) In the **Priority** field, select the priority for this job. The default job priority is Medium.
10. In the **Number of copies** field, enter the number of copies you need for the backup being copied to tape. You can choose a maximum of 4 copies and a minimum of 1 copy.

> **Note:**
>
> You cannot obtain duplicate copies of a backup after it has been copied to tape.

11. In the **Runtime Window** field, enter the amount of time in minutes, hours or days allowed for this job to complete. Jobs that do not start within the specified window will be completed in the next available window slot.
12. In the **From Tag** field, specify a tag name to only copy backups associated with a certain tag to tape.
13. Under the **Schedule** section, specify whether you want this job to run immediately or at a later specified time.

> **See Also:**
>
> "Scheduling a Tape Backup Job Using Cloud Control"

14. Click **OK**.

    Cloud Control displays a message notifying that your job request has been submitted successfully.

    You can click the job name in the message to check the queued backup images for this job.

> **See Also:**
>
> - "About Copying Backups to Tape with Recovery Appliance" for more information on tape backup operations
> - "Example: Creating a Tape Backup Job Using Cloud Control"

Example: Creating a Tape Backup Job Using Cloud Control

This example uses a combination of tape backup jobs to manage copying all your backups to tape and ensure that they are up-to-date. The combination of tape backup jobs stores all your backups on tape systematically and reduces chances of loss of information.

This example uses Cloud Control to perform all tasks.

> **See Also:**
>
> "Creating a Tape Backup Job Using Cloud Control"

To create an example tape backup scenario:

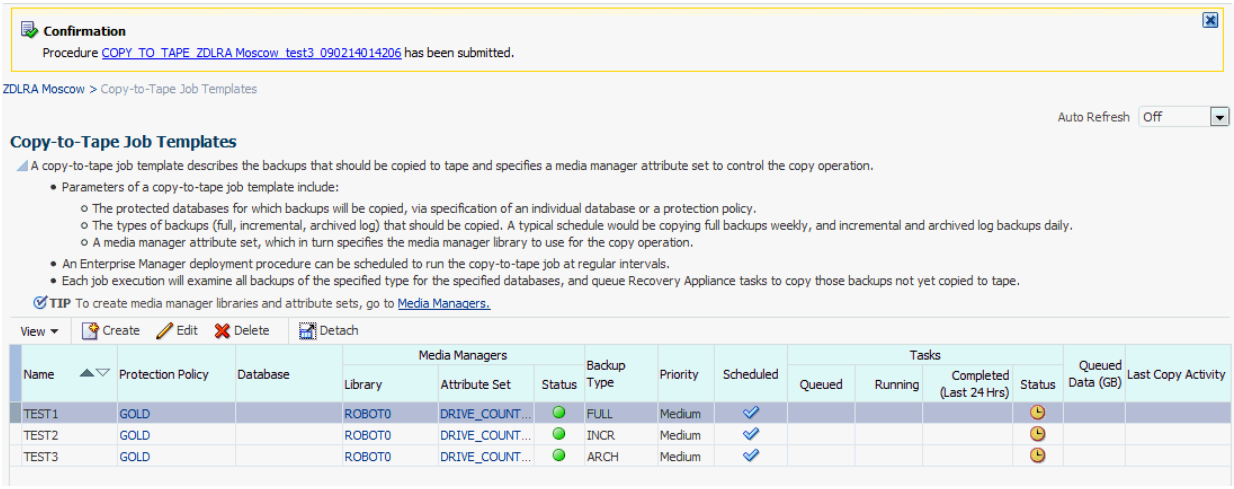
1. Create your first tape backup job and name this job `Test1`.
2. Use the default media manager library `ROBOT0` and the attribute set `DRIVE_COUNT_2` for this job.

   To edit the default media manager library values and default attribute set values, complete the steps in "Editing a Media Manager Library" and "Editing an Attribute Set", respectively.
3. Select the scope of this job as the Protection Policy `GOLD`.
4. Select **Full** as the Backup Type for this job.

   This selection implies that all your full database backups for databases included in `GOLD` will be copied to tape.
5. Schedule `Test1` to run on Sunday at 10:00 am and set it to repeat every 1 week.
6. Similarly, create a second tape backup job and name this job `Test2`.
7. Let this job use the same media manager library, protection policy and attribute set as `Test1`.
8. Select **Incremental** from the Backup Type drop-down list for this job.
9. Schedule `Test2` to run from Monday to Saturday at 12:30 pm and set it to repeat daily.
10. Using the same steps as above, create a third tape backup job and name it `Test3`.
11. Select **Archived Logs** as the Backup Type for this job.
12. Schedule `Test3` to run at 6 hour intervals.

Figure 7-4 displays the Recovery Appliance Copy-to-Tape Templates screen after these jobs have been successfully submitted.

Figure 7-4 Tape Backup Jobs Example



In this scenario, the Recovery Appliance copies all full backups to tape once a week. Afterward, the Recovery Appliance copies all incremental backups with the latest changes daily to maintain the updated backup copies on tape. Similarly, the system copies all archived redo log files every 6 hours to ensure efficient copy and storage of backups on tape.

Creating a Tape Backup Job Using DBMS_RA

Each SBT job defines the backups to be copied to tape and the media pool to which to copy them. After you create SBT jobs, you must schedule their execution.

To create an SBT job template:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.
2. Run the CREATE_SBT_JOB_TEMPLATE procedure for each SBT job that you want to create.

> **Note:**
>
> `CREATE_SBT_JOB_TEMPLATE` is an overloaded procedure. With one procedure signature, you specify the `db_unique_name` of a protected database for which to consider backups for copying. With the other procedure signature, shown below, you specify a protection policy name. In this case, backups for all protected databases assigned to the protection policy are considered for copying.

3. If a protection policy name is specified with the `protection_policy_name` parameter, then when the SBT job runs, backups for all databases assigned to the protection policy are considered for copying to tape. If a `db_unique_name` is specified, then only backups for that database are considered for copying.

4. Using the `attribute_set_name` parameter, specify the name of an SBT attribute set, which is a collection of attributes that control tape backup operations. The attribute set specifies the SBT library to use for the copy operation. It also specifies SBT channel parameters and parameters to pass to the media management software library. These parameters are merged with the parameters specified in the SBT library object.
5. Using the `backup_type` parameter, add the types of backups to copy, expressed as a comma-delimited list of the following types: `ALL`, `INCR`, `ARCH`, or `FULL`. For example, if `'INCR,ARCH'` is specified, then all incremental (level 1) backups and archived log files that have not yet been copied to the named media manager are included.
6. With the `priority` parameter, enter a priority level for this job. When many SBT jobs are scheduled to run simultaneously, the job priority determines the job that runs first. Job priority is needed when there are not enough tape drives to service all of the jobs that are scheduled to run simultaneously. Job priority is expressed as one of the following predefined values:
   - `1000 (SBT_PRORITY_LOW)`
   - `100 (SBT_PRIORITY_MEDIUM)`
   - `10 (SBT_PRIORITY_HIGH)`
   - `1 (SBT_PRIORITY_CRITICAL)`

   `0` is the highest possible priority. Lower priority values take precedence over higher values. The default priority is `100 (SBT_PRIORITY_MEDIUM)`.

> **See Also:**
>
> - "About Copying Backups to Tape with Recovery Appliance" for more information on tape backup operations
> - "CREATE_SBT_JOB_TEMPLATE" for more information on procedure related arguments
> - "Scheduling Tape Backup Jobs with Oracle Scheduler" for more information on how to run an SBT job
> - "Example: Creating a Tape Backup Job Using DBMS_RA"

Example: Creating a Tape Backup Job Using DBMS_RA

This example illustrates how to create an SBT job using the `CREATE_SBT_JOB_TEMPLATE` procedure.

```
BEGIN
  DBMS_RA.CREATE_SBT_JOB_TEMPLATE (
    template_name           => 'oltp_arch_lastfull',
    protection_policy_name  => 'oltp',
    attribute_set_name      => 'wholedb',
    backup_type             => 'FULL,ARCH',
    priority                => DBMS_RA.SBT_PRIORITY_HIGH,
    window                  => INTERVAL '4' HOUR);
END;
```

In this example, the SBT job selects all archived log files and the last full backup for every protected database assigned to the `oltp` protection policy. The last full backup could be either the most recent level 0 backup received, or a virtual full backup based on the most recent level 1 backup received, whichever is later.

The SBT job references the `wholedb` SBT attribute set, which specifies the SBT library to use for the copy operation, and specifies SBT channel parameters and parameters to pass to the media management software library.

The `backup_type` parameter copies all archived log backups not yet copied to SBT and the most recent full backup, if it has not already been copied to tape. For example, the `backup_type` of "full, arch" selects all archived log backups, and the most recent full backup. These are the backups that will be copied to SBT when this job is run, if they have not already been copied.

The four-hour window specifies the length of time that copy tasks generated by this job are eligible to be started. When the window expires, any SBT copy tasks that were generated by this job but not yet started will be suspended until the next time this SBT job is scheduled. Copy tasks that are already running when the window expires are allowed to complete.

Not shown are the optional `copies` and `from_tag` arguments.

Managing a Tape Backup Job

After you have created a tape backup job, you may need to modify some of its properties based on job requirements or delete it when no longer required.

> **See Also:**
>
> "Creating a Tape Backup Job" for more information on how to create a tape backup job

This chapter contains the following topics:

- Managing a Tape Backup Job Using Cloud Control
- Managing a Tape Backup Job Using DBMS_RA

Managing a Tape Backup Job Using Cloud Control

This section contains the following topics:

- Editing a Tape Backup Job
- Deleting a Tape Backup Job

Editing a Tape Backup Job

After you have created a tape backup job, you can modify its properties using Cloud Control. If you have scheduled the job to run at a later time, then you can edit its parameters and schedule settings.

To edit a tape backup job:

1. Complete the steps in "Creating a Tape Backup Job Using Cloud Control".
2. Select the tape backup job that you want to edit.
3. Click **Edit** to change the existing job properties and schedule settings.
4. Make the required changes to the existing **Media Manager Library**, **Attribute Set**, **Priority**, **Copies**, **Runtime Window**, and **From Tag** values, to edit job properties.
5. Make changes the existing schedule options, to edit the job schedule settings.

> **See Also:**
>
> "Scheduling a Tape Backup Job Using Cloud Control"

6. Click **OK**.

Deleting a Tape Backup Job

You can delete an existing tape backup job after the operation is complete and all required backups are copied to tape. If you delete an ongoing tape backup job, then the tape backup operation will stop copying selected backups to tape.

To delete a tape backup job:

1. Complete the steps in "Accessing the Recovery Appliance Home Page".
2. On the Recovery Appliance Home page, from the Recovery Appliance menu, select **Copy-to-Tape Job Templates**.

   The Copy-to-Tape Job Template page appears.
3. From the list of jobs, select the job that you want to delete.
4. Click **Delete**.

   A confirmation message appears asking whether you want to continue with deleting this job. Click **Yes**.

Managing a Tape Backup Job Using DBMS_RA

This section contains the following topics:

- Editing an SBT Job
- Deleting an SBT Job

Editing an SBT Job

You can modify one or more attributes of an SBT job by calling the given procedure in the `DBMS_RA` PL/SQL package.

To edit an SBT job:

1. Using SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.

2. Run the UPDATE_SBT_JOB_TEMPLATE procedure, providing the name of the SBT job to modify and the new values for its attributes.

Example 7-2 Modifying an SBT Job

This example modifies the priority of the SBT job named `oltp_arch_lastfull`. The values of other arguments present in CREATE_SBT_JOB_TEMPLATE and omitted in this call remain unchanged.

```
BEGIN
  DBMS_RA.UPDATE_SBT_JOB_TEMPLATE(
    template_name => 'oltp_arch_lastfull',
    priority      => DBMS_RA.SBT_PRIORITY_HIGH);
END;
```

> **See Also:**
>
> DBMS_RA Package Reference

Deleting an SBT Job

You can delete a SBT job by calling the given `DBMS_RA` PL/SQL package procedure.

To delete an SBT job:

1. Using SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the DELETE_SBT_JOB_TEMPLATE procedure, providing the name of the SBT job to delete.

> **See Also:**
>
> DBMS_RA Package Reference

Scheduling a Tape Backup Job

This section contains the following topics:

- Scheduling a Tape Backup Job Using Cloud Control
- Scheduling Tape Backup Jobs with Oracle Scheduler

Scheduling a Tape Backup Job Using Cloud Control

This section describes the steps required to set the schedule for a tape backup job.

To schedule a tape backup job:

1. Complete the steps in "Accessing Recovery Appliance in Cloud Control".
2. In the **Schedule** section, select one of the following:

- **Immediately**: Runs the tape backup job right away.
- **Later**: Runs the tape backup job at a future, specified time. Enter the date and time when you want the job to run in the `mm/dd/yyyy hh:mm:ss` format. If you want this job to repeat, then select the frequency from the **Repeat** drop-down list and enter the applicable values.

3. Click **OK** to save the job.

   To edit the schedule settings, select the required tape backup job, click **Edit**, and make the necessary changes.

> **See Also:**
>
> "Creating a Tape Backup Job Using Cloud Control" for more information on tape backup job properties

Scheduling Tape Backup Jobs with Oracle Scheduler

After creating an SBT job, you must schedule its execution. Typically, you schedule the job to run at regular intervals using a scheduling utility such as Oracle Scheduler.

When an SBT job runs, Recovery Appliance examines all backups of the specified type for the specified databases, and selects the ones that have not yet been copied to tape. Recovery Appliance then generates and queues tasks that copy those backups to tape. One task of type `BACKUP_SBT` is added to the Recovery Appliance task queue for each backup piece to copy.

To schedule a tape backup job with Oracle Scheduler, you create a job that will invoke `DBMS_RA.QUEUE_SBT_BACKUP_TASK`. This procedure takes a single argument, the name of the SBT job.

To schedule an SBT job with Oracle Scheduler:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.
2. Run `DBMS_SCHEDULER.CREATE_JOB`.

   For example, run the following PL/SQL anonymous block:

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB(
   job_name   => 'sbtjob1',
   job_type        => 'PLSQL_BLOCK',
   job_action      => 'dbms_ra.queue_sbt_backup_task(''oltp_arch_lastfull'');',
   start_date      => SYSTIMESTAMP,
   enabled         => TRUE,
   auto_drop       => TRUE,
   repeat_interval => 'freq=WEEKLY; BYDAY=MON,WED,FRI; BYHOUR=23');
END;
```

See Also:

- "Creating a Tape Backup Job Using DBMS_RA" to learn how to create an SBT job
- "QUEUE_SBT_BACKUP_TASK"
- See *Oracle Database Administrator's Guide* and *Oracle Database PL/SQL Packages and Types Reference* for information about Oracle Scheduler

Pausing and Resuming Tape Backup Operations

This section describes the steps to pause an ongoing tape backup job or media manager operations and how to resume them later.

This section contains the following topics:

- Pausing and Resuming Media Manager Library Operations Using Cloud Control
- Pausing and Resuming the SBT Library Using DBMS_RA

Pausing and Resuming Media Manager Library Operations Using Cloud Control

You can choose to pause media manager operations using Cloud Control, and then resume them when required.

This section contains the following topics:

- Pausing a Media Manager Library
- Resuming a Media Manager Library

Pausing a Media Manager Library

You pause ongoing media manager library operations in a situation where you want to put the current media manager library on hold and resume it after a certain period.

To pause media manager library operations:

1. Complete the steps in "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Media Managers**.
3. From the list of media manager libraries, select the library for which you want to pause all ongoing jobs.
4. Click **Pause**.

Resuming a Media Manager Library

You can resume operations of any paused media manager library for the library and its associated job operations to continue.

> **See Also:**
>
> "Pausing a Media Manager Library"

To resume media manager library operations:

1. Complete the steps in "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Media Managers**.
3. Select the paused media manager library for which you want to resume job operations.
4. Click **Resume**.

Pausing and Resuming the SBT Library Using DBMS_RA

This section contains the following topics:

- Pausing an SBT Library
- Resuming an SBT Library

Pausing an SBT Library

Complete these steps to pause an SBT library.

To pause an SBT Library:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance database as the Recovery Appliance administrator.
2. Run the `DBMS_RA.PAUSE_SBT_LIBRARY` procedure.

   For example, run the following PL/SQL anonymous block:

```
BEGIN
  DBMS_RA.PAUSE_SBT_LIBRARY(
    lib_name => 'osbsbt');
END;
```

The library status in the view `RA_SBT_LIBRARY` changes to `PAUSE`.

> **See Also:**
>
> - "PAUSE_SBT_LIBRARY"
> - "RA_SBT_LIBRARY"

Resuming an SBT Library

Complete these steps to resume a paused SBT library.

To resume an SBT Library

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the `DBMS_RA.RESUME_SBT_LIBRARY` procedure.

   For example, run the following PL/SQL anonymous block:

```
BEGIN
  DBMS_RA.RESUME_SBT_LIBRARY(
    lib_name => 'osbsbt');
END;
```

The library status in the view `RA_SBT_LIBRARY` changes back to `READY`.

> **See Also:**
>
> - "RESUME_SBT_LIBRARY"
> - "RA_SBT_LIBRARY"

Viewing the Status of Tape Backup Operations

This section describes the steps to check the status of media manager library and tape backup jobs to monitor the progress of their operations.

This section contains the following topics:

- Viewing the Status of Tape Backup Operations Using Cloud Control
- Viewing the Status of Tape Backup Operations Using DBMS_RA

Viewing the Status of Tape Backup Operations Using Cloud Control

This section contains the following topics:

- Viewing the Media Manager Library Status
- Viewing the Tape Backup Job Status

Viewing the Media Manager Library Status

You can check for any errors in the media management software that may affect the ongoing tape backup operations by viewing the current status of a media manager library.

To check the status of a media manager library:

1. Complete the steps "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Media Managers.**

   The Media Managers page lists the status of all existing media manager libraries in the corresponding column.

Viewing the Tape Backup Job Status

You can check for any error in the ongoing tape backup operation by viewing the current status of the job.

To view the status of a tape backup job:

1. Complete the steps "Accessing Recovery Appliance in Cloud Control".
2. From the Recovery Appliance Menu, select **Copy-to-Tape Job Templates**.

   The Copy-to-Tape Job Templates Page displays the status of each tape backup job in its corresponding column.

Viewing the Status of Tape Backup Operations Using DBMS_RA

This section contains the following topics:

- Checking the SBT Library Status
- Checking the Tape Backup Job Status
- Reviewing SBT Job Runs Using DBMS_RA
- Checking the Status of Oracle Scheduler Jobs

Checking the SBT Library Status

You can determine if an error condition exists in the media management software by querying the `RA_SBT_LIBRARY` view. You can also determine the `PAUSE` state of an SBT library.

To view the status of an SBT library:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the following query:

```
SELECT LIB_NAME, LAST_ERROR_TEXT, STATUS
FROM   RA_SBT_LIBRARY;

LIB_NAME            LAST_ERROR_TEXT                                  STATUS
------------------- ------------------------------------------------ ------
OSBSBT                                                               READY
```

Table 7-3 lists the possible values for `STATUS`.

Table 7-3 Values for the STATUS Column of RA_SBT_LIBRARY

Value	Meaning
READY	The SBT library was properly created and is ready to process tape I/O.
PAUSE	The SBT library was paused with PAUSE_SBT_LIBRARY.

Table 7-3 (Cont.) Values for the STATUS Column of RA_SBT_LIBRARY

Value	Meaning
ERROR	An error condition exists in the media management software. Tape backup operations cannot continue until you clear the error and call RESUME_SBT_LIBRARY. The `last_error_text` column describes the most recent error returned by the media management library. The error text also appears in the RA_SBT_TASK rows for the affected background SBT tasks.

> **See Also:**
>
> "RA_SBT_LIBRARY"

Checking the Tape Backup Job Status

When an SBT job runs it generates and queues a task for each backup piece to be copied. The view RA_SBT_TASK lists these tasks and their completion states.

> **Note:**
>
> Completed tasks are removed from this view after 30 days.

To check the status of SBT jobs:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the following query (sample output shown):

```
SELECT TASK_ID, STATE, DB_UNIQUE_NAME, ERROR_TEXT, BS_KEY, PIECE#
FROM   RA_SBT_TASK
WHERE  SBT_TEMPLATE_NAME = 'SBTJOB1'
ORDER BY DB_UNIQUE_NAME, BS_KEY, PIECE#;

TASK_ID STATE      DB_UNIQUE_NAME ERROR_TEXT           BS_KEY PIECE#
------- ---------- -------------- -------------------- ------ ------
    253 COMPLETED  OLTP1                                  89      1
    254 COMPLETED  OLTP1                                  95      1
    255 COMPLETED  OLTP1                                  99      1
    256 COMPLETED  OLTP1                                 117      1
    257 COMPLETED  OLTP1                                 141      1
```

A task failed if its STATE is KILLED or ABORTED.

Reviewing SBT Job Runs Using DBMS_RA

You can determine whether SBT jobs run according to your defined schedule by querying the RA_SBT_JOB.LAST_SCHEDULE_TIME column. This column indicates the last time that the SBT job was scheduled to run.

To review completed SBT jobs:

1. With SQL*Plus or SQL Developer, connect to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the following query (sample output shown):

```
SELECT TEMPLATE_NAME, BACKUP_TYPE, LAST_SCHEDULE_TIME
FROM   RA_SBT_JOB;

TEMPLATE_NAME          BACKUP_TYPE          LAST_SCHEDULE_TIME
-----------------  -------------------  ---------------------------
OLTP_ARCH_LASTFULL  ARCH, FULL          27-AUG-12 10.53.49 AM -08:00
OLTP_INCR           INCR                26-AUG-12 10.16.16 AM -08:00
```

> **See Also:**
>
> "RA_SBT_LIBRARY"

Checking the Status of Oracle Scheduler Jobs

To view the status of an Oracle Scheduler job created to run an SBT job:

- Query the following views:
  - `USER_SCHEDULER_JOBS`
  - `USER_SCHEDULER_JOB_LOG`

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details about data dictionary views for Oracle Scheduler

8

Replicating Backups with Recovery Appliance

This chapter explains the purpose for replication of Recovery Appliance for disaster recovery, provides examples of replication topologies, and concludes with sections on how to configure replication using Cloud Control and alternatively using `DBMS_RA`. This chapter contains the following major sections:

- About Recovery Appliance Replication
- Configuring Recovery Appliance Replication Using Cloud Control
- Configuring Recovery Appliance for Replication Using DBMS_RA

About Recovery Appliance Replication

As part of a disaster recovery strategy, Recovery Appliance can replicate backups to other Recovery Appliances. Also, you can offload tape archival to a replicated Recovery Appliance, thereby freeing resources on the primary Recovery Appliance. Replication is driven by protection policy, which means that all databases associated with the policy are replicated, and it is fully automatic after the initial setup.

Oracle requires that you create a replication user account exclusively for use with Recovery Appliance replication, and that you create a unique replication user account for each upstream appliance within the organization.

Oracle recommends that the replication user account takes the form of `REPUSER_FROM_[ZDLRA_DB_NAME` or `ZDLRA_DB_LOCATION]`.

For example, if two Recovery Appliances have the `DB_UNIQUE_NAME` of ZDLRA1 and ZDLRA2, then the replication user accounts could be `REPUSER_FROM_ZDLRA1` and `REPUSER_FROM_ZDLRA2`. Or if those same Recovery Appliances were in Florence and Vienna, then the replication user accounts could be `REPUSER_FROM_FLORENCE` and `REPUSER_FROM_VIENNA`

The replication user account **should not** be used as a regular VPC user employed by protected databases to connect and send backups to the Recovery Appliance.

This section contains the following topics:

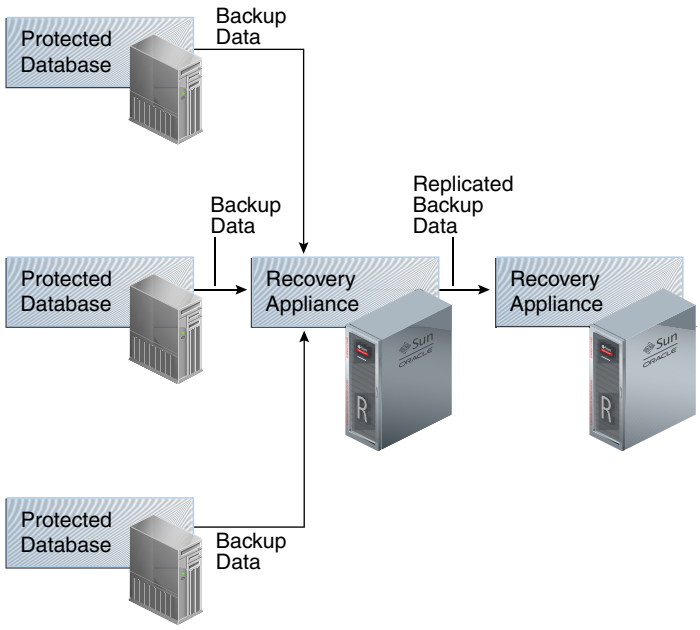
- Overview of Recovery Appliance Replication
- User Interfaces for Recovery Appliance Replication
- Basic Tasks for Configuring Recovery Appliance Replication

> **See Also:**
>
> "Secure Replication", and "Recovery Appliance Replication"

Overview of Recovery Appliance Replication

In the simple replication topology in Figure 8-1, a protected database sends backups to one Recovery Appliance, which passes the backups on to another Recovery Appliance. This topology is called one-way Recovery Appliance replication. The first Recovery Appliance is the upstream Recovery Appliance and the second is the downstream Recovery Appliance.

Figure 8-1 Simple Replication Topology



This section contains the following topics:

- Protection Policies for Replication
- Replication Topology Examples
- How Recovery Appliance Replicates Backups: Basic Process
- How RMAN Restores Backups in a Replication Environment

Protection Policies for Replication

Replication for a protected database occurs when the following conditions are met:

- On the upstream Recovery Appliance, a replication server configuration specifies a Recovery Appliance acting as a downstream replication Recovery Appliance (`CREATE_REPLICATION_SERVER`).
- On the upstream Recovery Appliance, a protection policy is associated with the replication server configuration (`ADD_REPLICATION_SERVER`).
- On the upstream Recovery Appliance, a protected database is assigned (`ADD_DB`) to the protection policy associated with the replication server configuration.

- On the downstream Recovery Appliance, a protection policy for the replicated backups must exist (`CREATE_PROTECTION_POLICY`), and the protected databases must be added to it (`ADD_DB`).

When you complete configuration of the protection policy on the upstream Recovery Appliance, the Recovery Appliance immediately replicates only the last full backup for each database protected by the policy. The backup can be either the most recent level 0 backup received, or a virtual full backup based on the most recent level 1 backup received, whichever is later. The upstream Recovery Appliance replicates new backups as it receives them.

Replication Topology Examples

Replication topologies can be as complex as required. The primary variables are as follows:

- The total number of Recovery Appliances in the replication environment, and their relationships to one another
- The protection policies (`CREATE_PROTECTION_POLICY`) on the upstream Recovery Appliance that manage the *outgoing* replicated backups, and the policies on the downstream Recovery Appliance that manage the *incoming* replicated backups
- The replication server configurations (`CREATE_REPLICATION_SERVER`) that exist on each Recovery Appliance in the replication environment
- The association between a replication server configuration and a protection policy (`ADD_REPLICATION_SERVER`)

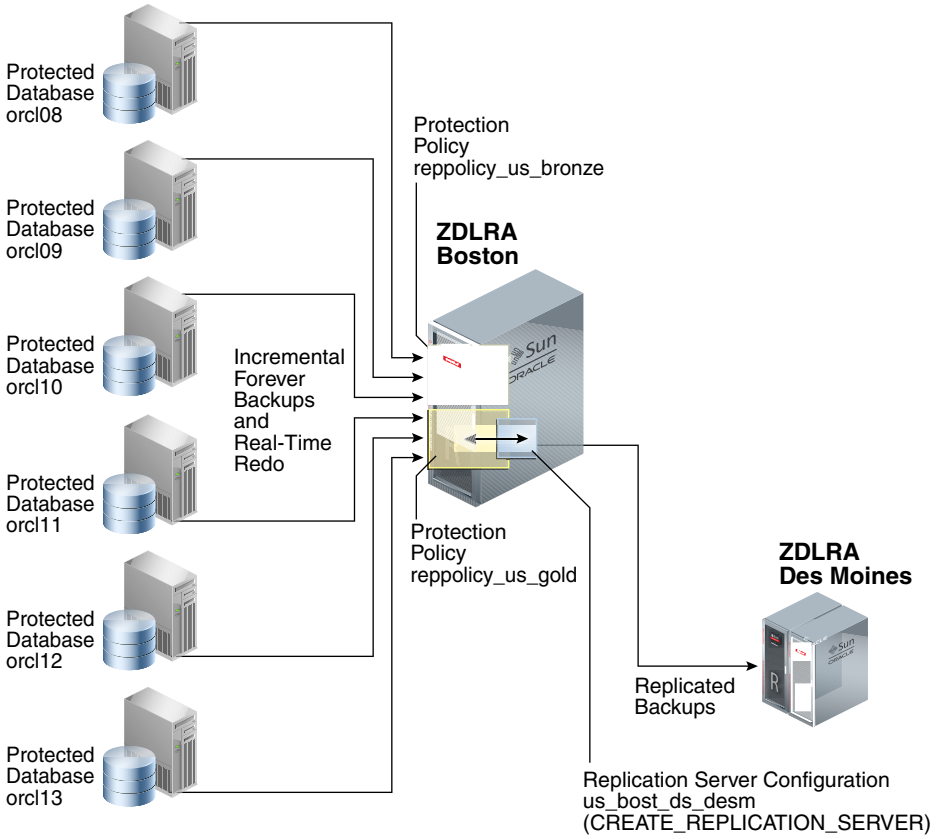
You can chain replication so that an upstream Recovery Appliance replicates to multiple downstream Recovery Appliances, while a downstream Recovery Appliance receives backups from multiple upstream Recovery Appliances. A downstream Recovery Appliance can receive both backups from both its own protected databases and replicated backups. Any Recovery Appliance in the replication topology can simultaneously perform the upstream and downstream replication roles.

> **Note:**
>
> If a Recovery Appliance is both upstream and downstream, then you must configure it for both roles.

Replication to One Downstream Recovery Appliance

Figure 8-2 shows three databases associated with the `reppolicy_us_bronze` protection policy (`orcl08`, `orcl09`, and `orcl10`), and three databases associated with the `reppolicy_us_gold` protection policy (`orcl11`, `orcl12`, and `orcl13`). Only `reppolicy_us_gold` is associated with a replication server configuration, which is named `us_bost_ds_desm`. In this topology, the upstream `ZDLRA Boston` only transfers backups from databases protected by the `reppolicy_us_gold` policy to the downstream `ZDLRA Des Moines`.

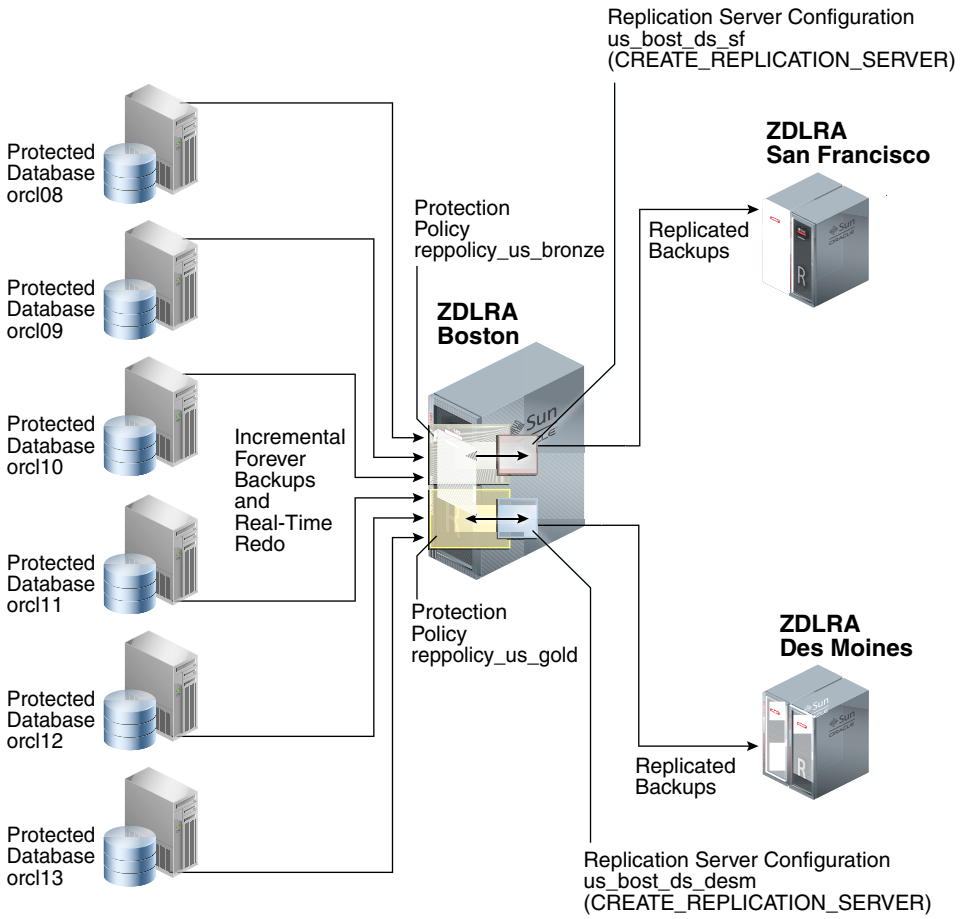
Figure 8-2 Databases Replicating to One Recovery Appliance



Replication to Multiple Downstream Recovery Appliances

Because each protected database has its own protection policy, each policy can be associated with a different replication server configuration. For example, in Figure 8-3, the `reppolicy_us_bronze` policy is associated with replication server configuration `us_bost_ds_sf`, which replicates backups for databases protected by `reppolicy_us_bronze` (`orcl08`, `orcl09`, and `orcl10`) to the downstream Recovery Appliance named `ZDLRA San Francisco`. The `reppolicy_us_gold` policy is associated with replication server configuration `us_bost_ds_desm`, which replicates backups for databases protected by `reppolicy_us_gold` (`orcl11`, `orcl12`, and `orcl13`) to the downstream Recovery Appliance named `ZDLRA Des Moines`.

Figure 8-3 Databases Replicated to Multiple Recovery Appliances

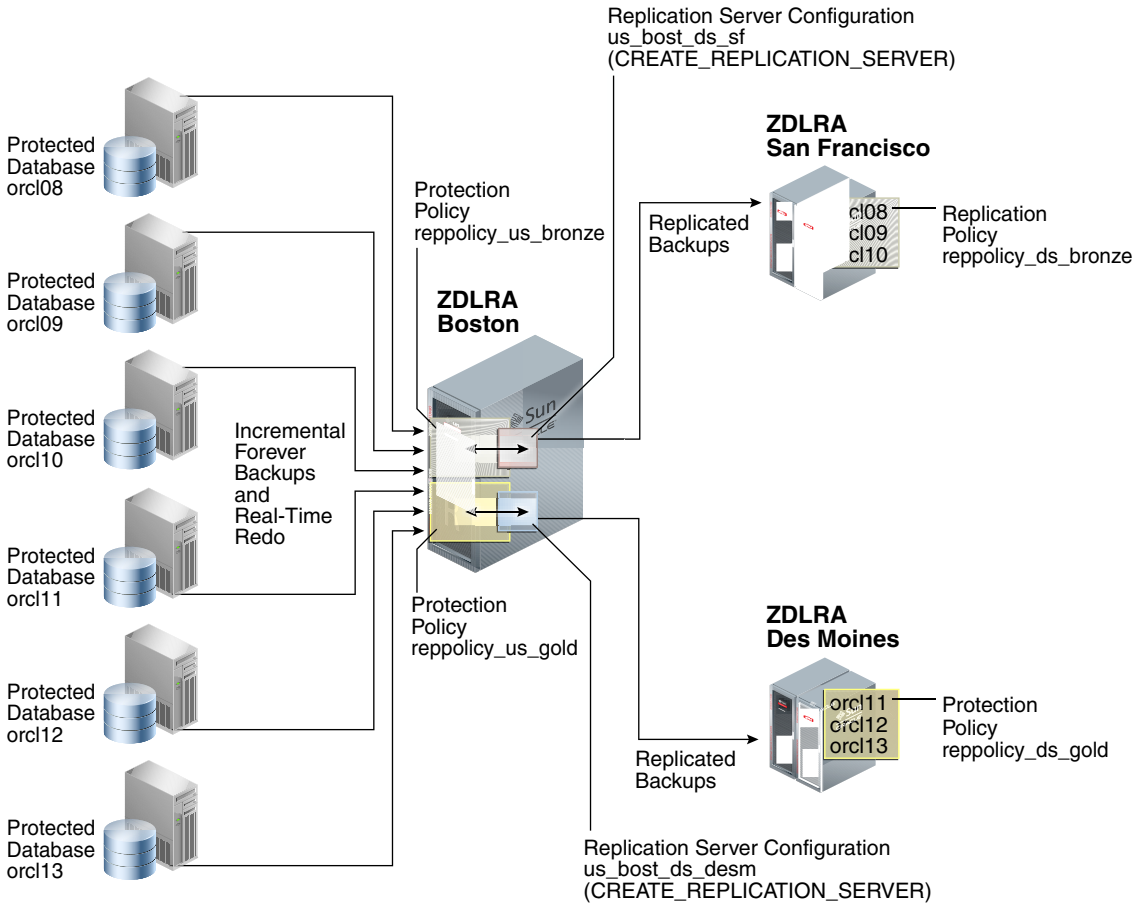


Replication Using Different Policies on Downstream Recovery Appliances

At each downstream Recovery Appliance in a replication scheme, the protection policy defines a disk recovery window goal and tape retention policy for received backups. The downstream configuration is completely independent of the upstream configuration. Thus, you have the flexibility to configure a downstream Recovery Appliance with more storage and longer recovery windows than its upstream Recovery Appliances, for example, using the downstream Recovery Appliance as a longer-term retention backup repository.

Figure 8-4 shows that `reppolicy_us_bronze` backups up on the upstream Recovery Appliance are protected by the `reppolicy_ds_bronze` policy on `ZDLRA San Francisco`. The `reppolicy_us_gold` backups up on the upstream Recovery Appliance are protected by the `reppolicy_ds_gold` policy on `ZDLRA Des Moines`.

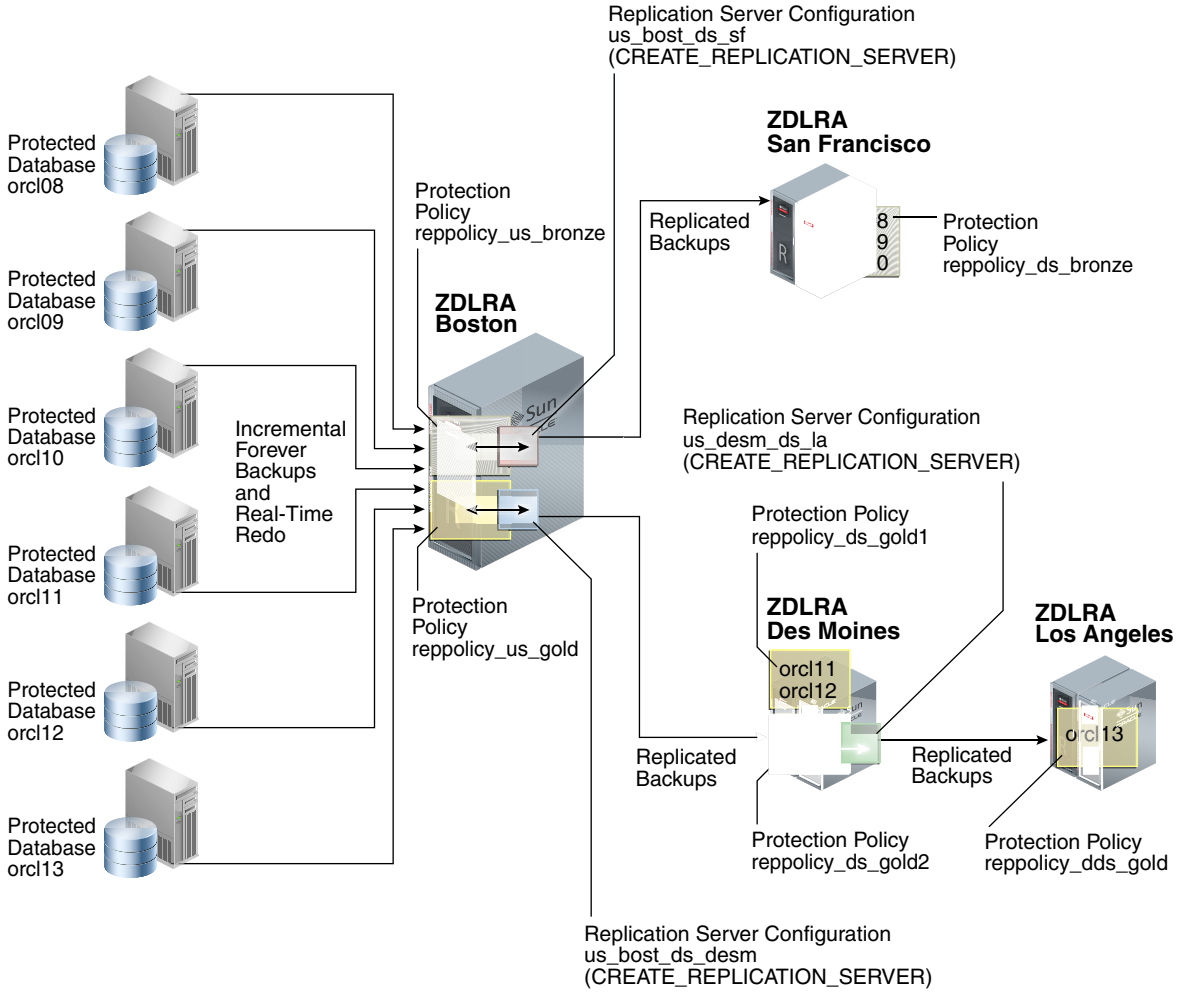
Figure 8-4 Different Protection Policies on Each Recovery Appliance



Cascaded Replication

Figure 8-5 shows a more complicated topology. Databases `orcl11`, `orcl12`, and `orcl13` are protected by the `reppolicy_us_gold` protection policy on `ZDLRA Boston`, which is furthest upstream. The `reppolicy_us_gold` policy replicates the backups for these databases to `ZDLRA Des Moines`, which is immediately downstream. However, two separate protection policies exist on `ZDLRA Des Moines`: `reppolicy_ds_gold1`, which protects databases `orcl11` and `orcl12`, and `reppolicy_ds_gold2`, which protects only database `orcl13`.

Figure 8-5 Cascaded Replication, with Different Protection Policies on Each Recovery Appliance



In Figure 8-5, the `reppolicy_ds_gold2` protection policy is associated with the replication server configuration `us_desm_ds_la`. `ZDLRA Des Moines` then replicates backups of `orcl13`, which is the only database protected by `reppolicy_ds_gold2`, to `ZDLRA Los Angeles`, which is the Recovery Appliance that is farthest downstream. The backups of `orcl13` that reside on `ZDLRA Los Angeles` are protected by the `reppolicy_dds_gold` policy. This configuration, in which three or more Recovery Appliances are linked in a chain, is called cascaded replication.

How Recovery Appliance Replicates Backups: Basic Process

Assume that a protected database backs up to a Recovery Appliance using the incremental-forever policy. When an protected database sends a backup to a Recovery Appliance configured for replication, the following basic steps occur:

1. The upstream Recovery Appliance ingests the backup, checking the protection policy to determine whether it is associated with a replication server configuration.

2. If a replication server configuration exists for the protection policy, then the upstream Recovery Appliance replicates the backup. The replication process includes:
   - Creating metadata records to track the replicated records

     > **Note:**
     >
     > When real-time redo transport is enabled, incoming redo changes are not replicated in real time by Recovery Appliance. When an archived redo log backup is created, the Recovery Appliance automatically replicates this backup along with the data file backups.

   - Transferring the data blocks over the network to each specified downstream Recovery Appliance
3. The downstream Recovery Appliance ingests the backup, creating a virtual backup.

   > **Note:**
   >
   > The ingest phase on the downstream is the same as the ingest phase described in Step 1. Thus, if the *downstream* Recovery Appliance is also configured to replicate the backup, then it assumes the role of an *upstream* Recovery Appliance, and then replicates the backup to the Recovery Appliances that are directly downstream, and so on.

4. Shortly afterward, the upstream Recovery Appliance sends a reconcile request to the downstream Recovery Appliance, which in turn sends metadata about the backup to the upstream Recovery Appliance.

   In Recovery Appliance replication, reconciling is the process by which a Recovery Appliance receives metadata from the Recovery Appliances that are immediately downstream.

Thus, after the backup is replicated, both the upstream and downstream recovery catalog have a record of the protected database backup.

How RMAN Restores Backups in a Replication Environment

To restore a protected database, RMAN typically connects `AS CATALOG` to the same Recovery Appliance to which it originally sent backups. For example, in Figure 8-2, if RMAN needed to restore `orcl11`, then RMAN would connect to the catalog on the upstream Recovery Appliance.

If backups exist on any Recovery Appliances in the replication scheme, then the upstream Recovery Appliance can retrieve and restore the backups from the other Recovery Appliances. For example, in Figure 8-2, if RMAN needed to restore `orcl11`, but the backup had been purged from the upstream Recovery Appliance, then the downstream Recovery Appliance could provide the backups to the upstream Recovery Appliance, which could then restore them.

If necessary, RMAN can also restore a backup directly from a downstream Recovery Appliance. RMAN connects `AS CATALOG` to the downstream Recovery Appliance, and

then restores the backup. For example, in Figure 8-2, if RMAN needed to restore `prod3`, but the upstream Recovery Appliance was temporarily inaccessible, then RMAN could connect directly to the catalog on the downstream Recovery Appliance, and then restore the backups directly to the protected database host.

> **Note:**
>
> When using either Oracle Enterprise Manager Cloud Control (Cloud Control) or the command line, restoring backups from a downstream Recovery Appliance requires additional configuration. See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*.

User Interfaces for Recovery Appliance Replication

This section contains the following topics:

- Accessing the Replication Page in Cloud Control
- DBMS_RA Procedures Relating to Replication
- Recovery Catalog Views for Replication

Accessing the Replication Page in Cloud Control

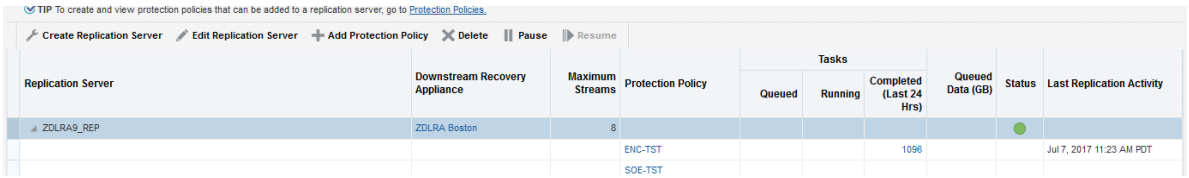
The Replication page in Cloud Control is the recommended interface for configuring Recovery Appliance replication.

To access the Replication page:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. From the **Recovery Appliance** menu, select **Replication**.

   The Replication page appears, as shown in Figure 8-6.

Figure 8-6 Replication Page



In the preceding sample, the replication server named `ZDLRA9_REP` is already configured. The Status column shows that it is available.

> **See Also:**
>
> Cloud Control help for more information about the Replication page

DBMS_RA Procedures Relating to Replication

You can use the `DBMS_RA` package to create and manage replication. Table 8-1 describes the principal program units relating to replication.

Table 8-1 Principal Procedures Relevant for Replication

Program Unit	Description
CREATE_REPLICATION_SERVER	Creates a replication server configuration that specifies a downstream Recovery Appliance to which this Recovery Appliance replicates backups.
DELETE_REPLICATION_SERVER	Deletes a replication server configuration.
ADD_REPLICATION_SERVER	Adds a replication server configuration to the protection policy that was created by the `CREATE_REPLICATION_SERVER` procedure.
REMOVE_REPLICATION_SERVER	Removes a replication server configuration from the protection policy that was created by the `CREATE_REPLICATION_SERVER` procedure.
ADD_DB	Adds a database to the protection policy.
CREATE_PROTECTION_POLICY	Creates a protection policy. To enable replication for databases assigned to this policy, you must associate a replication server configuration with this policy by running `ADD_REPLICATION_SERVER`.
UPDATE_DB	Updates the properties of a protected database.

> **See Also:**
>
> DBMS_RA Package Reference

Recovery Catalog Views for Replication

You can monitor replication using the Recovery Appliance catalog views. Table 8-2 summarizes the views that are most useful for replication.

Table 8-2 Views for Replication

View	Description
RA_REPLICATION_SERVER	This view describes the *downstream* Recovery Appliances that are directly receiving replicated backups from this particular Recovery Appliance. For example, Recovery Appliance A replicates to Recovery Appliance B, which replicates to Recovery Appliance C. The RA_REPLICATION_SERVER view on Recovery Appliance A lists Recovery Appliance B, but not Recovery Appliance C. The same view on Recovery Appliance B lists only Recovery Appliance C. The same view on Recovery Appliance C has no rows because no Recovery Appliances are downstream.
RA_DATABASE	The POLICY_NAME column of this view lists the protection policy used by this protected database. The REPLICATION_USAGE column shows the cumulative amount of disk space (in GB) replicated for this database.
RA_PROTECTION_POLICY	This view describes the defined protection policies.
RC_BACKUP_PIECE_DETAILS	Lists detailed information about all available backup pieces recorded in the recovery catalog.

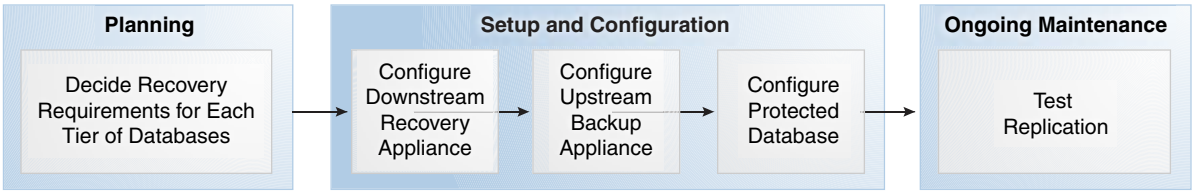
See Also:

Recovery Appliance View Reference

Basic Tasks for Configuring Recovery Appliance Replication

Figure 8-7 shows the basic workflow for configuring replication. "Planning for Recovery Appliance" describes the stage in the workflow in which you decide the recovery requirements for each service tier of databases.

Figure 8-7 Replication Workflow



You can perform the configuration using either Cloud Control or the command-line interface. Oracle recommends that you use Cloud Control because the steps are considerably less complicated.

Configuring Recovery Appliance Replication Using Cloud Control

This section describes how to configure Recovery Appliance replication using the Replication page in Cloud Control.

Prerequisites

Your environment must meet the following prerequisites:

- The upstream and downstream Recovery Appliance can communicate with each other over the network.
- Every protected database whose backup data will be replicated must be enrolled with the upstream Recovery Appliance.
- The downstream Recovery Appliance must be started and configured to receive backups.

Assumptions

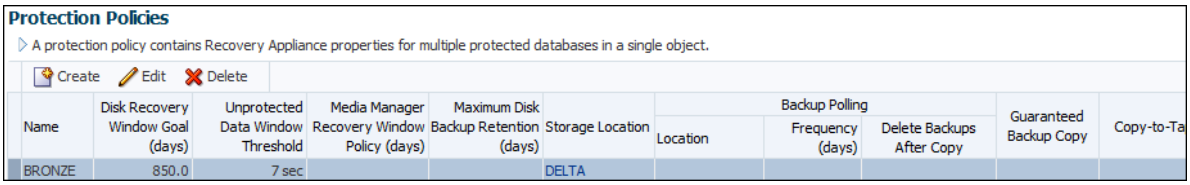
Assume that the following statements are true of your Recovery Appliance environment:

- The protected databases `orcl11` and `orcl12` back up to upstream Recovery Appliance `ZDLRA Boston`.
- You want `ZDLRA Boston` to replicate to downstream Recovery Appliance `ZDLRA Des Moines`.
- A replication user account named `repuser_from_boston` exists on the *downstream* Recovery Appliance (`ZDLRA Des Moines`).
- A virtual private catalog account named `vpc_boston1` exists on the *upstream* Recovery Appliance (`ZDLRA Boston`).
- You know the credentials for the operating system user who owns the upstream Recovery Appliance (`ZDLRA Boston`) database installation.

To configure Recovery Appliance replication:

1. On the *downstream* Recovery Appliance, access the Create Protection Policy page as `RASYS`, as described in "Accessing the Create Protection Policy Page in Cloud Control".

Figure 8-8 Protection Policies Page



In this example, you access the Create Protection Policy page for `ZDLRA Des Moines`, which is the downstream Recovery Appliance.

2. Create a replication protection policy, as described in "Creating a Protection Policy Using Cloud Control".

Figure 8-9 Create Protection Policy Page

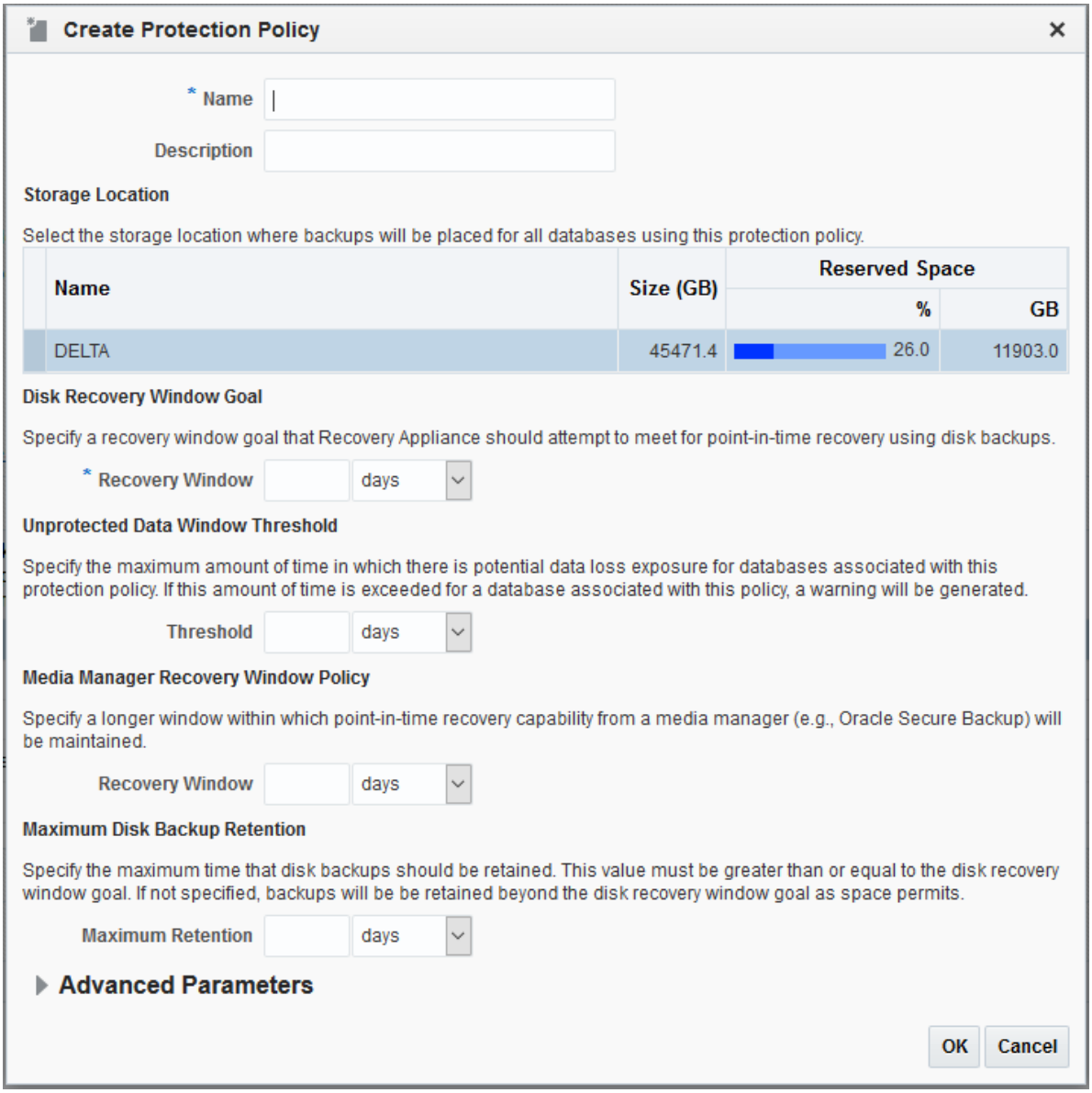
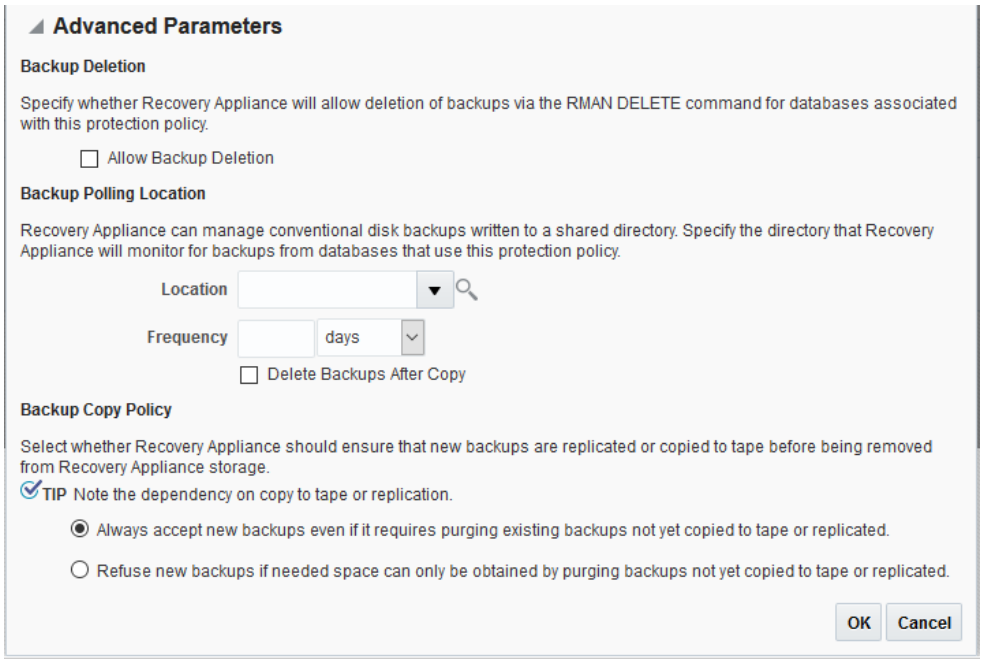


Figure 8-10 Protection Policy Advanced Parameters

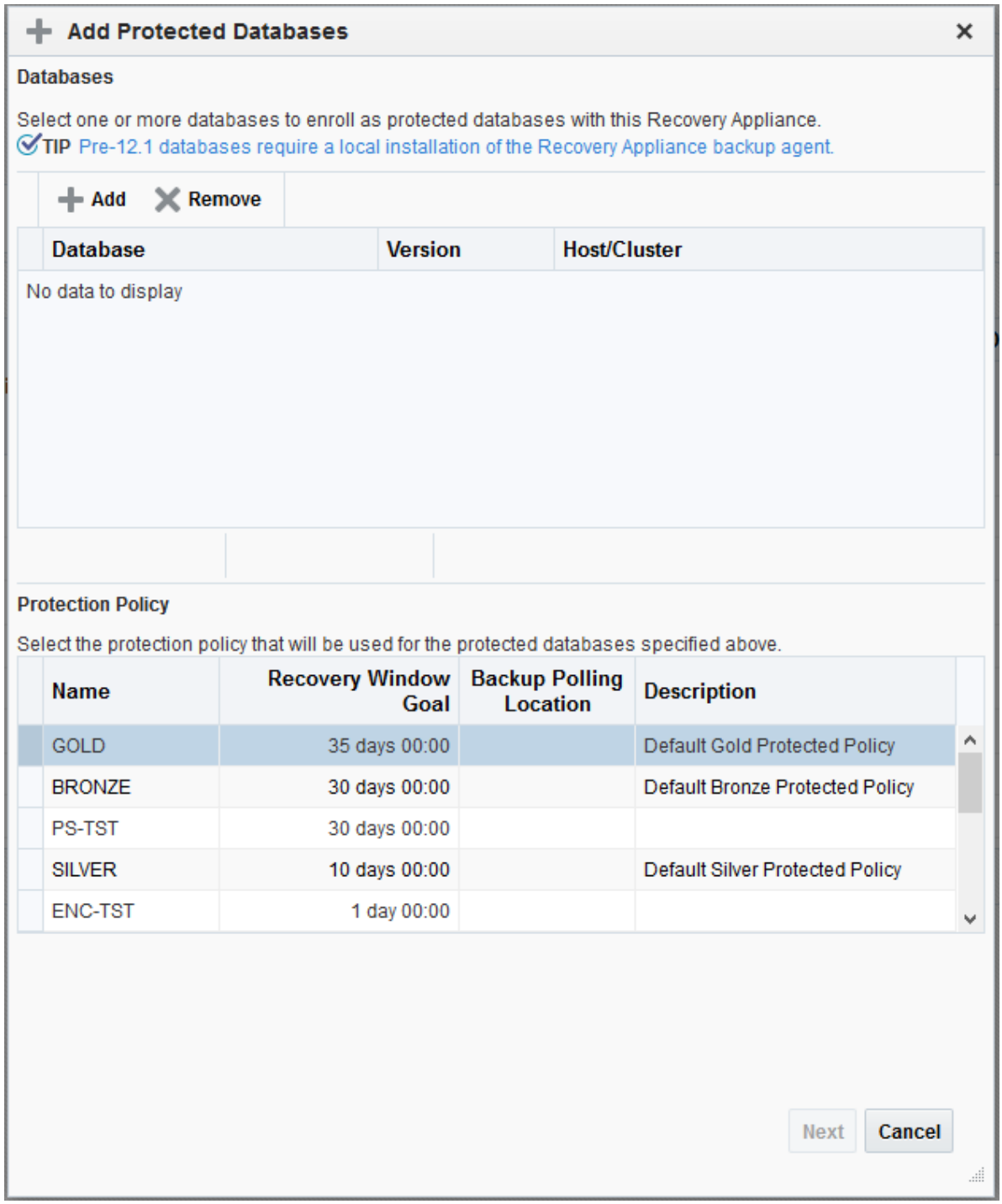


In this example, you create a policy named `reppolicy_ds_gold`.

3. Add the databases and grant access to the replication user account on the downstream Recovery Appliance, as described in "Enrolling Protected Databases Using Cloud Control".

> **Note:**
>
> You are not required to run the `REGISTER DATABASE` command for the protected databases in the *downstream* recovery catalog, which is the last step of enrollment.

Figure 8-11 Add Protected Databases Page



In this example, add protected databases `orcl11` and `orcl12`, and grant `repuser_from_boston` access to these databases.

4. On the *upstream* Recovery Appliance, go to the Recovery Appliance Home page.
5. To create a new protection policy for replication, complete the following substeps; otherwise, skip to Step 6.

a. Access the Create Protection Policy page as `RASYS`, as described in "Accessing the Create Protection Policy Page in Cloud Control".

   In this example, you access the Create Protection Policy page for `ZDLRA Boston`, which is the upstream Recovery Appliance.

b. Create a replication protection policy, as described in "Creating a Protection Policy Using Cloud Control".

   In this example, you create a policy named `reppolicy_us_gold`.

6. Add the databases to an existing protection policy, and grant access to the upstream virtual private catalog account, as described in "Enrolling Protected Databases Using Cloud Control".

   In this example, add protected databases `orcl11` and `orcl12` to the protection policy `reppolicy_us_gold`, and grant `vpc_boston1` access to these databases.

7. From the **Recovery Appliance** menu, select **Replication**.

   The Recovery Appliance Login page appears.

8. Enter your login credentials, and then click **Login**.

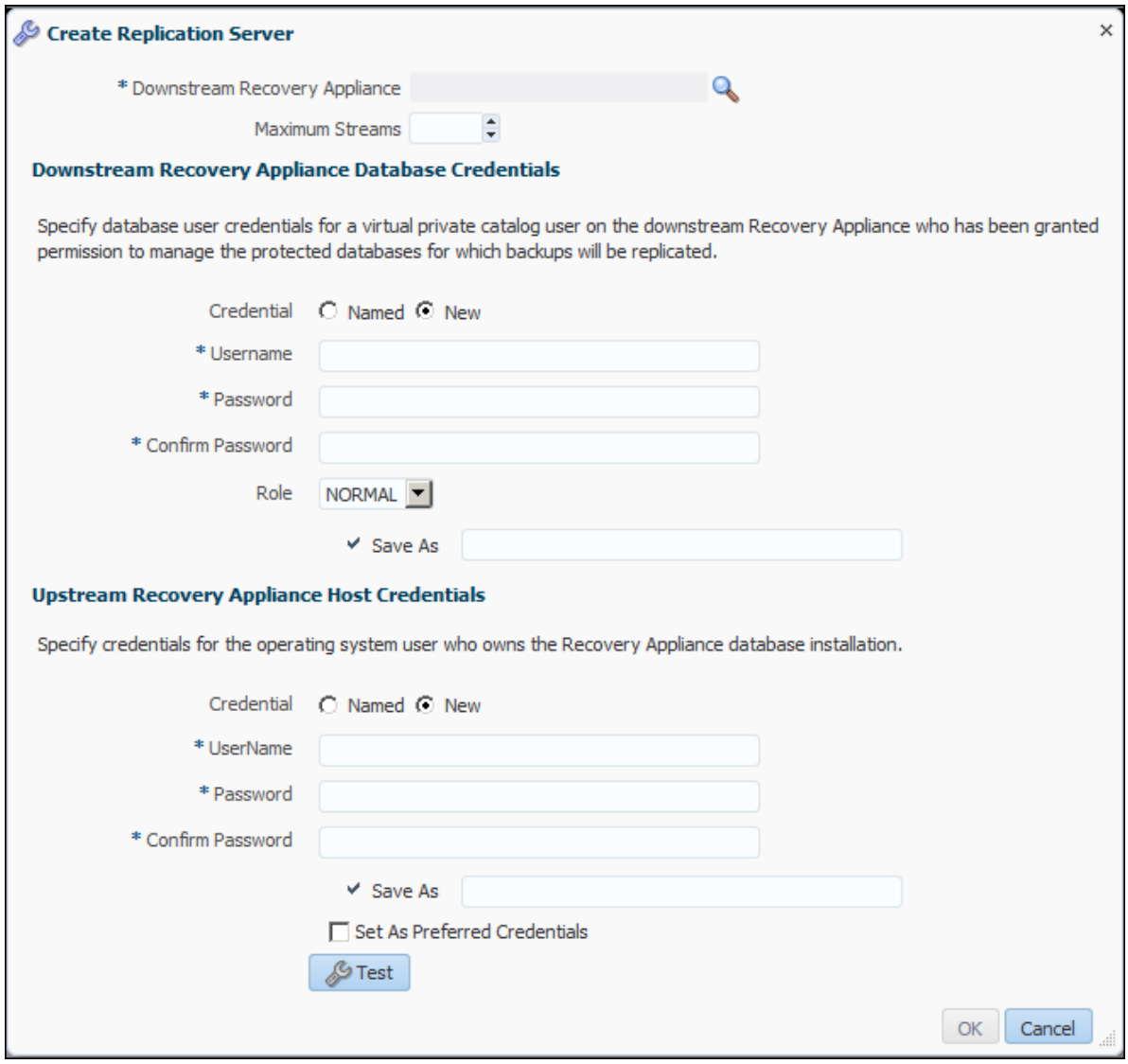
   The Replication page appears.

9. Click **Create Replication Server**.

> **Caution:**
>
> If you create the replication server on the upstream Recovery Appliance *before* the downstream Recovery Appliance has added the databases to a protection policy and granted database access, then replication will not work.

   The Create Replication Server page appears.

Figure 8-12 Create Replication Server Page



10. Enter values as follows, and then click **OK**:
    - In the **Downstream Recovery Appliance** field, click the magnifying glass, and then from the list of discovered targets, select the Recovery Appliance that you want to configure in the downstream role.

      For example, select **ZDLRA Des Moines**.
    - In the Downstream Recovery Appliance Database Credentials section, specify credentials for a virtual private catalog account on the downstream Recovery Appliance.

      > **Note:**
      >
      > This catalog account must have been granted permission to manage the replicated backups on the downstream Recovery Appliance. See "Creating Virtual Private Catalog Accounts".

For example, enter `vpc_des_moines1`.

- In the Upstream Recovery Appliance Database Credentials section, specify credentials for the operating system user who owns the upstream Recovery Appliance database installation.

An Information message appears showing the job submission ID.

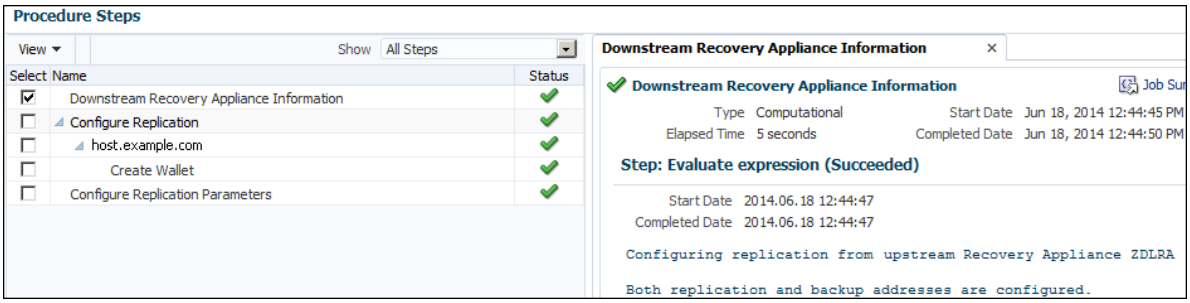
11. To view a job report, click the job submission ID.

    The Procedure Activity page appears.

12. In the Procedure Steps section, select any or all of the steps to obtain detailed information.

    Figure 8-13 shows a snippet of the Procedure Steps page, with the Downstream Recovery Appliance Information step selected.

Figure 8-13 Procedure Steps



13. To return to the home page for the Recovery Appliance, click **Targets**, **Recovery Appliances**, and then the name of the upstream Recovery Appliance.
14. From the **Recovery Appliance** menu, select **Replication**.

    The Replication page appears.

15. Select the replication server configuration that you created in Step 9, and then click **Add Protection Policy**.

    The Add Protection Policy page appears.

16. Select the replication policy that you created in Step 6, and then click **OK**.

    In this example, select the policy named `reppolicy_us_gold`.

    The Replication page now shows the replication server configuration with the protection policy attached.

> **See Also:**
>
> Cloud Control online help for more information about the replication pages

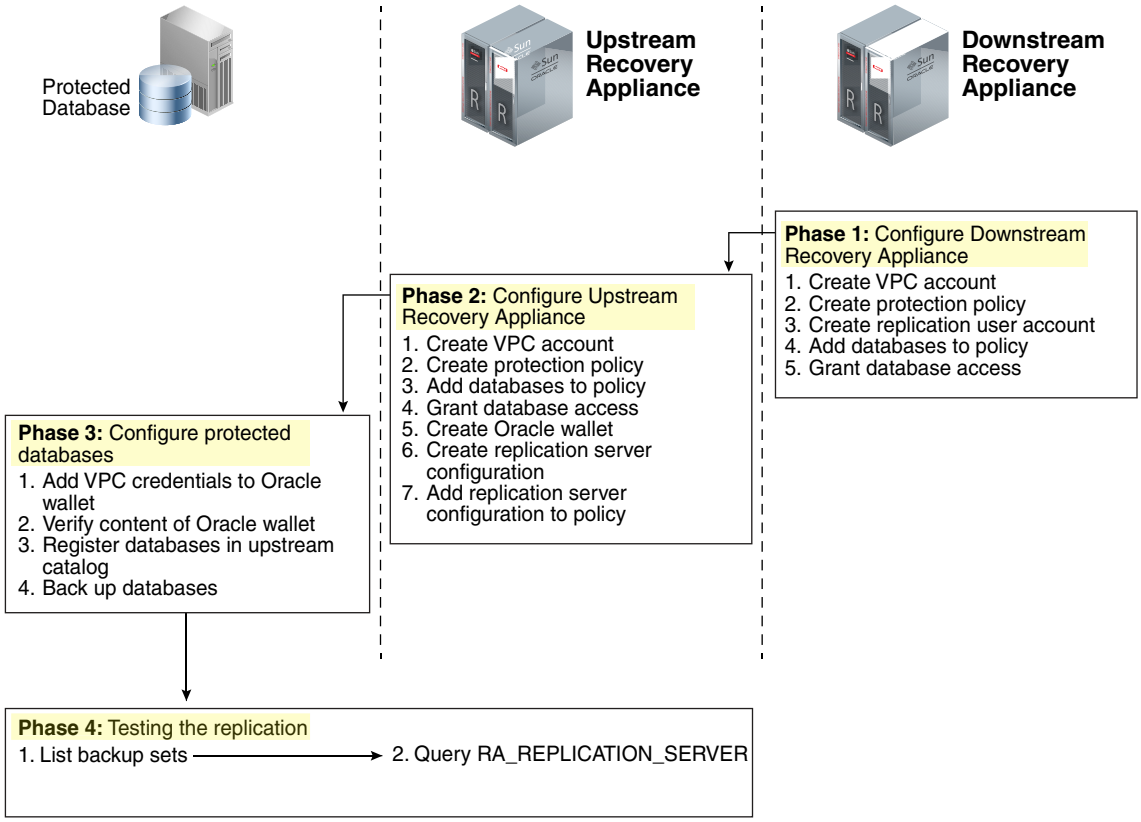
Configuring Recovery Appliance for Replication Using DBMS_RA

This section explains how to configure replication using command-line tools. The basic work flow is as follows:

1. Configure the downstream Recovery Appliance, as described in "Configuring a Downstream Recovery Appliance for Replication Using DBMS_RA".
2. Configure the upstream Recovery Appliance, as described in "Configuring an Upstream Recovery Appliance for Replication Using DBMS_RA".
3. Configure the protected databases involved in the replication, as described in "Configuring a Protected Database for Recovery Appliance Replication".
4. Test the replication, as described in "Testing a Recovery Appliance Replication Server Configuration".

Figure 8-14 is a graphic illustration of the configuration phases.

Figure 8-14 Overview of Manual Configuration for Replication



Assumptions for the Replication Examples

In the replication tasks that follow, assume that the following conditions are true:

- You back up databases `orcl11` and `orcl12` to a Recovery Appliance named `ZDLRA Boston` that you want to configure in the *upstream* replication role.
- You intend to use `ZDLRA Des Moines` as the *downstream* Recovery Appliance.
- On the downstream Recovery Appliance, you intend to create a Recovery Appliance user account named `repuser_from_boston`. This account is the replication user account.

> **Note:**
>
> The naming convention for this account uses the Recovery Appliance from where the backups will be replicated—in this case, `ZDLRA Boston`. In names of the protection policies in our examples, we use `us` for upstream and `ds` for downstream.

- On the downstream Recovery Appliance, you intend to create a protection policy named `reppolicy_ds_gold`. This policy is exclusively for use by replication.
- On the downstream Recovery Appliance, you intend to create a virtual private catalog account named `vpc_des_moines1`. RMAN uses this account to back up and restore databases `orcl11` and `orcl12`.
- On the upstream Recovery Appliance, you intend to create a protection policy named `reppolicy_us_gold`. This policy is exclusively for use by replication.
- On the upstream Recovery Appliance, you intend to create a virtual private catalog account named `vpc_boston1`. RMAN uses this account to back up and restore databases `orcl11` and `orcl12`.

Configuring a Downstream Recovery Appliance for Replication Using DBMS_RA

This section explains how to configure a downstream Recovery Appliance.

> **Note:**
>
> When a Recovery Appliance has both the upstream and downstream roles, these instructions pertain to the role of a downstream Recovery Appliance only.

Task 1: Create a virtual private catalog account on the downstream Recovery Appliance

When backing up or restoring protected databases, RMAN uses this account to connect to the recovery catalog on the downstream Recovery Appliance.

This task assumes that you want to create a virtual private catalog account named `vpc_des_moines1` on the downstream Recovery Appliance.

To create virtual private catalog account:

- Follow the instructions in "Creating Virtual Private Catalog Accounts".

For example, execute the following statement to create user account `vpc_des_moines1`:

```
# ./racli add vpc_user --username=vpc_des_moines1
```

Enter the password for `vpc_des_moines1` user when prompted.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* to learn more about virtual private catalogs

Task 2: Create a replication protection policy on the downstream Recovery Appliance

To create a protection policy specifying recovery windows and other properties of backups replicated to this downstream Recovery Appliance, execute `DBMS_RA.CREATE_PROTECTION_POLICY`.

This task assumes that you create a `reppolicy_ds_gold` policy to protect the `orcl11` and `orcl12` databases. You will later associate this policy with a Recovery Appliance.

To create a replication protection policy:

1. With SQL*Plus or SQL Developer, connect to the downstream Recovery Appliance database as `RASYS`.
2. Create a protection policy with the `DBMS_RA.CREATE_PROTECTION_POLICY` procedure.

   For example, execute the following PL/SQL program:

```
BEGIN
  DBMS_RA.CREATE_PROTECTION_POLICY (
    protection_policy_name => 'reppolicy_ds_gold',
    description            => 'For protected dbs in gold tier',
    storage_location_name  => 'delta',
    recovery_window_goal   => INTERVAL '28' DAY,
    guaranteed_copy        => 'NO');
END;
```

> **See Also:**
>
> - "Creating a Protection Policy Using DBMS_RA"
> - "CREATE_PROTECTION_POLICY" for definitions of procedure arguments

Task 3: Create a replication user account on the downstream Recovery Appliance

When you configure a *downstream* Recovery Appliance to replicate backups for a protected database, you must create a replication user account that the *upstream* Recovery Appliance uses to log in to this downstream Recovery Appliance. The

credentials for the user on the *downstream* Recovery Appliance are stored in the Oracle wallet of the *upstream* Recovery Appliance (see "Task 5: Create an Oracle wallet on the upstream Recovery Appliance").

> **Note:**
>
> For ease of administration, Oracle recommends that you create a replication user account exclusively for use with Recovery Appliance replication, and that you create a separate replication user account for each upstream appliance.

This task assumes that you want to create an account named `repuser_from_boston` that the upstream Recovery Appliance uses to authenticate on this Recovery Appliance.

To create a replication user account:

1. With SQL*Plus or SQL Developer, connect to the downstream Recovery Appliance database as `SYSTEM` or any user with the `DBA` role.
2. Create the replication user account.

   For example, execute the following SQL statements to create the `repuser_from_boston` database user account and grant it `CREATE SESSION` privileges:

   ```
   CREATE USER repuser_from_boston IDENTIFIED BY ********;
   GRANT CREATE SESSION TO repuser_from_boston;
   ```

   > **Note:**
   >
   > Oracle recommends that you use a highly complex password to enhance security. You add this password and user name to the Oracle wallet in a subsequent step. After you save these credentials in the wallet, you will not need to enter the password manually again.

> **See Also:**
>
> *Oracle Database Security Guide* to learn how to create database user accounts

Task 4: Add databases to the protection policy on the downstream Recovery Appliance

To add the protected databases to the replication protection policy, execute `DBMS_RA.ADD_DB`. You must also specify the amount of disk space reserved for each protected database.

This task assumes that you want to add databases `orcl11` and `orcl12` to the `reppolicy_ds_gold` protection policy that you created in Task 2: Create a replication

protection policy on the downstream Recovery Appliance and allocate 128 GB of reserved space for each protected database.

To add databases to a protection policy:

1. With SQL*Plus or SQL Developer, connect to the downstream Recovery Appliance database as `RASYS`.
2. Add metadata for each protected database using the `DBMS_RA.ADD_DB` procedure.

   For example, execute the following PL/SQL programs:

```
BEGIN
  DBMS_RA.ADD_DB (
    db_unique_name         => 'orcl11',
    protection_policy_name => 'reppolicy_ds_gold',
    reserved_space         => '128G');
END;
BEGIN
  DBMS_RA.ADD_DB (
    db_unique_name         => 'orcl12',
    protection_policy_name => 'reppolicy_ds_gold',
    reserved_space         => '128G');
END;
```

> **See Also:**
>
> - "Adding Protected Database Metadata Using DBMS_RA"
> - "ADD_DB"

Task 5: Grant database access on the downstream Recovery Appliance

Execute `DBMS_RA.GRANT_DB_ACCESS` to grant protected database access to the following database accounts:

- The virtual private catalog account created in "Task 1: Create a virtual private catalog account on the downstream Recovery Appliance"
- The replication user account created in "Task 3: Create a replication user account on the downstream Recovery Appliance"

To grant protected database access to the replication and catalog accounts:

1. With SQL*Plus or SQL Developer, connect to the downstream Recovery Appliance database as `RASYS`.
2. For each protected database that will send backups to the upstream Recovery Appliance that must authenticate with this account, grant privileges to the replication user.

   The following example grants the replication user `repuser_from_boston` the required privileges on protected databases `orcl11` and `orcl12`:

```
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'repuser_from_boston',
    db_unique_name => 'orcl11');
END;
```

```
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'repuser_from_boston',
    db_unique_name => 'orcl12');
END;
```

3. For each protected database on each upstream Recovery Appliance that will authenticate with this account, grant privileges to the virtual private catalog account.

   The following example grants the recovery catalog account `vpc_des_moines1` the required privileges on protected databases `orcl11` and `orcl12`:

```
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'vpc_des_moines1',
    db_unique_name => 'orcl11');
END;
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'vpc_des_moines1',
    db_unique_name => 'orcl12');
END;
```

> **See Also:**
>
> - "Granting Database Access to a Recovery Appliance Account Using DBMS_RA"
> - "GRANT_DB_ACCESS"

Configuring an Upstream Recovery Appliance for Replication Using DBMS_RA

This section explains how to configure an upstream Recovery Appliance. This section assumes that you have completed the steps in "Configuring a Downstream Recovery Appliance for Replication Using DBMS_RA".

> **Note:**
>
> When a Recovery Appliance has both the upstream and downstream roles, these instructions pertain to the upstream role only.

Task 1: Create a virtual private catalog account on the upstream Recovery Appliance

When backing up protected databases, RMAN uses this account to connect to the recovery catalog on the upstream Recovery Appliance.

This section assumes that you want to create a virtual private catalog account named `vpc_boston1` on the upstream Recovery Appliance.

To create virtual private catalog account:

- Follow the instructions in "Creating Virtual Private Catalog Accounts".

  For example, execute the following statement to create user account `vpc_boston1`:

  ```
  # ./racli add vpc_user --username=vpc_boston1
  ```

  Enter the password for `vpc_boston1` user when prompted.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* to learn more about virtual private catalogs

Task 2: Create a protection policy on the upstream Recovery Appliance

Execute `DBMS_RA.CREATE_PROTECTION_POLICY` to create a protection policy to specify the disk recovery windows and other properties of backups to this upstream Recovery Appliance. The upstream Recovery Appliance replicates these backups to its downstream Recovery Appliance.

This task assumes that you create a `reppolicy_us_gold` policy to protect the `orcl11` and `orcl12` databases. In the next task, you associate this protection policy with the protected databases.

To create a protection policy for Recovery Appliance replication:

1. With SQL*Plus or SQL Developer, connect to the upstream Recovery Appliance metadata database as `RASYS`.
2. Create each protection policy with the `DBMS_RA.CREATE_PROTECTION_POLICY` procedure.

   For example, execute the following PL/SQL program:

   ```
   BEGIN
     DBMS_RA.CREATE_PROTECTION_POLICY (
       protection_policy_name => 'reppolicy_us_gold',
       description            => 'For protected dbs in gold tier',
       storage_location_name  => 'delta',
       recovery_window_goal   => INTERVAL '28' DAY,
       guaranteed_copy        => 'NO');
   END;
   ```

> **See Also:**
>
> - "Creating a Protection Policy Using DBMS_RA"
> - "CREATE_PROTECTION_POLICY" for definitions of procedure arguments

Task 3: Add databases to the protection policy on the upstream Recovery Appliance

To add the protected databases to the replication protection policy, execute the `DBMS_RA.ADD_DB` procedure. You must also specify the amount of disk space reserved for each protected database.

This task assumes that you want to add databases `orcl11` and `orcl12` to the `reppolicy_us_gold` protection policy that you created in "Task 2: Create a protection policy on the upstream Recovery Appliance", and allocate 128 GB of reserved space for each protected database.

To add databases to a protection policy:

1. With SQL*Plus or SQL Developer, connect to the upstream Recovery Appliance metadata database as `RASYS`.
2. Add metadata for each protected database using the `DBMS_RA.ADD_DB` procedure.

   For example, execute the following PL/SQL programs:

```
BEGIN
  DBMS_RA.ADD_DB (
    db_unique_name         => 'orcl11',
    protection_policy_name => 'reppolicy_us_gold',
    reserved_space         => '128G');
END;
BEGIN
  DBMS_RA.ADD_DB (
    db_unique_name         => 'orcl12',
    protection_policy_name => 'reppolicy_us_gold',
    reserved_space         => '128G');
END;
```

See Also:

- "Adding Protected Database Metadata Using DBMS_RA"
- "ADD_DB"

Task 4: Grant database access to the virtual private catalog account on the upstream Recovery Appliance

To grant protected database access to the *upstream* catalog account created in "Task 1: Create a virtual private catalog account on the upstream Recovery Appliance", execute `DBMS_RA.GRANT_DB_ACCESS`. This step makes it possible for RMAN to connect to the recovery catalog when it backs up or restores the protected databases.

To grant protected database access to the virtual private catalog:

1. With SQL*Plus or SQL Developer, connect to the upstream Recovery Appliance metadata database as `RASYS`.
2. For each protected database whose backups will be replicated, grant privileges to the virtual private catalog account.

The following example grants the catalog account `vpc_boston1` the required privileges on protected databases `orcl11` and `orcl12`:

```
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'vpc_boston1',
    db_unique_name => 'orcl11');
END;
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'vpc_boston1',
    db_unique_name => 'orcl12');
END;
```

See Also:

- "Granting Database Access to a Recovery Appliance Account Using DBMS_RA"
- "GRANT_DB_ACCESS"

Task 5: Create an Oracle wallet on the upstream Recovery Appliance

On the upstream Recovery Appliance, use the `mkstore` utility to create an Oracle auto-login wallet and add the replication user credentials created in "Task 3: Create a replication user account on the downstream Recovery Appliance". The upstream Recovery Appliance requires these credentials when it logs in to a downstream appliance. Each stored credential contains the name and verifier of a Recovery Appliance user account.

Note:

If an existing wallet is an auto-login wallet (one that does not require you to enter a password each time the wallet is accessed), then you may use it. An Oracle wallet has a file extension of `*.sso`. To use an existing Oracle wallet, skip Step 2 below.

This task assumes the following:

- You want to create the Oracle wallet used for replication in the `/dbfs_repdbfs/REPLICATION` directory on the upstream Recovery Appliance host.
- You want to add credentials for replication user `repuser_from_boston`.

To create an Oracle Wallet on the upstream Recovery Appliance:

1. Log in to the upstream Recovery Appliance host as the operating system user who installed Recovery Appliance or as a member of that user's operating system group.
2. To create the Oracle wallet, run the following command, where *`wallet_location`* is an existing directory on the upstream Recovery Appliance in which to store the wallet:

```
mkstore -wrl wallet_location -createALO
```

For example, the following command creates an auto-login wallet in the `/dbfs_repdbfs/REPLICATION` directory:

```
mkstore -wrl /dbfs_repdbfs/REPLICATION -createALO
```

The `mkstore` utility creates a file named `cwallet.sso` in the designated location.

3. To add the credentials, run the following command:

```
mkstore -wrl wallet_location -createCredential serv_name ds_rep_user pwd
```

The placeholders are defined as follows:

- *wallet_location* is the directory in which to create the wallet. The directory must exist.
- *serv_name* is an Oracle network service name that you use in an EZ Connect descriptor to identify the downstream Recovery Appliance on an Oracle network.
- *ds_rep_user* is the user name of the replication user account on the downstream Recovery Appliance.
- *pwd* is the secure password of the replication user on the downstream Recovery Appliance.

For example, the following command adds credentials for the net service name `radsm01repl-scan.acme.com` using port `1522` and a database name of `zdlradsm`, and the replication user name `repuser_from_boston`:

```
mkstore -wrl /dbfs_repdbfs/REPLICATION -createCredential \
"radsm01repl-scan.acme.com:1522/zdlradsm" "repuser_from_boston" "pwd"
```

4. Verify that credentials were properly added for all users by running the following command, which lists the credentials in the Oracle wallet (no passwords or verifiers are displayed):

```
mkstore -wrl wallet_location -listCredential
```

For example, the following command lists the credentials in the Oracle wallet stored in `/dbfs_repdbfs/REPLICATION`:

```
mkstore -wrl /dbfs_repdbfs/REPLICATION -listCredential

Oracle Secret Store Tool : Version 12.1.0.1 Copyright (c) 2004, 2012, Oracle
and/or its affiliates. All rights reserved.
List credential (index: connect_string username)
1: radsm01repl-scan1.acme.com:1522/zdlradsm repuser_from_boston
```

See Also:

- *Oracle Database Net Services Administrator's Guide* for the location of `tnsnames.ora`
- *Oracle Database Net Services Administrator's Guide* to learn more about net service names

Task 6: Create the replication server configuration on the upstream Recovery Appliance

For each downstream Recovery Appliance to which this upstream Recovery Appliance will replicate, create a replication server configuration by executing `DBMS_RA.CREATE_REPLICATION_SERVER`.

> **Caution:**
>
> If you run `CREATE_REPLICATION_SERVER` on the upstream Recovery Appliance *before* the downstream Recovery Appliance has added the databases to a protection policy (`ADD_DB`) and granted database access (`GRANT_DB_ACCESS`), then an `ORA-600` error can result.

This task assumes the following:

- You want to create a replication server configuration named `zdlradsm_rep`.

  > **Note:**
  >
  > The replication server configuration name is arbitrary. However, Oracle recommends that you use the service name of the downstream Recovery Appliance, which is also the database name (`zdlradsm` in this example) followed by `_rep`.

- You want the upstream Recovery Appliance to log in to its downstream Recovery Appliance using the replication account `repuser_from_boston`. You created this account in "Task 3: Create a replication user account on the downstream Recovery Appliance".
- The configuration uses the net service name `radsm01repl-scan.acme.com:1522/zdlradsm` that you stored in the Oracle wallet created in "Task 5: Create an Oracle wallet on the upstream Recovery Appliance".
- The Oracle wallet is stored in `/dbfs_repdbfs/REPLICATION`.
- The file name of the Recovery Appliance Backup Module, which is preinstalled on every Recovery Appliance, is `/u01/app/oracle/product/12.1.0.2/dbh1/lib/libra.so`. The module functions as an SBT media management library. RMAN references this module when allocating or configuring a channel for backup to the Recovery Appliance (see "Configuring a Protected Database for Recovery Appliance Replication").

To create a replication server configuration:

1. With SQL*Plus or SQL Developer, connect to the upstream Recovery Appliance metadata database as `RASYS`.
2. Run the `DBMS_RA.CREATE_REPLICATION_SERVER` procedure for each downstream Recovery Appliance.

   The following example creates the replication server configuration named `zdlradsm_rep` for the downstream Recovery Appliance named `ZDLRA Des Moines`:

```
BEGIN
  DBMS_RA.CREATE_REPLICATION_SERVER (
    replication_server_name => 'zdlradsm_rep',
    sbt_so_name      => '/u01/app/oracle/product/12.1.0.2/dbh1/lib/libra.so',
    catalog_user_name       => 'RASYS',
    wallet_alias            => 'radsm01repl-scan.acme.com:1522/zdlradsm',
    wallet_path             => '/dbfs_repdbfs/REPLICATION');
END;
```

3. Confirm the creation of the replication server configuration.

   For example, run the following query:

```
SELECT COUNT(*) should_be_one
FROM   RA_REPLICATION_SERVER
WHERE  REPLICATION_SERVER_NAME = 'ZDLRADSM_REP';

SHOULD_BE_ONE
-------------
1
```

   If the configuration was created correctly, then the return value is 1.

> **See Also:**
>
> - "CREATE_REPLICATION_SERVER" for procedure argument descriptions
> - *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* to learn more about the Recovery Appliance Backup Module
> - *Oracle Database Backup and Recovery User's Guide* for a list of valid client configuration file parameters and their definitions

Task 7: Associate the upstream Recovery Appliance with a protection policy

Specify the downstream Recovery Appliances to which each protected database replicates by assigning the replication server configuration to a protection policy. When this task is completed, Recovery Appliance replication is enabled.

> **Note:**
>
> You can assign multiple replication server configurations to a protection policy.

This task assumes the following:

- You want to use the replication server configuration named `zdlradsm_rep`, which you created in "Task 6: Create the replication server configuration on the upstream Recovery Appliance".
- You want to add the replication server configuration to protection policy `reppolicy_us_gold`, which you created in "Task 2: Create a protection policy on the upstream Recovery Appliance".

To associate a replication server configuration with a protection policy:

1. Ensure you are connected to the Recovery Appliance metadata database as the Recovery Appliance administrator.
2. Run the `DBMS_RA.ADD_REPLICATION_SERVER` procedure for each combination of protection policy and replication server configuration.

   For example, execute the following PL/SQL program:

```
BEGIN
  DBMS_RA.ADD_REPLICATION_SERVER (
   replication_server_name => 'zdlradsm_rep',
   protection_policy_name  => 'reppolicy_us_gold');
END;
```

> **See Also:**
>
> "ADD_REPLICATION_SERVER"

Configuring a Protected Database for Recovery Appliance Replication

Each protected database that participates in a Recovery Appliance replication environment must be correctly configured. For example, for each protected database, you must:

- Add the Oracle wallet credentials for the virtual private catalog owner on the upstream *and* downstream Recovery Appliances to the Oracle wallet.

  > **Note:**
  >
  > The replication configuration does not require you to add the downstream credentials. However, if the upstream Recovery Appliance were inaccessible, and if RMAN tried to restore backups from the downstream Recovery Appliance, then RMAN would need to connect directly to the virtual private catalog in the downstream Recovery Appliance. In this case, the Oracle wallet would require the downstream credentials.

- Verify the content of the Oracle wallet.
- Register the database in the virtual private catalog of the *upstream* Recovery Appliance.
- Back up the protected database, making sure to specify the correct Oracle wallet location when allocating the RMAN channel.

To learn how to configure protected databases, see *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*.

Testing a Recovery Appliance Replication Server Configuration

For every protected database involved in a replication scheme, use the following procedure to test replication from an upstream Recovery Appliance to all downstream

Recovery Appliances. You can repeat this procedure to test each replication path of a complex replication topology.

This section assumes the following:

- You want to test the replication of backups of `orcl11` from `ZDLRA Boston`, which is the upstream Recovery Appliance, to `ZDLRA Des Moines`, which is the downstream Recovery Appliance.
- `ZDLRA Boston` also backs up to tape.

To test the replication of a protected database:

1. Start RMAN, and connect to a protected database as `TARGET`, and the virtual private catalog on the upstream Recovery Appliance as `CATALOG`.

   For example, enter the following command at the system prompt to connect to `orcl11` as `TARGET` and `zdlra_boston` as `CATALOG`:

   ```
   rman TARGET ra_admin@orcl11 CATALOG /@zdlra01bosingest-scan1.acme.com:1521/
   zdlrabos:dedicated
   ```

2. List the backup sets, and confirm that the backups exist on the upstream and downstream Recovery Appliances.

   For example, run the following command (sample output included):

   ```
   RMAN> LIST BACKUPSET;
   .
   .
   .

   BS Key  Size
   ------- ----------
   54746   224.25M

     List of Archived Logs in backup set 54746
     Thrd Seq     Low SCN    Low Time            Next SCN   Next Time
     ---- ------- ---------- ------------------- ---------- ---------
     1    17854   153525644  2014/07/01 12:59:40 153545145  2014/07/01 13:00:34
     1    17855   153545145  2014/07/01 13:00:34 153564529  2014/07/01 13:01:36
     1    17856   153564529  2014/07/01 13:01:36 153585644  2014/07/01 13:02:26
     1    17857   153585644  2014/07/01 13:02:26 153606722  2014/07/01 13:03:18
     1    17858   153606722  2014/07/01 13:03:18 153629480  2014/07/01 13:04:11
     1    17859   153629480  2014/07/01 13:04:11 153651278  2014/07/01 13:05:05
     1    17860   153651278  2014/07/01 13:05:05 153672263  2014/07/01 13:05:59

     Backup Set Copy #1 of backup set 54746
     Device Type Elapsed Time Completion Time     Compressed Tag
     ----------- ------------ ------------------- ---------- ---
     SBT_TAPE    02:52:20     2014/07/01 13:14:46 NO         TAG20140701T131434

       List of Backup Pieces for backup set 54746 Copy #1
       BP Key  Pc# Status      Media                    Piece Name
       ------- --- ----------- ---------------------- ----------
       54747   1   AVAILABLE   Oracle Recovery Appliance (ZDLRA Boston)
   4qpca79s_1_1_DB1211LG

     Backup Set Copy #2 of backup set 54746
     Device Type Elapsed Time Completion Time     Compressed Tag
     ----------- ------------ ------------------- ---------- ---
     SBT_TAPE    02:52:20     2014/07/01 16:06:56 NO         TAG20140701T131434
   ```

```
    List of Backup Pieces for backup set 54746 Copy #2
    BP Key  Pc# Status      Media                    Piece Name
    ------- --- ----------- ---------------------- ----------
    55019   1   AVAILABLE   Oracle Recovery Appliance (ZDLRA Des Moines)
RA_SBT_54971_4qpca79s_1_2_54746_1
.
.
.
```

In the preceding output, backup set `54746` has two copies. Copy #1 resides on `ZDLRA Boston`, which is the upstream Recovery Appliance, and copy #2 resides on `ZDLRA Des Moines`, which is the downstream Recovery Appliance.

3. With SQL*Plus or SQL Developer, connect to the *upstream* Recovery Appliance as `RASYS`.
4. Confirm that the upstream Recovery Appliance has the correct replication status.

   For example, query `RA_REPLICATION_SERVER`, which should show a state of `RUNNING` for the replication server configuration that you created in "Task 6: Create the replication server configuration on the upstream Recovery Appliance":

```
SELECT REPLICATION_SERVER_NAME AS "RS_NAME",
       REPLICATION_SERVER_STATE AS "RS_STATE",
FROM   RA_REPLICATION_SERVER;

RS_NAME         RS_STATE
--------------- --------
ZDLRADSM_REP RUNNING
```

   If all preceding tests reveal the expected results, then the upstream Recovery Appliance is replicating backups of this protected database successfully.

9

Implementing Additional High Availability Strategies

Besides replication, other high availability strategies can be used with Recovery Appliance to increase protection against data loss in certain scenarios.

The Oracle Maximum Availability Architecture (MAA) best practice to protect the appliance against site disasters and system outages is to implement a disaster recovery strategy using Recovery Appliance replication. With a replica appliance, protected database backup, redo, and restore operations continue uninterrupted, preserving complete data protection.

If your organization does not have a disaster recovery strategy or if you would like to add local system high availability to your existing disaster recovery strategy, you can use the Backup and Redo Failover feature of Recovery Appliance. This feature is available starting with Recovery Appliance software update 12.1.1.1.8.

Another component of a high availability (HA) and disaster recovery solution is Oracle Data Guard. Oracle Data Guard minimizes service interruption and resulting data loss by maintaining a synchronized standby database for the protected database.

See Also:

- "Managing Temporary Outages with a Backup and Redo Failover Strategy" for information and instructions for configuring Backup and Redo Failover
- "Maximum Availability: Recovery Appliance with Oracle Data Guard" for information about Oracle Data Guard
- "Replicating Backups with Recovery Appliance " for information about Recovery Appliance replication

Managing Temporary Outages with a Backup and Redo Failover Strategy

Backup and Redo Failover is a high availability feature that allows protected databases to temporarily direct backups and redo to an alternate Recovery Appliance when the primary Recovery Appliance experiences an outage or requires planned maintenance. This allows protected database backups and redo to continue uninterrupted and preserves complete data protection. It also prevents the local archived log destinations of the database from filling up and impacting the database, which can occur with no alternate backup destination.

Overview of the Backup and Redo Failover Feature

In an environment where Backup and Redo Failover is configured, a protected database sends backups and redo to a primary Recovery Appliance under normal circumstances. When that appliance is unavailable, the protected database sends backups and redo to an alternate Recovery Appliance until service on the primary is restored.

The alternate appliance does not create virtual full backups from the temporary backups it receives; it only stores the backup pieces (incremental and archived log backups). When the primary appliance is back online and operational, the alternate appliance forwards all temporary backups to the primary appliance, which uses them to create the corresponding virtual full backups. After all virtual full backups are created, the protected database resumes sending backups and redo to the primary appliance. The alternate appliance deletes the temporary backup pieces from local storage only after they are successfully forwarded to the primary appliance.

Configuring Backup and Redo Failover

This section explains how to configure Backup and Redo Failover. The basic work flow is as follows:

1. Configure the primary Recovery Appliance, as described in "Configuring the Primary Recovery Appliance for Backup and Redo Failover".
2. Configure the alternate Recovery Appliance, as described in "Configuring the Alternate Recovery Appliance for Backup and Redo Failover".
3. Configure replication from the alternate Recovery Appliance to the primary Recovery Appliance, as described in "Configuring Replication for Backup and Redo Failover".
4. Configure the protected database to send backups, as described in "Configuring the Protected Database for Backup and Redo Failover".

Configuring the Primary Recovery Appliance for Backup and Redo Failover

To configure the primary Recovery Appliance for Backup and Redo Failover, you perform many of the tasks for setting up a downstream Recovery Appliance in a replication scenario.

Task 1: Create a VPC user account and a replication user account on the primary Recovery Appliance

Follow the instructions in "Creating Virtual Private Catalog Accounts".

For example, log in to the Recovery appliance as root, change to the bin directory, and use the following command to create the VPC user:

```
# ./racli add vpc_user --user_name=vpcuser
```

Enter the password for `vpcuser` user when prompted.

To create the replication user `repuser_from_alternate` with the `CREATE SESSION` privilege:

```
CREATE USER repuser_from_alternate IDENTIFIED BY password;
GRANT CREATE SESSION TO repuser_from_alternate;
```

The `user_name` created on the alternate must be the same as the VPC user created on the primary. However, the passwords do not need to be the same.

Task 2 Create a protection policy on the primary Recovery Appliance

Follow the instructions in "Creating a Protection Policy Using DBMS_RA". Ensure that the `store_and_forward` field is set to `NO`.

For example, execute the following PL/SQL program to create a `primary_brf` policy:

```
BEGIN
  DBMS_RA.CREATE_PROTECTION_POLICY (
    protection_policy_name => 'primary_brf',
    description            => 'For protected dbs on primary',
    storage_location_name  => 'delta',
    recovery_window_goal   => INTERVAL '28' DAY,
    guaranteed_copy        => 'NO',
    store_and_forward      => 'NO');
END;
```

Task 3: Add a database to the protection policy on the primary Recovery Appliance

Follow the instructions in "Adding Protected Database Metadata Using DBMS_RA".

For example, execute the following PL/SQL program to add `orcl12` to the `primary_brf` policy that you created in the previous task:

```
BEGIN
  DBMS_RA.ADD_DB (
    db_unique_name         => 'orcl12',
    protection_policy_name => 'primary_brf',
    reserved_space         => '128G');
END;
```

Task 4: Grant database access to the VPC user and the replication user on the primary Recovery Appliance

Follow the instructions in "Granting Database Access to a Recovery Appliance Account Using DBMS_RA".

For example, execute the following PL/SQL programs to grant the VPC user `vpcuser` and the replication user `repuser_from_alternate` the required privileges on protected database `orcl12`:

```
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'vpcuser',
    db_unique_name => 'orcl12');
END;
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'repuser_from_alternate',
```

```
    db_unique_name => 'orcl12');
END;
```

Configuring the Alternate Recovery Appliance for Backup and Redo Failover

To configure the alternate Recovery Appliance, you perform tasks similar to setting up an upstream Recovery Appliance in a replication scenario.

Task 1 Create a protection policy for Backup and Redo Failover on the alternate Recovery Appliance

Follow the instructions in "Creating a Protection Policy Using DBMS_RA". Ensure that you set the `store_and_forward` field to `YES`.

For example, execute the following PL/SQL program to create an `alt_brf` policy:

```
BEGIN
  DBMS_RA.CREATE_PROTECTION_POLICY (
    protection_policy_name => 'alt_brf',
    description            => 'For protected dbs on alternate',
    storage_location_name  => 'delta',
    recovery_window_goal   => INTERVAL '28' DAY,
    guaranteed_copy        => 'NO',
    store_and_forward      => 'YES');
END;
```

Task 2: Add the database to the protection policy on the alternate Recovery Appliance

Follow the instructions in "Adding Protected Database Metadata Using DBMS_RA".

For example, execute the following PL/SQL program to add `orcl12` to the `alt_brf` policy that you created in the previous task:

```
BEGIN
  DBMS_RA.ADD_DB (
    db_unique_name         => 'orcl12',
    protection_policy_name => 'alt_brf',
    reserved_space         => '128G');
END;
```

Task 3: Grant database access to the VPC user on the alternate Recovery Appliance

You created this user in "Task 1: Create a VPC user account and a replication user account on the primary Recovery Appliance".

Follow the instructions in "Granting Database Access to a Recovery Appliance Account Using DBMS_RA".

For example, execute the following PL/SQL program to grant the VPC user `vpcuser` the required privileges on protected database `orcl12` :

```
BEGIN
  DBMS_RA.GRANT_DB_ACCESS (
    username       => 'vpcuser',
    db_unique_name => 'orcl12');
END;
```

Configuring Replication for Backup and Redo Failover

After you configure the primary and the alternate Recovery Appliances, you perform tasks similar to setting up replication from the alternate to the primary appliance. In this scenario, the alternate appliance has the upstream role and the primary appliance has the downstream role.

Task 1: Configure an Oracle wallet on the alternate Recovery Appliance

On the alternate Recovery Appliance, use the `mkstore` utility to create an Oracle auto-login wallet and add the credentials for the replication user you created in "Task 1: Create a VPC user account and a replication user account on the primary Recovery Appliance". The alternate Recovery Appliance requires these credentials when it logs in to the primary Recovery Appliance.

To configure an auto-login wallet on the alternate Recovery Appliance:

1. Run the following command to create an Oracle wallet in the `/dbfs_repdbfs/REPLICATION` directory:

   ```
   mkstore -wrl /dbfs_repdbfs/REPLICATION -createALO
   ```

   The `mkstore` utility creates a file named `cwallet.sso` in the designated location.

2. Run the following command to add the replication user credentials:

   ```
   mkstore -wrl wallet_location -createCredential serv_name rep_user pwd
   ```

   The placeholders are defined as follows:

   - *`wallet_location`* is the directory in which you created the wallet in the previous step.
   - *`serv_name`* is the Oracle network service name that you use in an EZ Connect descriptor to identify the primary Recovery Appliance on the Oracle network.
   - *`rep_user`* is the user name of the replication user account. This user was created in Task 1: Create a VPC user account and a replication user account on the primary Recovery Appliance . The replication user is not created on the alternate.
   - *`pwd`* is the secure password of the replication user *`rep_user`*.

   For example, the following command adds credentials for the net service name `rapribrf-scan.acme.com` using port `1522` and a database name of `rapri`, and the replication user name `repuser_from_alternate`:

   ```
   mkstore -wrl /dbfs_repdbfs/REPLICATION -createCredential \
   "rapribrf-scan.acme.com:1522/rapri" "repuser_from_alternate" "pwd"
   ```

3. Verify that the credentials were properly added for this user by running the following command:

   ```
   mkstore -wrl /dbfs_repdbfs/REPLICATION -listCredential

   Oracle Secret Store Tool : Version 12.1.0.1 Copyright (c) 2004, 2012, Oracle
   and/or its affiliates. All rights reserved.
   List credential (index: connect_string username)
   1: rapribrf-scan.acme.com:1522/rapri repuser_from_alternate
   ```

   The results do not display the password.

Task 2: Create the replication server configuration on the alternate Recovery Appliance

For the primary Recovery Appliance to which this alternate Recovery Appliance will forward backups after an outage, create a replication server configuration by executing `DBMS_RA.CREATE_REPLICATION_SERVER`.

This task assumes the following:

- You want to create a replication server configuration named `raprimary_rep`.
- You want the alternate Recovery Appliance to log in to the primary Recovery Appliance using the replication account `repuser_from_alternate`. You created this account in "Task 1: Create a VPC user account and a replication user account on the primary Recovery Appliance".
- The configuration uses the net service name `rapribrf-scan.acme.com:1522/rapri` that you stored in the Oracle wallet you created in "Task 1: Configure an Oracle wallet on the alternate Recovery Appliance".
- The Oracle wallet is stored in `/dbfs_repdbfs/REPLICATION`.
- The file name of the Recovery Appliance Backup Module, which is preinstalled on every Recovery Appliance, is `/u01/app/oracle/product/12.1.0.2/dbh1/lib/libra.so`. The module functions as an SBT media management library. RMAN references this module when allocating or configuring a channel for backup to the Recovery Appliance (see "Configuring a Protected Database for Recovery Appliance Replication").

To create the replication server configuration:

1. With SQL*Plus or SQL Developer, connect to the alternate Recovery Appliance metadata database as `RASYS`.
2. Run the `DBMS_RA.CREATE_REPLICATION_SERVER` procedure for the primary Recovery Appliance.

   The following example creates the replication server configuration named `raprimary_rep` for the primary Recovery Appliance:

   ```
   BEGIN
     DBMS_RA.CREATE_REPLICATION_SERVER (
       replication_server_name => 'raprimary_rep',
       sbt_so_name      => '/u01/app/oracle/product/12.1.0.2/dbh1/lib/libra.so',
       catalog_user_name       => 'RASYS',
       wallet_alias            => 'rapribrf-scan.acme.com:1522/rapri',
       wallet_path             => 'file:/dbfs_repdbfs/REPLICATION');
   END;
   ```

3. Confirm the creation of the replication server configuration. The `replication_server_name` is converted to upper-case and stored as such. Therefore queries with the name should also be upper-case.

   For example, run the following query:

   ```
   SELECT COUNT(*) should_be_one
   FROM   RA_REPLICATION_SERVER
   WHERE  REPLICATION_SERVER_NAME = 'RAPRIMARY_REP';

   SHOULD_BE_ONE
   -------------
   1
   ```

If the configuration was created correctly, then the return value is `1`.

Task 3: Associate the alternate Recovery Appliance with the protection policy for Backup and Redo Failover

Specify the primary Recovery Appliance to which the alternate Recovery Appliance forwards backups after an outage by assigning the replication server configuration to a protection policy.

This task assumes the following:

- You want to use the replication server configuration named `raprimary_rep`, which you created in "Task 2: Create the replication server configuration on the alternate Recovery Appliance".
- You want to add the replication server configuration to protection policy `alt_brf`, which you created in "Task 1 Create a protection policy for Backup and Redo Failover on the alternate Recovery Appliance".

To associate the replication server configuration with the Backup and Redo Failover protection policy:

1. Ensure you are connected to the metadata database on the alternate Recovery Appliance as the Recovery Appliance administrator.
2. Run the `DBMS_RA.ADD_REPLICATION_SERVER` procedure for the Backup and Redo Failover protection policy and replication server configuration.

   For example, execute the following PL/SQL program:

```
BEGIN
  DBMS_RA.ADD_REPLICATION_SERVER (
   replication_server_name => 'raprimary_rep',
   protection_policy_name  => 'alt_brf');
END;
```

> **See Also:**
>
> "ADD_REPLICATION_SERVER"

Configuring the Protected Database for Backup and Redo Failover

After you configure replication for Backup and Redo Failover, the protected database administrator should perform the tasks in this section so that the protected database can send backups to the primary Recovery Appliance under normal conditions, and to the alternate Recovery Appliance during a planned or unplanned outage.

Task 1: Configure sqlnet.ora to point to the wallet location

Ensure that the sqlnet.ora file contains the location of the Oracle wallet.

The following example shows how the wallet location entry should appear:

```
SQLNET.WALLET_OVERRIDE = true
WALLET_LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
```

```
    (DIRECTORY = /u01/app/oracle/product/12.1.0/dbhome_1/dbs/zdlra)
  )
 )
```

Task 2: Create an auto-login wallet in the location specified in sqlnet.ora

The following example creates an auto-login wallet in the directory specified in "Task 1: Configure sqlnet.ora to point to the wallet location":

```
$ mkstore -wrl /u01/app/oracle/product/12.1.0/dbhome_1/dbs/zdlra/ -createALO
```

Task 3: Add the credentials for the primary and alternate Recovery Appliances to the wallet

In this task the protected database administrator adds credentials for the primary and alternate appliances using the VPC user you created in "Task 1: Create a VPC user account and a replication user account on the primary Recovery Appliance" to the wallet.

The following examples add the `vpcuser` credentials for the primary appliance `rapribrf-scan.acme.com:1521/rapri:dedicated` and the alternate appliance `raaltbrf-scan.acme.com:1521/raalt:dedicated` to the wallet on the protected database:

```
$ mkstore -wrl /u01/app/oracle/product/12.1.0/dbhome_1/dbs/zdlra/ -createCredential
"rapribrf-scan.acme.com:1521/rapri:dedicated" "vpcuser" "pwd"
$ mkstore -wrl /u01/app/oracle/product/12.1.0/dbhome_1/dbs/zdlra/ -createCredential
"raaltbrf-scan.acme.com:1521/raalt:dedicated" "vpcuser" "pwd"
```

Task 4: Register the database with the alternate Recovery Appliance and back up the control file

For this task, the protected database administrator performs steps 1 and 2, and the Recovery Appliance administrator performs step 3.

To register the database and back up the control file:

1. Using RMAN, connect to the protected database as `TARGET` and to the alternate Recovery Appliance catalog as `CATALOG`, and then run the `REGISTER DATABASE` command.
2. After the `REGISTER DATABASE` command is completed, back up the current control file to the alternate appliance:

   ```
   BACKUP DEVICE TYPE SBT CURRENT CONTROLFILE;
   ```

3. Verify that the control file backup was replicated from the alternate appliance to the primary appliance.

Task 5: Ensure that the database is registered with the primary Recovery Appliance

This step is to confirm that the protected database is registered with the primary appliance. Because replication is configured when the database is registered with the alternate appliance in the previous task, the database should automatically be registered with the primary appliance.

To confirm registration with the primary appliance:

1. In RMAN, connect to the database using the primary appliance credentials in the CATALOG connect string.

```
rman TARGET / CATALOG /@rapribrf-scan.acme.com:1521/rapri:dedicated
```

2. Run the `REGISTER DATABASE` command.

   The following error should display:

   ```
   RMAN-20002: target database already registered in recovery catalog
   ```

> **Note:**
>
> - The protected database administrator must also create a separate `RMAN` backup script that directs backups to the alternate Recovery Appliance when the primary appliance is not available, and redirects backups to the primary appliance when it is back in service. This script must connect to the alternate Recovery Appliance catalog and have the `CONFIGURE CHANNEL` or `ALLOCATE CHANNEL` command with `credential_alias` set to the alternate appliance. See *Zero Data Loss Recovery Appliance Protected Database Configuration Guide* for an example of how to create an RMAN backup script for the Recovery Appliance.
> - To send real time redo to the alternate Recovery Appliance during the outage of the primary appliance, an additional log archive destination must be defined as an `ALTERNATE` for the log archive destination used to connect to the primary appliance. The connect string must be defined in the Oracle auto-login wallet, similar to the connect string required for the primary appliance, and using the same VPC user (although the password may be different). See *Data Guard Concepts and Administration* for an example of how to use the `ALTERNATE` attribute to automatically fail over to a alternate remote destination.

Implementing DR Failover to Downstream Recovery Appliance

As part of disaster recovery, protected databases should failover to a downstream Recovery Appliance as the target for sending backup files and redo transport if the upstream Recovery Appliance is unavailable. This section provides steps on how to configure a protected database for transparent failover of backup operations and redo transport to a downstream Recovery Appliance.

For sake of clarity, this examples makes the following assumptions:

- If you have real time redo transport enabled, it receives an error and stops sending redo to the upstream Recovery Appliance. Within a minute, real time redo transport connects to the downstream Recovery Appliance and resumes sending redo there.
- The name of the example protected database is `CDB122DR`. It is a Container Databases with One Pluggable Database.
- The name of the example upstream Recovery Appliance is `RAHADR1`.
- The name of the example downstream Recovery Appliance is `RAHADR2`.

- A common VPC user called HADR_COMMON_VPCUSER was created on both Recovery Appliances and ***must*** use the same password on both.
- A local VPC user called HADR_LOCAL_VPCUSER has been created on both Recovery Appliances but the password can be different between the two.
- The replication server between RAHADR1 and RAHADR2 is using the VPC user REPUSER_FROM_HADR1.

Setup and Configuration for Failover

This section establishes VPC users for the Recovery Appliances to use later for failover.It modifies the network configuration files needed, configures the replication server, creates protection policies, registers the protected database, and adds several grants to the upstream and downstream Recovery Appliances.

Creating VPC Users

This task creates database VPC user accounts in the upstream and downstream Recovery Appliances.

When creating the accounts, keep in mind these password requirements.

- The first VPC user (HADR_LOCAL_VPCUSER) account may be used by other protected databases and can have different passwords between the RAHADR1 and RAHADR2 Recovery Appliances.
- The second VPC user (HADR_COMMON_VPCUSER) account must use the same password on both the RAHADR1 and RAHADR2 Recovery Appliances and can be used by other protected databases

The following conditions are applicable to this specific example.

- Recovery Appliance RAHADR1 has previously been installed with a DB_UNIQUE_NAME of rahadr1.
- Recovery Appliance RAHADR2 has previously been installed with a DB_UNIQUE_NAME of rahadr2.

1. Create two VPC users for the protected database on the upstream Recovery Appliance RAHADR1.

```
# racli add vpc_user --user_name HADR_LOCAL_VPCUSER
[HADR_LOCAL_VPCUSER] New Password: ***********
Sun Mar 25 08:27:53 2018: Start: Add vpc user HADR_LOCAL_VPCUSER.
Sun Mar 25 08:27:53 2018: Add vpc user HADR_LOCAL_VPCUSER successfully.
Sun Mar 25 08:27:53 2018: End: Add vpc user HADR_LOCAL_VPCUSER.

# racli add vpc_user --user_name HADR_COMMON_VPCUSER
[HADR_COMMON_VPCUSER] New Password: ***********
Sun Mar 25 08:27:53 2018: Start: Add vpc user HADR_COMMON_VPCUSER.
Sun Mar 25 08:27:53 2018: Add vpc user HADR_COMMON_VPCUSER successfully.
Sun Mar 25 08:27:53 2018: End: Add vpc user HADR_COMMON_VPCUSER.
```

2. Create two VPC users for the protected database on the downstream Recovery Appliance RAHADR2.

```
# racli add vpc_user --user_name HADR_LOCAL_VPCUSER
[HADR_LOCAL_VPCUSER] New Password: ***********
Sun Mar 25 08:27:53 2018: Start: Add vpc user HADR_LOCAL_VPCUSER.
Sun Mar 25 08:27:53 2018: Add vpc user HADR_LOCAL_VPCUSER successfully.
Sun Mar 25 08:27:53 2018: End: Add vpc user HADR_LOCAL_VPCUSER.

# racli add vpc_user --user_name HADR_COMMON_VPCUSER
[HADR_COMMON_VPCUSER] New Password: ***********
Sun Mar 25 08:27:53 2018: Start: Add vpc user HADR_COMMON_VPCUSER.
Sun Mar 25 08:27:53 2018: Add vpc user HADR_COMMON_VPCUSER successfully.
Sun Mar 25 08:27:53 2018: End: Add vpc user HADR_COMMON_VPCUSER.
```

3. If the VPC user account used by the replication server for sending backups from the upstream (HARADR1) to the downstream (RAHADR2) Recovery Appliances hasn't been created, create the VPC user now.

```
# racli add vpc_user --user_name REPUSER_FROM_HADR1
[REPUSER_FROM_HADR1] New Password: ***********

Sun Mar 25 08:35:01 2018: Start: Add vpc user REPUSER_FROM_HADR1.
Sun Mar 25 08:35:01 2018: Add vpc user REPUSER_FROM_HADR1 successfully.
Sun Mar 25 08:35:01 2018: End: Add vpc user REPUSER_FROM_HADR1.
```

Modifying Configuration for Transport Failover

This task modifies the Oracle network configuration files that are used for transparent failover to the downstream Recovery Appliance.

If you have a RAC database, this should be performed on each host where the protected database runs.

1. Verify that there are no `${ORACLE_HOME}/dbs/ra${ORACLE_SID}.ora` files on any of the hosts.

   This file has the effect of overriding all the configuration parameters defined in this step and should be removed if present.

2. Configure a TNS alias in the `tnsnames.ora` file that will be used by RMAN to connect to the correct Recovery Appliance.

```
$ cd ${ORACLE_HOME}/network/admin
```

3. Edit `tnsnames.ora` and add the following entry:

```
DR_RAHADR =
(DESCRIPTION_LIST =
  (LOAD_BALANCE = off)
  (FAILOVER = on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5)
    (TRANSPORT_CONNECT_TIMEOUT = 3)
    (RETRY_COUNT = 3)
```

```
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ra1ingest-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rahadr1)
    )
  )
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5)
    (TRANSPORT_CONNECT_TIMEOUT = 3)
    (RETRY_COUNT = 3)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ra2ingest-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rahadr2)
    )
  )
)
DR_RAHADR1 =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5)
    (TRANSPORT_CONNECT_TIMEOUT = 3)
    (RETRY_COUNT = 3)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ra1ingest-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rahadr1)
    )
  )
)
DR_RAHADR2 =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5)
    (TRANSPORT_CONNECT_TIMEOUT = 3)
    (RETRY_COUNT = 3)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ra2ingest-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rahadr2)
    )
  )
)
```

Configuring the Replication Server

This task configures the replication server that sends the database backups from RAHADR1 to the RAHADR2 Recovery Appliance.

The operations and naming conventions used here are equivalent to those in Enterprise Manager when there is no dedicated replication network. For more information, see Replicating Backups with Recovery Appliance .

The replication server between RAHADR1 and RAHADR2 has not already been created.

1. If a replication wallet does not exist on RAHADR1, create a replication wallet that points to RAHADR2.

```
$ mkstore -wrl file:/dbfs_repdbfs/REPLICATION -createALO
```

2. Add the credentials to the wallet. On RAHADR1, add the credentials for logging into RAHADR2.

```
$ mkstore -wrl file:/dbfs_repdbfs/REPLICATION
-createCredential <rahadr2-scan>:1521/rahadr2 REPUSER_FROM_HADR1
my_v3ry_c0mplex_pa55w0rd
```

3. Create the Recovery Appliance replication server on RAHADR1.

```
$ sqlplus rasys/ra

SQL> exec dbms_ra.create_replication_server(
replication_server_name =>'RAHADR2_REP',
sbt_so_name => 'libra.so', max_streams => 8,
catalog_user_name=> 'RASYS',
wallet_alias => '<rahadr2-scan>:1521/rahadr2',
wallet_path => 'file:/dbfs_repdbfs/REPLICATION');

PL/SQL procedure successfully completed.
```

Configuring Upstream and Downstream Recovery Appliances

This task configures the protection policies for the protected database on the downstream and upstream Recovery Appliance, and then adds the protection policy to the replication server.

If a protection policy that is used, for example, by the CBR122DR database does not exist on the respective Recovery Appliances, these steps create them. The protection policy name does not have to be unique between the downstream and upstream Recovery Appliances.

To prevent a circular references between RAHADR1 and RAHADR2, the protection policy from RAHADR2 is not added to the replication server while the protection policy from RAHADR1 is added. All databases in the protection policy are replicated.

Note: Because RAHADR2 does not normally accept redo from the CDB122DR database, set the unprotected data window parameter is set to 1.25 days to avoid false alerts from occurring if the CDB122DR database is idle.

1. Log into SQLPLS as `rasys/ra` on RAHADR2, the downstream Recovery Appliance. This step and the next few are performed on RAHADR2, unless otherwise stated.

```
$ sqlplus rasys/ra
```

2. Create the protection policy.

```
SQL> exec dbms_ra.create_protection_policy(
protection_policy_name => 'cdb122dr_PP',
storage_location_name => 'DELTA',
recovery_window_goal => numtodsinterval(3,'DAY'),
unprotected_window => numtodsinterval(1.25,'DAY'),
allow_backup_deletion => 'NO');

PL/SQL procedure successfully completed.
```

3. Add the database (for this example) and its protection policy to the list of those protected by the Recovery Appliance.

```
SQL> exec dbms_ra.add_db(
db_unique_name => 'cdb122dr',
protection_policy_name=> 'cdb122dr_PP',
reserved_space => '1T');

PL/SQL procedure successfully completed.
```

4. Grant access to the replication user to the database (for this example).

```
SQL> exec dbms_ra.grant_db_access(
username => 'REPUSER_FROM_HADR1',
db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.
```

5. Log into sqlplus as `rasys/ra` on RAHADR1, the upstream Recovery Appliance. This step and all that follow are performed on RAHADR1.

```
$ sqlplus rasys/ra
```

6. Create a protection policy. The protection policy name does not have to be unique.

```
SQL> exec dbms_ra.create_protection_policy(
protection_policy_name =>'cdb122dr_PP',
storage_location_name => 'DELTA',
recovery_window_goal => numtodsinterval(3,'DAY'),
unprotected_window => numtodsinterval(5,'MINUTE'),
allow_backup_deletion => 'NO');

PL/SQL procedure successfully completed.
```

7. Add the database (for this example) and its protection policy to the list of those protected by the Recovery Appliance.

```
SQL> exec dbms_ra.add_db(
db_unique_name => 'cdb122dr',
protection_policy_name=> 'cdb122dr_PP',
reserved_space => '1T');

PL/SQL procedure successfully completed.
```

8. Grant access to the replication user to the database (for this example).

```
SQL> exec dbms_ra.grant_db_access(
username => 'HADR_LOCAL_VPCUSER',
db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.
```

9. Add the protection policy to the replication server. This step is performed on the upstream Recovery Appliance (RAHADR1). This step was not performed on the downstream Recovery Appliance, in order to prevent a circular reference between the two Recovery Appliances.

```
SQL> exec dbms_ra.add_replication_server(
replication_server_name =>'RAHADR2_REP',
protection_policy_name => 'cdb122dr_PP');

PL/SQL procedure successfully completed.
```

Registering the Protected Database on the Upstream Recovery Appliance

This task configures the wallet, adds VPC user credentials, tests those credentials, and registers the protected database with the upstream Recovery Appliance. If it is a RAC database, the steps need to be performed on each host where the protected database runs.

The operations and naming conventions used here are equivalent to those in Enterprise Manager.

1. Configure the `sqlnet.ora` file that will be used by RMAN to connect to the correctRecovery Appliance. Go to the proper directory.

```
$ cd ${ORACLE_HOME}/network/admin
```

2. Edit the `sqlnet.ora` file and ensure the following parameters are set correctly:

```
SQLNET.WALLET_OVERRIDE = true

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
```

```
      (DIRECTORY = /u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/zdlra)
    )
  )

SQLNET.EXPIRE_TIME = 10
```

3. Create a replication wallet that stores each of the VPC user credentials. Perform this step only if the replication wallet doesn't already exist. On each host:

```
$ mkstore -wrl file:/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/zdlra
-
createALO
```

4. Create credential aliases for each of the three credentials that will be used by RMAN. On each host, run the `mkstore` command. Enter the appropriate password when prompted.

```
$ mkstore -wrl file:/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/zdlra
-
createCredential dr_rahadr2 hadr_local_vpcuser hadr2_L0cal_Pa55w0rd

$ mkstore -wrl file:/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/zdlra
-
createCredential dr_rahadr1 hadr_local_vpcuser hadr1_L0cal_Pa55w0rd

$ mkstore -wrl file:/u01/app/oracle/product/12.2.0.1/dbhome_1/dbs/zdlra
-
createCredential dr_rahadr hadr_common_vpcuser c0mm0n_Pa55w0rd
```

5. Verify the credentials are working correctly by logging into each target using only the credential alias. On each host, run the following:

```
$ sqlplus /@dr_rahadr1
```

6. Register the protected database with the Recovery Appliance in RAHADR1. On one of the hosts, run:

```
$ rman target / catalog /@dr_rahadr1

RMAN> register database;
```

7. Perform a test backup of the current control file to Recovery Appliance hadr1 (RAHADR1). On one of the protected database hosts, perform a backup of the current control file.

```
$ rman target / catalog /@dr_rahadr1

RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' FORMAT '%d_%U' PARMS
"SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so,
ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/z
dlra credential_alias=dr_rahadr1')";

RMAN> backup device type sbt current controlfile tag 'controltest';
```

```
Starting backup at 05-JUN-18
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=2320 instance=cdb122dr1 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: RA Library (RAHADR1)
SID=6DE9FE3D49ED4598E05311F3850AC59F
allocated channel: ORA_SBT_TAPE_2
channel ORA_SBT_TAPE_2: SID=2516 instance=cdb122dr1 device type=SBT_TAPE
channel ORA_SBT_TAPE_2: RA Library (RAHADR1)
SID=6DE9FE48D84C48C8E05311F3850A89BE
channel ORA_SBT_TAPE_1: starting full datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
including current control file in backup set
channel ORA_SBT_TAPE_1: starting piece 1 at 05-JUN-18
channel ORA_SBT_TAPE_1: finished piece 1 at 05-JUN-18
piece handle=CDB122DR_2kt4m80u_1_1 tag=CONTROLTEST comment=API Version
2.0,MMS Version 3.17.1.26
channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:15
Finished backup at 05-JUN-18
Starting Control File and SPFILE Autobackup at 05-JUN-18
piece handle=c-3244939197-20180605-00 comment=API Version 2.0,MMS
Version
3.17.1.26
Finished Control File and SPFILE Autobackup at 05-JUN-18
```

8. List the backup set just created. Verify there are two copies of the control file, one on Recovery Appliance hadr1 (RAHADR1) and the other on Recovery Appliance hadr2 (RAHADR2).

```
RMAN> list backupset tag CONTROLTEST;

List of Backup Sets
===================
BS Key Type LV Size
------- ---- -- ----------
220 Full 138.75M
Control File Included: Ckp SCN: 9076177 Ckp time: 05-JUN-18
Backup Set Copy #1 of backup set 220
Device Type Elapsed Time Completion Time Compressed Tag
----------- ------------ --------------- ---------- ---
SBT_TAPE 07:00:21 05-JUN-18 NO CONTROLTEST
List of Backup Pieces for backup set 220 Copy #1
BP Key Pc# Status Media Piece Name
------- --- ----------- ---------------------- ----------
221 1 AVAILABLE Recovery Appliance (RAHADR1)
CDB122DR_2kt4m80u_1_1
Backup Set Copy #2 of backup set 220
Device Type Elapsed Time Completion Time Compressed Tag
----------- ------------ --------------- ---------- ---
SBT_TAPE 07:00:21 05-JUN-18 NO CONTROLTEST
List of Backup Pieces for backup set 220 Copy #2
BP Key Pc# Status Media Piece Name
------- --- ----------- ---------------------- ----------
```

```
246 1 AVAILABLE Recovery Appliance (RAHADR2)
RA_SBT_CDB122DR_3244939197_230_2kt4m80u_1_2_220
```

Adding Remaining Grants to the Upstream and Downstream Recovery Appliance

This task grants access to VPC users on both the upstream and downstream Recovery Appliances.

1. On RAHADR1, add the grant access to the one remaining VPC users.

```
SQL> exec dbms_ra.grant_db_access(
username => 'HADR_COMMON_VPCUSER',
db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.
```

2. On RAHADR2, add the grant access to the two remaining VPC users. These users are pre-setup in the event that backups failover, due to RAHADR1 not being available.

```
SQL> exec dbms_ra.grant_db_access(
username => 'HADR_LOCAL_VPCUSER',
db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.

SQL> exec dbms_ra.grant_db_access(
username => 'HADR_COMMON_VPCUSER',
db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.
```

3. Verify the credentials are working correctly by logging into each target using only the credential alias. On each host run:

```
$ sqlplus /@dr_rahadr2

$ sqlplus /@dr_rahadr
```

Configuring Channel Device Parameters

This task configures the channel device parameters for use with the `DR_RAHADR` alias.

1. On one of the protected database hosts, run:

```
$ rman target / catalog /@dr_rahadr1

RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' FORMAT '%d_%U' PARMS
"SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so,
ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
```

```
dbhome_1/dbs/z
dlra credential_alias=dr_rahadr')";
```

2. (Optional) configure the following parameters, which are best practice recommendations.

```
RMAN> CONFIGURE BACKUP OPTIMIZATION on;

RMAN> CONFIGURE CONTROLFILE AUTOBACKUP on;

RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;

RMAN> CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 2 BACKUP TYPE TO
BACKUPSET;

RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+RECOC1/cdb122dr/
snapcf.f';

RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO backed up 1 times to
device
type sbt;
```

Configuring Upstream and Downstream Recovery Appliance

This task creates host specific files for backups, loads the scripts on their respective hosts, and verifies the credentials.

1. On a host of the upstream Recovery Appliance, create the `backup_database_rahadr1.rman` text file with the following content.

```
{
allocate channel rahadr1_sbt_1 device type sbt
  format '%d_%U'
  PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/
libra.so,
  ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/zdlra
  credential_alias=dr_rahadr1')";

allocate channel rahadr1_sbt_2 device type sbt
  format '%d_%U'
  PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/
libra.so,
  ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/zdlra
  credential_alias=dr_rahadr1')";

backup
  tag '&1'
  cumulative incremental level 1
  filesperset 1
  section size 64g
  database
    plus archivelog
```

```
      not backed up
      filesperset 32
      delete input;
}
```

2. On a host of the downstream Recovery Appliance, create the `backup_database_rahadr2.rman` text file with the following content.

```
{
allocate channel rahadr2_sbt_1 device type sbt
  format '%d_%U'
  PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/
libra.so,
  ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/zdlra
  credential_alias=dr_rahadr2')";

allocate channel rahadr1_sbt_2 device type sbt
  format '%d_%U'
  PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/
libra.so,
  ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/zdlra
  credential_alias=dr_rahadr2')";

backup
  tag '&1'
  cumulative incremental level 1
  filesperset 1
  section size 64g
  database
    plus archivelog
      not backed up
      filesperset 32
      delete input;
}
```

3. Ensure the script on RAHADR1 does not exist by trying to delete it first. Then load the HADR1 script into the RAHADR1 Recovery Appliance.

```
$ rman target / catalog /@dr_rahadr1

RMAN> delete script backup_database;

RMAN> create script backup_database from file
'/home/oracle/backup_database_rahadr1.rman';
```

4. Ensure the script on RAHADR2 does not exist by trying to delete it first. Then load the HADR2 script into the RAHADR2 Recovery Appliance.

```
$ rman target / catalog /@dr_rahadr2

RMAN> delete script backup_database;
```

```
RMAN> create script backup_database from file
'/home/oracle/backup_database_rahadr2.rman';
```

5. Verify credentials have access to the database.

```
$ rman target / catalog /@dr_rahadr

RMAN> print script backup_database;

printing stored script: backup_database
{
allocate channel rahadr1_sbt_1 device type sbt
  format '%d_%U'
  PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/
libra.so,
  ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/zdlra
  credential_alias=dr_rahadr1')";

allocate channel rahadr1_sbt_2 device type sbt
  format '%d_%U'
  PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/
libra.so,
  ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/
dbhome_1/dbs/zdlra
  credential_alias=dr_rahadr1')";

backup
  tag '&1'
  cumulative incremental level 1
  filesperset 1
  section size 64g
  database
    plus archivelog
      not backed up
      filesperset 32
      delete input;
}
```

Backup Operation

This task starts the `backup_database` script.

The following RMAN command should be used for all RMAN backup operations.

> **Note:**
>
> When the script is run, the channel allocations indicate which Recovery Appliance is logged into and the Recovery Appliance database name.

1. Start RMAN

```
$ rman target / catalog /@dr_rahadr
```

2. Start the backup_database script. If RAHADR1 is running, the script logs into RAHADR1. Otherwise, the script logs into RAHADR2.

```
RMAN> run { execute script backup_database using 'Level1'; }

executing script: backup_database

allocated channel: rahadr1_sbt_1
channel rahadr1_sbt_1: SID=1936 instance=cdb122dr1 device type=SBT_TAPE
channel rahadr1_sbt_1: RA Library (RAHADR1)
SID=6DEA2A958DFBE0CFE05311F3850AB3AB

allocated channel: rahadr1_sbt_2
channel rahadr1_sbt_2: SID=394 instance=cdb122dr1 device type=SBT_TAPE
channel rahadr1_sbt_2: RA Library (RAHADR1)
SID=6DEA2A9CC2BBE0D0E05311F3850AC634
```

Backup Piece Gap Resolution

When the upstream Recovery Appliance (RAHADR1) becomes available again, the backups that had failed over to the downstream Recovery Appliance (RAHADR2) need to be transferred back to RAHADR1 to resolve the gap of virtual full backups.

The gap shows as `INDEX_BACKUP` tasks in `ORDERING_WAIT` state on RAHADR1, because the virtual full backup metadata is present through normal catalog reconcile with RAHADR2 configured as downstream, however the backups are not yet physically present on the upstream Recovery Appliance.

The PL/SQL script `tkrmrshadr.sql` performs this operation. It loads the `RA_POPULATE_BACKUP_PIECE` procedure into the database. Then the script creates a `DBMS_SCHEDULER_JOB` that runs every 15 minutes to look for `INDEX_BACKUP` tasks that are in an `ORDERING_WAIT` state. It determines which backup pieces need to be transferred to the upstream Recovery Appliance RAHADR1 from the downstream RAHADR2. The backup pieces are transferred in parallel if possible using the `DBMS_RA.POPULATE_BACKUP_PIECE` command.

The initial query is very quick. However, if pieces are found, then the job can run for an extended period of time due to the `INDEX_BACKUP` tasks that are created on RAHADR1 as a result of the `DBMS_RA.POPULATE_BACKUP_PIECE` calls.

The MD5SUM for the `tkrmrshadr.sql` is `beb79a1bdd61c91b34e0d777f75c2227`.

```
$ md5sum tkrmrshadr.sql beb79a1bdd61c91b34e0d777f75c2227 tkrmrshadr.sql

--------------********** -------------------
Installing tkrmrshadr.sql
- Install the script into all RAs participating in HADR.
- The script only needs to be installed once:
- As rasys: sqlplus rasys/<rasyspwd> @<full_dir_location_of_script>/
```

```
tkrmrshadr.sql
--------------**************** ------------------
```

Real-Time Redo Transport

Real-Time Redo Transport for protected databases can be configured to regularly use the upstream Recovery Appliance, but to failover to the downstream Recovery Appliance when the upstream one isn't available. When the upstream Recovery Appliance becomes available again, redo transport automatically changes from using the downstream back to using the upstream.

Configuring the VPC User for Real-Time Redo Transport

This task establishes the VPC user for redo transport and then you choose between (1) enabling parameters in Data Guard Broker and (2) enabling log archive parameters.

1. Configure the `redo_transport_user` to the local VPC user.

```
$ sqlplus / as sysdba

SQL> alter system set redo_transport_user=hadr_local_vpcuser;

System altered.
```

2. Choose one of the two options.
   - Option 1: Use Data Guard Broker to Configure Real-Time Redo Transport
   - Option 2: Use log_archive* Parameters to Configure Real-Time Redo Transport

Option 1: Use Data Guard Broker to Configure Real-Time Redo Transport

This task enables Data Guard Broker parameters that establish failover of real-time redo transport from the upstream to the downstream Recovery Appliance.

1. Enable the `dg_broker*` parameters from a SQLPLUS session as `sysdba`.

```
$ sqlplus / as sysdba

SQL> alter system set
dg_broker_config_file1='+DATAC1/cdb122dr/dr1cdb122dr.dat';
System altered.

SQL> alter system set
dg_broker_config_file2='+DATAC1/cdb122dr/dr2cdb122dr.dat';
System altered.

SQL> alter system set dg_broker_start=true;
System altered.
```

2. Configure Data Guard Broker with respect to the primary databases, connection identifiers for the Recovery Appliances, network timeouts, and maximum number of failures. In the end, enable the configuration changes.

```
$ dgmgrl sys/myPassword

DGMGRL for Linux: Release 12.2.0.1.0 - Production on Tue Jun 5 11:37:44
2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights
reserved.

Welcome to DGMGRL, type "help" for information.
Connected to "cdb122dr"
Connected as SYSDG.

DGMGRL> create configuration cdb122dr as primary database is cdb122dr
connect
identifier is '//scam06-scan3/cdb122dr';
Configuration "cdb122dr" created with primary database "cdb122dr"

DGMGRL> add recovery_appliance rahadr1 as connect identifier is
'dr_rahadr1';
Recovery Appliance "rahadr1" added

DGMGRL> add recovery_appliance rahadr2 as connect identifier is
'dr_rahadr2';
Recovery Appliance "rahadr2" added

DGMGRL> edit recovery_appliance rahadr1 set property MaxFailure=1;
Property "maxfailure" updated

DGMGRL> edit recovery_appliance rahadr1 set property ReopenSecs=10;
Property "reopensecs" updated

DGMGRL> edit recovery_appliance rahadr1 set property NetTimeout=8;
Property "nettimeout" updated

DGMGRL> edit recovery_appliance rahadr2 set property MaxFailure=1;
Property "maxfailure" updated

DGMGRL> edit recovery_appliance rahadr2 set property NetTimeout=8;
Property "nettimeout" updated

DGMGRL> edit database cdb122dr set property RedoRoutes = '(LOCAL :
(rahadr1
async priority=1, rahadr2 async priority=2))';
Warning: ORA-16677: Standby database has the same or higher priority
than
other members specified in the RedoRoutes group.
Property "redoroutes" updated

DGMGRL> enable configuration;
Enabled.
```

> **Note:**
>
> If Redo Transport does not start, then you may need to bounce the protected database. For a RAC database, this can be done in a rolling fashion.

Option 2: Use log_archive* Parameters to Configure Real-Time Redo Transport

This task enables manually changes several `log_archive*` parameters that establish failover of real-time redo transport from the upstream to the downstream Recovery Appliance.

- Log into `sqlplus` as `rasys/ra` and change several parameters with respect to the primary databases, connection identifiers for the Recovery Appliances, network timeouts, and maximum number of failures. In the end, enable the configuration changes.

```
$ sqlplus rasys/ra

SQL> alter system set log_archive_config =
'dg_config=(cdb122dr,rahadr1,rahadr2)';

SQL> alter system set log_archive_dest_2='service=dr_rahadr1 ASYNC
NOAFFIRM
delay=0 optional compression=disable max_failure=1 max_connections=1
reopen=10 db_unique_name=rahadr1 net_timeout=8 group=1 priority=1
valid_for=(online_logfile,all_roles)';

SQL> alter system set log_archive_dest_3='service=dr_rahadr2 ASYNC
NOAFFIRM
delay=0 optional compression=disable max_failure=1 max_connections=1
reopen=300 db_unique_name=rahadr2 net_timeout=8 group=1 priority=2
valid_for=(online_logfile,all_roles)';

SQL> alter system set log_archive_dest_state_2=enable;

SQL> alter system set log_archive_dest_state_3=enable;
```

> **Note:**
>
> If Redo Transport does not start, then you may need to bounce the protected database. For a RAC database, this can be done in a rolling fashion.

10

Monitoring the Recovery Appliance

This chapter explains how to perform basic monitoring of a Recovery Appliance, including configuring the metric and configuration settings.

About Monitoring the Recovery Appliance

This section contains the following topics:

- Purpose of Monitoring the Recovery Appliance
- Overview of Recovery Appliance Monitoring Capabilities
- Cloud Control Interface for Monitoring the Recovery Appliance
- Basic Tasks for Monitoring the Recovery Appliance

> **See Also:**
>
> "Protection Policies" for an architectural overview

Purpose of Monitoring the Recovery Appliance

A crucial part of ongoing Recovery Appliance administration is regularly monitoring the overall health of the Recovery Appliance, and checking the status of protected databases, backup and replication jobs, and storage usage.

Overview of Recovery Appliance Monitoring Capabilities

This section describes the monitoring tools supplied by Oracle.

Cloud Control

The primary monitoring tool for Recovery Appliance administrators is the Oracle Enterprise Manager Cloud Control (Cloud Control) incident and event notification framework. The primary interface is the Recovery Appliance home page, which prominently displays warnings, alerts, and errors. The monitoring framework integrated with Cloud Control is an effective way of managing issues and tracking them until resolution.

Space management is a crucial part of administering the Recovery Appliance. To have sufficient time to accommodate storage demands, you must know when estimated storage needs are approaching the amount of total storage available. Cloud Control provides warnings and error messages regarding aggregate storage usage, providing ample time to make necessary changes.

Cloud Control enables you to customize settings to meet your management goals. For example, you can receive warnings if the space needed to meet the recovery window goal of a specific database is a user-specified percentage of its reserved space. You can also configure email alerts so that you receive immediate notification of issues without having to log in to the system.

> **See Also:**
>
> Cloud Control online help

Oracle Configuration Manager

Oracle Configuration Manager collects configuration information (by default, every day) and uploads it to the Oracle Management Repository. If you log a service request, then the configuration data is associated with the service request. Oracle Support Services can analyze the data and provide better service.

Benefits of Oracle Configuration Manager include the following:

- Reduces time for resolution of support issues
- Provides pro-active problem avoidance
- Improves access to best practices and the Oracle knowledge base
- Improves understanding of customer's business needs and provides consistent responses and services

Oracle Configuration Manager software is installed in each Oracle home. Typically, each Oracle home has a collector configured that gathers and uploads information under its My Oracle Support (MOS) credentials. You can also configure a central collector, which gathers information for the Oracle home in which it resides *and* Oracle homes in which the collector is disconnected or not configured.

> **See Also:**
>
> - *Oracle Configuration Manager Installation and Administration Guide* for an introduction to Oracle Configuration Manager
> - *Oracle Configuration Manager Collection Overview*

Auto Service Request (ASR)

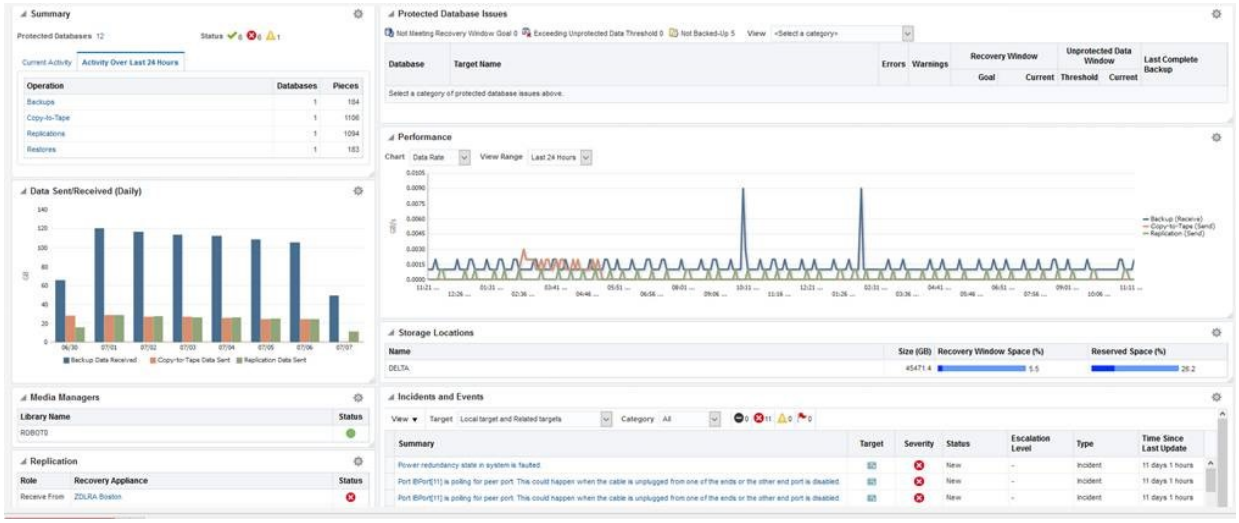
Auto Service Request (ASR) is a feature that automatically opens service requests when specific Recovery Appliance hardware faults occur. ASR detects faults in the most common server components, such as disks, fans, and power supplies. ASR monitors only server components and does not detect all possible faults.

ASR is not a replacement for other monitoring mechanisms, such as SMTP and SNMP alerts, within the customer data center. It is a complementary mechanism that expedites and simplifies the delivery of replacement hardware.

> **See Also:**
>
> *Zero Data Loss Recovery Appliance Owner's Guide* to learn how to set up ASR

Cloud Control Interface for Monitoring the Recovery Appliance

The primary interface for monitoring the Recovery Appliance is the Recovery Appliance Home page. The Home page lists any existing warnings and alerts, as shown in the following graphic:



The following sections of the Home page show monitoring information:

- Summary

  This section shows the number of databases with no issues, with alerts, and with warnings. In Cloud Control, an alert is an indicator that a particular metric condition has been encountered. For example, an alert might indicate that a metric threshold has been reached.

- Media Managers and Replication

  These sections show the status of copy-to-tape and Recovery Appliance replication services.

- Protected Database Issues

  This section summarizes the backup status for protected databases, and provides a category filter so you can view which databases are affected.

- Incidents and Events

  This section displays incidents and events reported for the Recovery Appliance and all associated targets. You can filter by target and category. You can click the Summary link to drill down to the Incident Manager to view detailed information about the incident.

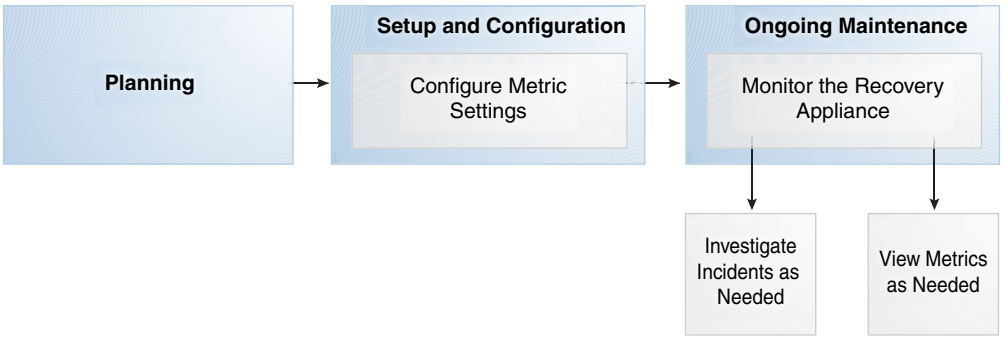
> **Note:**
>
> Warnings automatically clear when the underlying issue is resolved.

> **See Also:**
>
> - "Accessing the Recovery Appliance Home Page"
> - Cloud Control online help to learn more about the components of the Recovery Appliance Home page

Basic Tasks for Monitoring the Recovery Appliance

This section explains the basic tasks involved in monitoring the Recovery Appliance. The following diagram shows the overall workflow described in Recovery Appliance Workflow , with the monitoring tasks highlighted.

Figure 10-1 Monitoring Tasks in the Recovery Appliance Workflow



Typically, you perform monitoring tasks in the following sequence:

1. During the configuration phase (see "Setup and Configuration for Recovery Appliance"), configure your metric settings. For example, you may want to configure the Recovery Appliance to issue a warning if a threshold is passed.

   "Modifying the Metric and Collection Settings" describes this task.

2. During the ongoing maintenance phase (see "Maintenance Tasks for Recovery Appliance"), modify protection policies as needed. Typical modification tasks include:

   - Investigate incidents as needed.

     "Viewing the Incident Manager Page" describes this task.

   - View metrics as needed.

     "Modifying the Metric and Collection Settings" describes this task.

Modifying the Metric and Collection Settings

The Metric and Collection Settings page provides details about thresholds and schedules for target metric collection. Using this page, you can edit the warning threshold and critical threshold values of target metrics and other collected items, and the time intervals for collection. The page shows a pencil icon in the Edit column for modifiable settings.

Prerequisites

You must log in to the Recovery Appliance metadata database as `RASYS`.

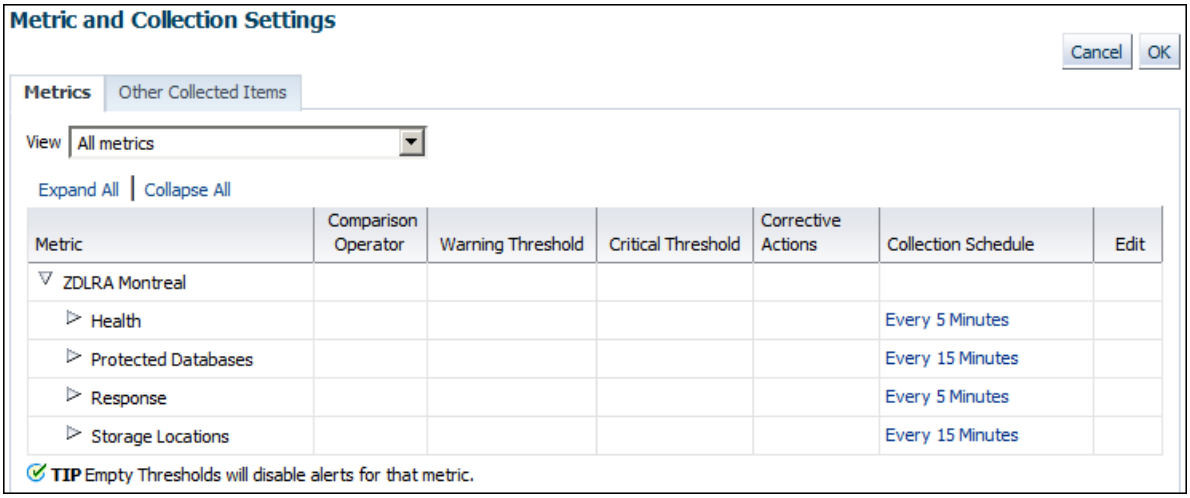
Assumptions

You want to receive warnings when the space needed to meet the recovery window goal for a database is 80% percent of its reserved space setting. You want the critical threshold to be 95%.

To modify the metric and collection settings:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. From the **Recovery Appliance** menu, click **Monitoring**, and then click **Metric and Collection** settings.

   The Metric and Collection Settings page appears.



3. If **All metrics** is not selected in the **View** menu, then select it.

   The page refreshes to show all the available metrics.
4. Expand **Protected Databases**.
5. Scroll down the page until you find the row that says `Recovery Window Space as a Percentage of Reserved Space`.
6. For this row, enter the following values, and then click **OK**:
   - In the Warning Threshold column, enter `80`.

- In the Critical Threshold column, enter 95.

A confirmation message appears.

> **Note:**
>
> To change the default text of the alert message that is generated when these thresholds are passed, click the pencil icon.

7. Modify other metric settings as needed.

> **See Also:**
>
> Cloud Control online help to learn more about metric and collection settings

Viewing the Incident Manager Page

The Incidents and Events section shows all incidents, events, and warnings for a Recovery Appliance. Click any incident to open the Incident Manager page. Incident Manager provides, in one location, the ability to search, view, manage, and resolve incidents and problems impacting your environment.

Prerequisites

You must log in to the metadata database as RASYS.

Assumptions

This tutorial assumes that Incidents and Events section of the Recovery Appliance Home page for your Recovery Appliance shows a warning. You want to get more details about it.

To view the Incident Manager page:

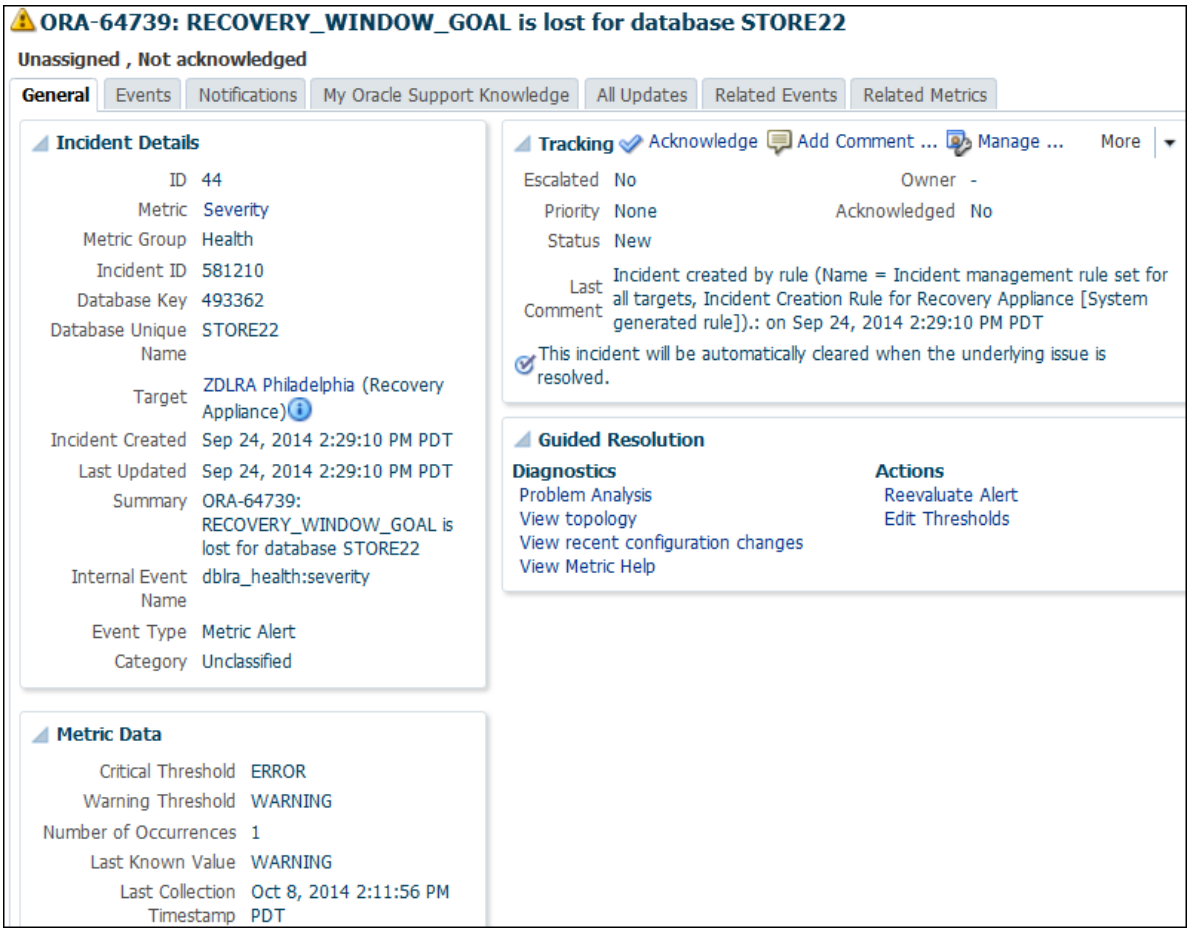
1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. Review the Incidents and Events section for possible problems.

   For example, the section shows the following warning:

   ```
   ORA-64739: RECOVERY_WINDOW_GOAL is lost for database STORE22
   ```

3. Click the summary link of the incident that you are interested in.

   The Incident Manager page for the selected warning appears, with the General subpage selected:



4. Click the subpages to get detailed information about the incident.

> **See Also:**
>
> Cloud Control online help to learn more about the Incident Manager

Monitoring Performance

Recovery Appliance ships with two utilities—rastat.pl and network_throughput_test.sh—that can assist you in evaluating the performance of your system.

Generating Performance Statistics by Using the rastat Utility

`rastat.pl` is a command line utility that runs tests against the Recovery Appliance to gather performance statistics which can help you identify system bottlenecks.

The tests can generate statistics on:

- backup data sent to the Recovery Appliance over the network
- restore data received from the Recovery Appliance over the network
- Recovery Appliance ASM disk group read or write I/O

- Recovery Appliance container file read or write I/O
- Recovery Appliance container file allocation rate

The utility is a Perl script that can be run from any Linux or Unix-based client machine that is either a protected database or an upstream Recovery Appliance. The I/O tests however, can also be run directly from the Recovery Appliance server.

You can run multiple tests in parallel on one or more protected databases to simulate a real environment. Each test result represents the performance of an individual client. Note that ongoing activities between other protected databases and the Recovery Appliance being tested, such as backup and restore or other testing, can impact the resulting statistics.

Prerequisites for Running the rastat Utility

Before you run the rastat utility, ensure that the following requirements are met:

- The platform on which you will be running rastat is either Linux or Unix.
- If you will be running the utility from a protected database, copy the `rastat.pl` file from the `/opt/oracle.RecoveryAppliance/client/` directory of a Recovery Appliance compute server to the protected database.
- Complete the steps to enroll the protected database with the Recovery Appliance as described in "Enrolling Protected Databases".
- Ensure that the `$ORACLE_HOME` and `$ORACLE_SID` environment variables are configured if you do not plan to set them by using the applicable options when you run the utility.

Running the rastat Utility

This section describes how to run rastat.pl and provides several examples of how to execute various performance tests, along with sample output.

> **Note:**
>
> If the `NETBACKUP` and `NETRESTORE` tests do not display the results to the standard output, you can view results by looking at the sbtio<pid>.log files.

To run the rastat utility:

1. Ensure that the system from which you are running the utility meets the requirements, as described in "Prerequisites for Running the rastat Utility".
2. Open a command prompt window.
3. Enter the applicable command syntax for the tests you want to run, and press Enter.

   Refer to the "rastat Utility Reference " for information about the general syntax and the options for each test.

Example 1: Running rastat to Test Backup Performance

In the following example, the NETBACKUP test is specified, the backup file size is set to 2048 megabytes, the Recovery Appliance VPC user connection string is supplied, and the RMAN configuration is set by using the --parms option.

```
>$ORACLE_HOME/perl/bin/perl rastat.pl --test=NETBACKUP --filesize=2048M
--catalog=rman/rman@inst2 --parms='SBT_LIBRARY=/u01/oracle/lib/libra.so,
ENV=(RA_WALLET=location=file:/u01/oracle/dbs/ra_wallet
credential_alias=ra-scan:1521/zdlra5:dedicated)'

NETWORK TEST FROM PROTECTED DATABASE TO RECOVERY APPLIANCE

393 MB/s, 2048 MB sent
```

Example 2: Running rastat to Test I/O Reads from a Recovery Appliance ASM Disk Group

In the following example, the ASMREAD test is specified, the test file size is set to 2048 megabytes, the Recovery Appliance SYS user connection string is supplied, and +RCVAREA is specified as the disk group to read from.

```
>$ORACLE_HOME/perl/bin/perl rastat.pl --test=ASMREAD --filesize=2048M
--rasys=admin/admin@inst2 --diskgroup=+RCVAREA

RECOVERY APPLIANCE READ IO TEST FROM DISK

Disk Group: +RCVAREA

2048 MB, 6.06s read IO time, .65s CPU time, 337.99 MB/s, 10.79% CPU usage

PL/SQL procedure successfully completed.
```

Example 3: Running rastat to Test I/O Writes to a Recovery Appliance Container Group

In the following example, the CONTAINERWRITE test is specified, the test file size is set to 2048 megabytes, the Recovery Appliance SYS user connection string is supplied, and the BLOCK_POOL container group is specified as the disk group to write to.

```
>$ORACLE_HOME/perl/bin/perl rastat.pl --test=CONTAINERWRITE --filesize=2048M
--rasys=admin/admin@inst2 --diskgroup=/:BLOCK_POOL

RECOVERY APPLIANCE WRITE IO TEST TO CONTAINER FILES

Disk Group: /:BLOCK_POOL

2048 MB, 9.55s write IO time, 3.50s CPU time, 214.35 MB/s, 36.60% CPU usage

PL/SQL procedure successfully completed.
```

Example 4: Running rastat to Test File Allocation to a Recovery Appliance Container Group

In the following example, the CONTAINERALLOC test is specified, the test file size is set to 2048 megabytes, the Recovery Appliance SYS user connection string is supplied, and the BLOCK_POOL container group is specified as the disk group to write to.

```
>$ORACLE_HOME/perl/bin/perl rastat.pl --test=CONTAINERALLOC --filesize=2048M
--rasys=admin/admin@inst2 --diskgroup=/:BLOCK_POOL
```

```
RECOVERY APPLIANCE CONTAINER FILE ALLOCATION TEST

Disk Group: /:BLOCK_POOL

2048 MB, 6.24s allocation time, 3.69s CPU time, 328.34 MB allocated per second,
59.09% CPU usage

PL/SQL procedure successfully completed.
```

Testing Network Throughput

You can measure theoretical network throughput in a Recovery Appliance environment by using the network_throughput_test.sh script that ships with the appliance.

See My Oracle Support Note Doc ID 2022086.1 (http://support.oracle.com/epmos/faces/DocumentDisplay?id=2022086.1) for information and instructions on how to use the utility.

11

Accessing Recovery Appliance Reports

This chapter explains how to access the pre-created Oracle Business Intelligence (BI) Publisher reports in Oracle Enterprise Manager Cloud Control (Cloud Control). To learn how to access the Backup Reports page for a protected database in Cloud Control, see *Zero Data Loss Recovery Appliance Protected Database Configuration Guide*.

This chapter contains the following topics:

About Recovery Appliance Reports

This section contains the following topics:

> **See Also:**
>
> "Protection Policies" for an architectural overview

Purpose of Recovery Appliance Reports

A principal task for a Recovery Appliance administrator is storage capacity planning. Through BI Publisher, Recovery Appliance provides pre-created BI Publisher reports that enable you to meet the following goals:

- Ensure that the Recovery Appliance has sufficient storage space for its needs

  By using the capacity reports, you can plan for additional storage, reduce the number of new protected databases added to the Recovery Appliance, or adjust protection policies so that the aggregate recovery window space decreases.

- Ensure that the network is not overloaded

  The capacity reports also indicate whether the Recovery Appliance has maximized network capacity. In some cases, you can reduce network load by redistributing network traffic more evenly throughout the day. If network traffic is not distributed, and if network peaks are close to maximizing network bandwidth, then you may need to adjust the backup window times of some protected databases.
- Provide a good view of system performance and activity for service requests
- Obtain a brief or highly detailed status report for any protected database, which can sometimes be useful for troubleshooting databases that are not meeting recovery window goals

Overview of Recovery Appliance Reports

BI Publisher is an enterprise reporting solution for authoring, managing, and delivering reports and other documents.

Pre-Created BI Publisher Reports

Recovery Appliance provides the following pre-created BI Publisher reports:

- Capacity Planning Summary

  This report provides an overview of storage for the Recovery Appliance so you can forecast when it will run out of capacity. Of special usefulness is the summary table, which provides a quick view of the number of days until capacity is exceeded. The network capacity planning summary provides a view of the aggregated network traffic over various time periods. This view shows both average and maximum rates, which are based on network samples.
- Capacity Planning Details

  This report provides a finer granularity of information about capacity planning. It provides information on storage capacity, network throughput, CPU utilization for each protected database host, and disk and flash storage I/O throughput over time. Unlike the capacity planning summary, the detailed report also has memory and IOPS summary information, and detailed daily data.
- Recovery Window Summary

  This report lists protected databases that are not meeting their recovery window goal or are exceeding their unprotected window threshold (see `unprotected_window` in "CREATE_PROTECTION_POLICY"). You can use this report as a quick view of recovery window and unprotected data window issues for a Recovery Appliance, and then follow up on individual protected databases using the Protected Database Details report.
- Protected Database Details

  This report contains extensive status information about a protected database, including summaries of the following:

  – The protection policy
  – Recovery Appliance storage location
  – The disk recovery window goal

  - The reserved space, which is the minimum amount of disk space in the Recovery Appliance reserved for the database to meet its disk recovery window goal
  - The status of real-time redo transport, which eliminates data loss
  - Data sent and received over time for backup, copy-to-tape, and Recovery Appliance replication operations, which gives a good overview of the traffic coming to and from the protected database to the Recovery Appliance

- Top 10 Protected Databases by Data Transfer

  This report ranks the top 10 protected databases according to the amount of backup data transferred to or from the Recovery Appliance. The report aggregates data by hour or by day. Specifically, the report measures the following amounts:

  - Backup data sent to the Recovery Appliance
  - Replication data sent by the Recovery Appliance
  - Copy-to-tape data sent by the Recovery Appliance

  This report does not correlate directly to how much space is being used by backups of the ranked databases.

The preceding reports are *only* accessible through BI publisher. Also, the BI Publisher software included with Cloud Control can only access data in the Enterprise Manager Repository, not data in the Recovery Appliance schema.

> **Note:**
>
> Although performance data is accessible through SQL queries of V$ and recovery catalog views, Oracle highly recommends that you use the BI Publisher reports instead. Recovery Appliance has finite CPU and other resources, so if users run frequent or expensive SQL queries, then overall Recovery Appliance performance can suffer.

> **See Also:**
>
> - Recovery Appliance View Reference for a complete list of all the Recovery Appliance views
> - Oracle Enterprise Manager Licensing Information

BI Publisher Report Scheduling

Oracle recommends that you configure BI Publisher to generate reports automatically on a regular schedule (for example, every week), and to send the reports by email to the backup management team. You can also generate reports as needed using the techniques described in this chapter.

> **See Also:**
>
> *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Publisher* to learn how to schedule reports using BI Publisher

Accessing the Recovery Appliance Reports Page in Cloud Control

This section explains how to access the Recovery Appliance Reports page, which links to all pre-created BI Publisher reports.

To access the Recovery Appliance Reports page:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".
2. From the **Enterprise** menu, select **Reports**, and then **BI Publisher Enterprise Reports**.

   The BI Publisher Enterprise Reports page appears.
3. Expand the **Enterprise Manager Cloud Control** folder, and then expand the **Recovery Appliance Reports** subfolder.

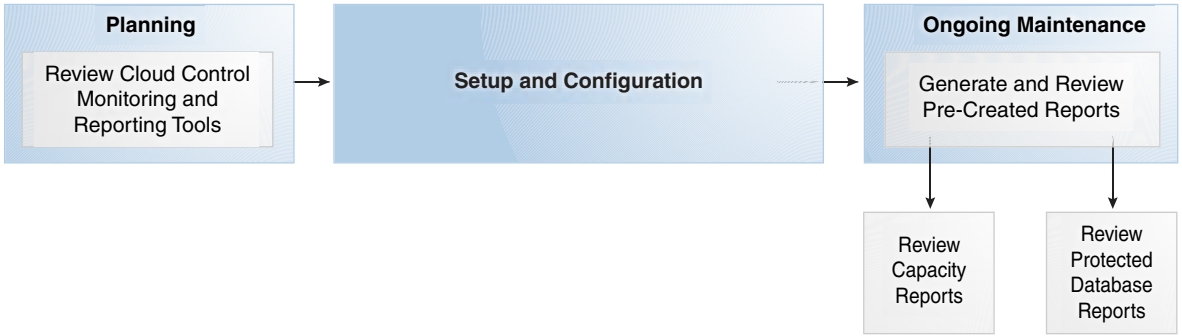
   Links to the pre-created reports are shown.

> **See Also:**
>
> Cloud Control online help to learn how to view reports in BI Publisher

Basic Tasks for Accessing Recovery Appliance Reports

This section explains the basic tasks involved in managing reports. Figure 11-1 shows the overall workflow described in Recovery Appliance Workflow , with the reporting tasks highlighted.

Figure 11-1 Reporting Tasks in the Recovery Appliance Workflow



Typically, you perform reporting tasks in the following sequence:

1. During the planning phase, familiarize yourself with the monitoring and reporting tools available through Cloud Control.

   "Planning for Recovery Appliance" describes these tasks.

2. During the ongoing maintenance phase (see "Maintenance Tasks for Recovery Appliance"), review the reports as needed. Typical tasks include:

   - Review the Storage Capacity Planning Summary every week, using the Capacity Planning Details to get more detailed information.

     "Accessing the Storage Capacity Reports" describes this task.

   - Review the protected database reports as needed:

     – "Accessing the Recovery Window Summary Report" describes this task.

     – "Accessing the Protected Database Details Report from the Recovery Appliance Reports Page" describes this task.

     – "Accessing the Protected Database Data Transfer Report" describes this task.

Accessing the Storage Capacity Reports

This tutorial explains how to view the storage capacity reports for a Recovery Appliance.

Assumptions

Assume that the following statements are true of your Recovery Appliance environment:

- The Recovery Appliance named `ZDLRA London` was set up over two weeks ago.
- 20 protected databases back up to `ZDLRA London`.

To review reports on storage capacity:

1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. Click **Capacity Planning Summary**.

   The BI Publisher Enterprise page appears.
3. Enter your BI Publisher credentials, and then click **Sign In**.

   The Capacity Planning Summary page appears.
4. In Recovery Appliance, select a Recovery Appliance, and then click **Apply**.

   For example, select **ZDLRA London**.
5. Review the Storage Capacity Planning Summary section to determine the storage growth rate and days until capacity is exceeded.

   For example, the following graphic shows the summary for ZDLRA London:

Storage Capacity Planning Summary

	Last 7 Days	Last 31 Days	Last Year
Storage Growth Rate (GB/day, average)	2557.17	No Data Available	No Data Available
Days Until Capacity is Exceeded*	6.85	No Data Available	No Data Available

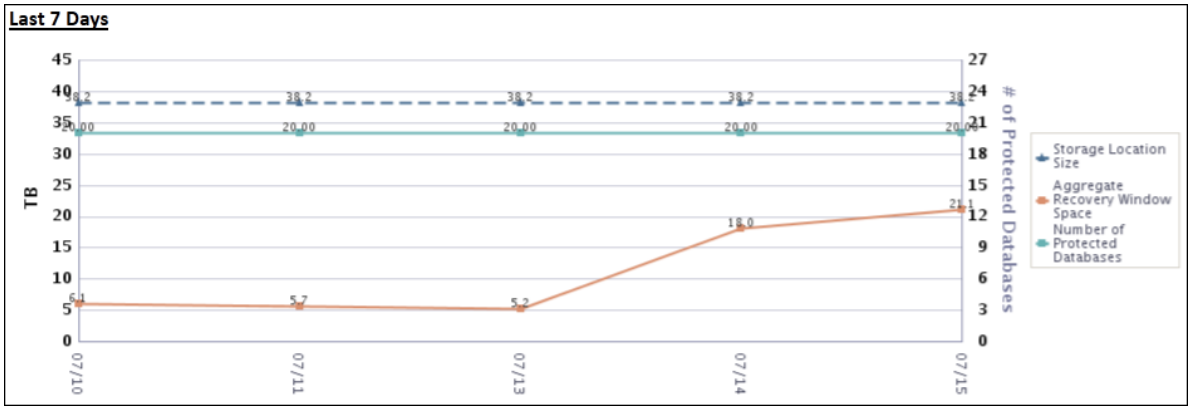
*The appropriate number of days in which the needed aggregate recovery window space will exceed the available storage location space, calculated at the weekly, monthly and yearly average growth.

** Needed recovery window space has decreased during this period.

The key metric is the 6.85 days until capacity is exceeded. Within a week, the Recovery Appliance will either purge backups or reject incoming backups, depending on the protection policy settings (see "Guaranteed Copy").

6. View the historical storage trends.

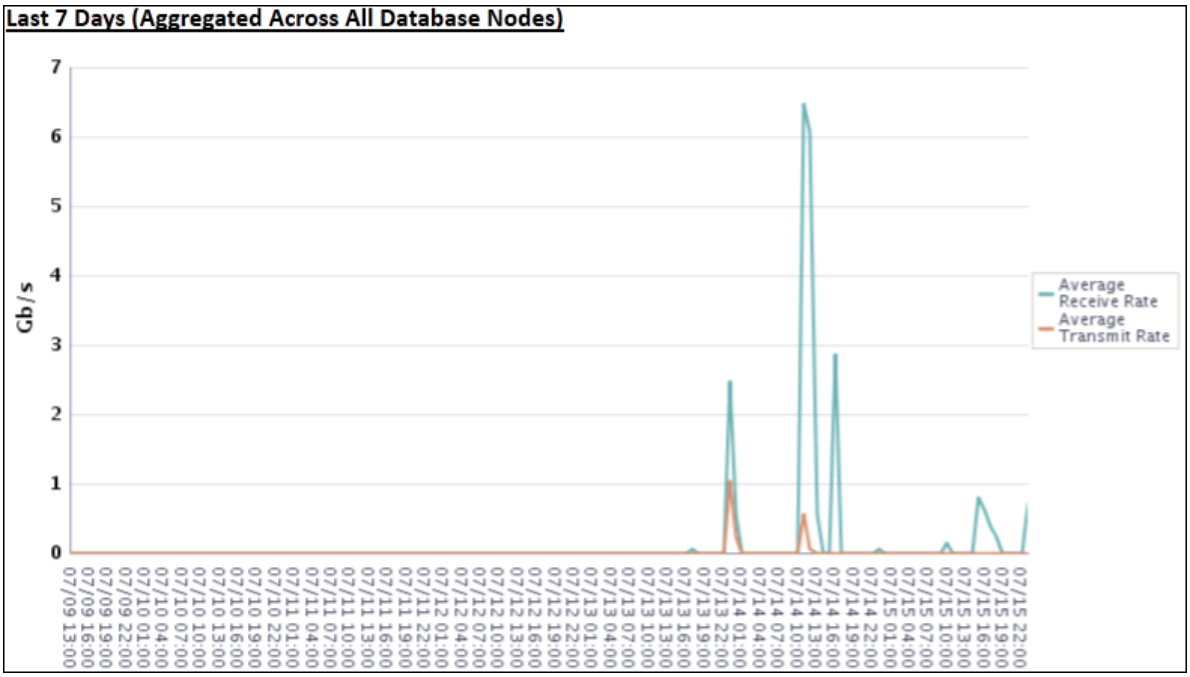
   For example, examine the trends for the past week:



The preceding chart shows that over the last two days, the total space needed to satisfy the recovery window goals for all protected databases has increased from 5.2 TB to 21 TB. The number of protected databases has been steady at 20.

7. In the Network Capacity Planning Summary, check the network throughput.

   For example, examine the network throughput for the past week:



The preceding chart shows that the receive rate spiked during the last two days.

The network capacity planning summary provides a view of the aggregated network traffic over various time periods (24 hours, 7 days, and 30 days).

8. Return to the pre-created reports page.
9. Click **Capacity Planning Details**.

   The Recovery Appliance: Capacity Planning - Details page appears.

   This report provides an even finer granularity of information than the capacity planning summary. Unlike the capacity planning report, the detailed report also has memory and IOPS information.
10. If the desired Recovery Appliance is not already selected in Recovery Appliance, then select it and click **Apply**.
11. At the beginning of the report, click **Storage Capacity Planning Details** to jump to this section.
12. Scroll down to the Cell Disk IO subsection of the IO Historical Data section.

    For example, for each storage server, the following table shows the IOPS and throughput at 15 minute intervals:

Daily Data	Read IOPS	Write IOPS	Total IOPS	Read Throughput MB/sec	Write Throughput MB/sec	Disk Util %
07/17 11:00	8	12	20	5.62	1.57	0
07/17 10:45	10	14	25	2.39	2.94	0
07/17 10:30	16	32	48	6.27	5.15	.14
07/17 10:15	105	107	211	61.05	29.88	5.03
07/17 10:00	175	68	243	90.55	18.23	5.81

Accessing the Recovery Window Summary Report

This tutorial explains how to view the recovery window summary of the databases protected by your Recovery Appliance.

Assumptions

Assume that the following statements are true of your environment:

- The Recovery Appliance named `ZDLRA Montreal` protects 13 databases.
- You want to determine whether any database has not been meeting its recovery window.

To review the recovery window summary report:

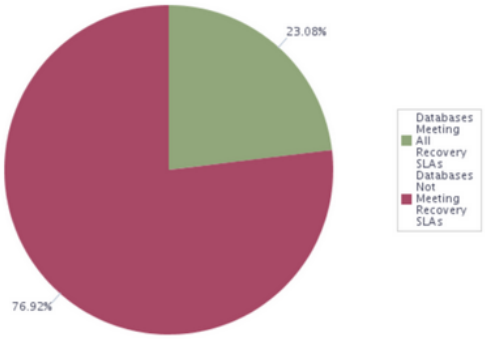
1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. Click **Recovery Window Summary**.

   The Recovery Appliance: Recovery Window Summary page appears.
3. Enter your BI Publisher credentials, and then click **Sign In**.

   The Recovery Appliance: Recovery Window Summary page appears.
4. In Recovery Appliance, select a Recovery Appliance, and then click **Apply**.

   For example, select **ZDLRA Montreal**.

   The page refreshes, displaying the summary at the top, as in the following example:



		ZDLRA Montreal						Total
		BRONZE	TEST	GOLD	NZTEST	SILVER	SOLARIS	
Databases Meeting All Recovery SLAs	Meeting Recovery Window Goal and Within Unprotected Data Window Threshold	0	0	1	0	0	2	3
Databases Not Meeting Recovery SLAs	Exceeding Unprotected Data Window Threshold and Not Meeting Recovery Window Goal	0	1	0	0	0	0	1
	Exceeding Unprotected Data Window Threshold but Meeting Recovery Window Goal	2	0	3	1	1	0	7
	Not Meeting Recovery Window Goal but Within Unprotected Data Window Threshold	0	0	0	0	2	0	2
Total		2	1	4	1	3	2	13

The pie chart shows that over 75% of the databases are not meeting their service level agreements. The table shows that out of 13 total databases, 3 databases are not meeting their recovery window goal. Also, 8 databases are not within their unprotected data threshold.

5. Scroll down to charts showing the databases not meeting their recovery window goals.

   The following example shows two sample reports:

Not Meeting Recovery Window Goal but Within Unprotected Data Window Threshold

Database Name	Protection Policy	Recovery Window Goal (Days Hours:Minutes:Seconds)	Current Recovery Window (Days Hours:Minutes:Seconds)	Unprotected Data Window Threshold (Days Hours:Minutes:Seconds)	Unprotected Data Window (Days Hours:Minutes:Seconds)	Real-Time Redo Transport
DB1116	SILVER	001 00:00:00	000 01:05:40	000 12:00:00	000 01:02:54	Disabled
DB1123	SILVER	001 00:00:00	000 02:45:46	000 12:00:00	000 01:42:58	Disabled

Exceeding Unprotected Data Window Threshold and Not Meeting Recovery Window Goal

Database Name	Protection Policy	Recovery Window Goal (Days Hours:Minutes:Seconds)	Current Recovery Window (Days Hours:Minutes:Seconds)	Unprotected Data Window Threshold (Days Hours:Minutes:Seconds)	Unprotected Data Window (Days Hours:Minutes:Seconds)	Real-Time Redo Transport
DB11242	TEST	001 00:00:00	000 00:49:09	000 00:00:05	001 00:38:47	Disabled

The preceding example shows that databases `DB1116` and `DB1123` both have recovery window goals of 1 day, but have recovery windows that are many hours short of this goal. `DB11242` also has a goal of 1 day, but has an actual recovery window of less than an hour.

Accessing the Protected Database Details Report

You can access the Protected Database Details report either from the Recovery Appliance Reports page or the Protected Database page.

Accessing the Details Report from the Protected Databases Page

This access path is an alternative to using the Enterprise menu to go to the Recovery Appliance Reports page.

To access the Protected Database Details page:

1. Access the Recovery Appliance Home page, as described in "Accessing the Recovery Appliance Home Page".

   For example, access the home page for `ZDLRA Montreal`.

2. From the **Recovery Appliance** menu, select **Protected Databases**.

   The Protected Databases page appears.

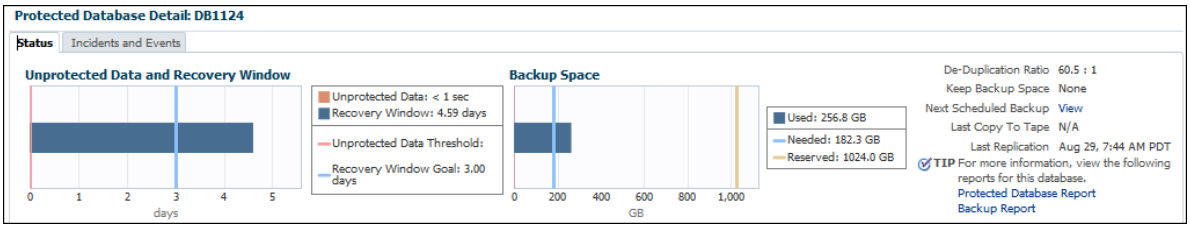
3. In the Protected Databases table, select a database.

   For example, select **DB1124**.

   At the bottom of the page, the Protected Database Detail section is refreshed.

4. In the Protected Database Detail section, select **Status**.

   The Status subpage is displayed, as shown in the following example:



5. Click **Protected Database Report**.

   The BI Publisher page appears.

6. Enter your BI Publisher credentials, and then click **Sign In**.

   The Recovery Appliance: Protected Database Details page appears (see Step 4 in "Accessing the Protected Database Details Report from the Recovery Appliance Reports Page").

Accessing the Protected Database Details Report from the Recovery Appliance Reports Page

This tutorial explains how to access detailed reports for protected databases.

Assumptions

Assume that Recovery Appliance named `ZDLRA Philadelphia` protects 12 databases.

To review reports on protected databases:

1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. Click **Protected Database Details**.

   The Recovery Appliance: Protected Database Details page appears.

   > **Note:**
   >
   > You can also access this report directly from the Protected Databases page, as explained in "Accessing the Details Report from the Protected Databases Page".

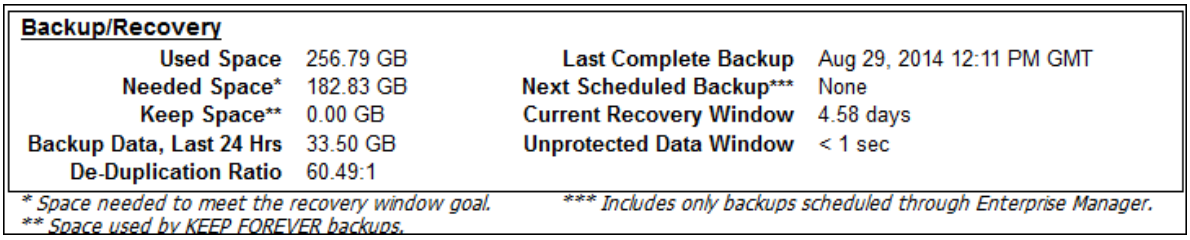
3. In Recovery Appliance, select a Recovery Appliance.

   For example, select **ZDLRA Philadelphia**.
4. In Protected Database, select a database, and then click **Apply**.

   For example, select **DB1124**.

   The Recovery Appliance: Protected Database Details page appears.
5. Scroll down to the Backup/Recovery section.

   For example, the following graphic shows the statistics for the DB1124 database:



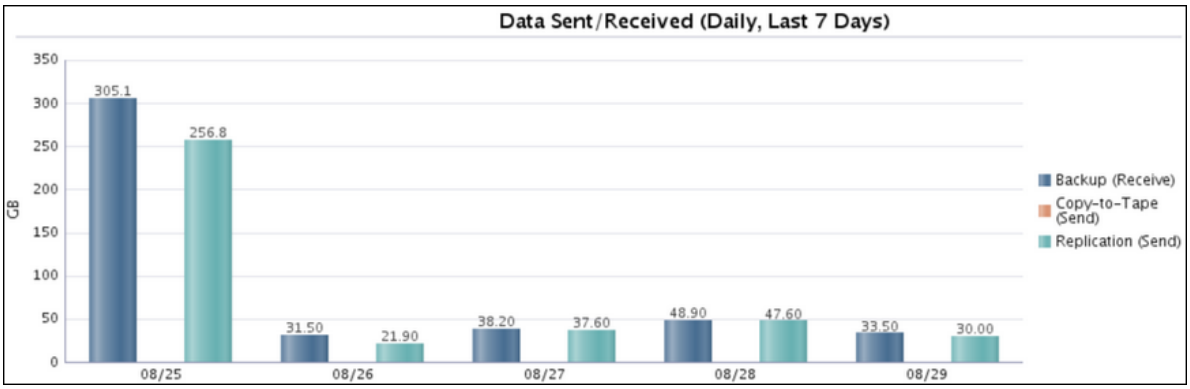
Backup/Recovery

Used Space	256.79 GB	**Last Complete Backup**	Aug 29, 2014 12:11 PM GMT
Needed Space*	182.83 GB	**Next Scheduled Backup*****	None
Keep Space**	0.00 GB	**Current Recovery Window**	4.58 days
Backup Data, Last 24 Hrs	33.50 GB	**Unprotected Data Window**	< 1 sec
De-Duplication Ratio	60.49:1		

** Space needed to meet the recovery window goal.*
*** Space used by KEEP FOREVER backups.*
**** Includes only backups scheduled through Enterprise Manager.*

   The preceding statistics indicate that the database needs 182.83 GB to meet its recovery window goal.
6. Examine the backup history.

   For example, the following chart shows the data sent and received over the past week by DB1124:



The preceding chart shows that the daily backup data was between 31 and 49 GB every day except 8/25, when it was 305.1 GB. On every day, the amount of replication data sent was slightly less than the backup data received.

7. Repeat the preceding two steps as needed to investigate the backup activity of all the protected databases.

Accessing the Protected Database Data Transfer Report

This tutorial explains how to view the data transfer reports for a Recovery Appliance.

Assumptions

Assume that the following statements are true of your environment:

- The Recovery Appliance named `ZDLRA Montreal` protects 10 databases.
- The Recovery Appliance replicates backups for some databases, but does not archive to tape.
- You want to determine which database has been transferring the most data over the past week.

To review the data transfer report:

1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. Click **Top 10 Protected Databases by Data Transfer**.

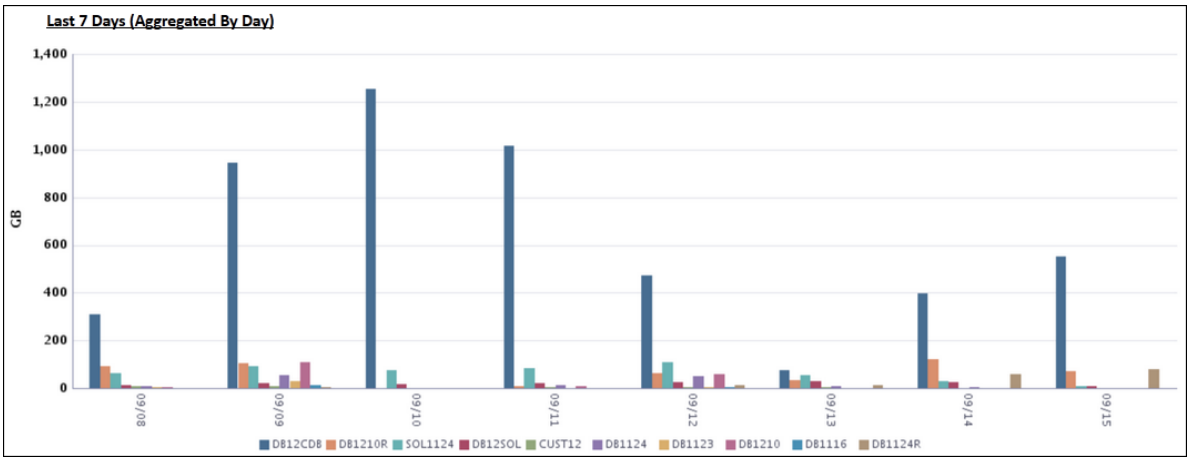
   The BI Publisher Enterprise page appears.
3. Enter your BI Publisher credentials, and then click **Sign In**.

   The Top 10 Protected Databases by Data Transfer page appears.
4. In Recovery Appliance, select a Recovery Appliance, and then click **Apply**.

   For example, select **ZDLRA Montreal**.
5. Click **Top 10 Databases by Backup Data**.

   The Backup Data section is shown.
6. Scroll down to the Last 7 Days (Aggregated By Day) section.

   For example, the period from 09/08 to 09/15 shows the following data:



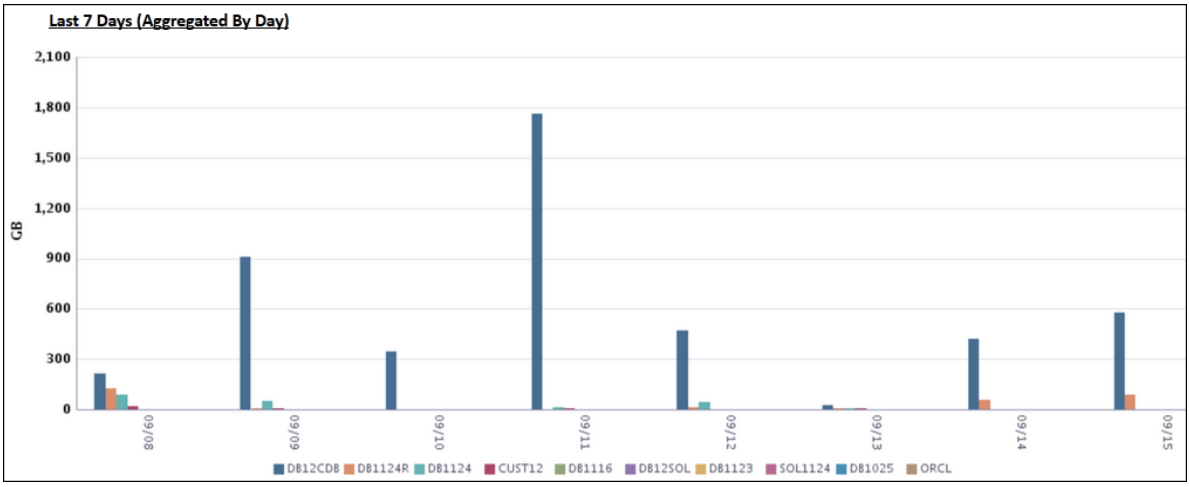
According to the preceding chart, database DB12CDB backed up the most data every day. The peak transfer was on 09/09, with about 1.25 TB.

7. Return to the top of the report, and then click **Top 10 Databases by Replication Data**.

   The Replication Data section is shown.

8. Scroll down to the Last 7 Days (Aggregated By Day) section.

   For example, the period from 09/08 to 09/15 shows the following data:



According to the preceding chart, database DB12CDB replicated the most data every day. The peak transfer was on 09/11, with around 1.7 TB.

Accessing the Protected Database Chargeback Report

This tutorial describes how to view the charge calculation information for a protected database.

- About the Protected Database Chargeback Report

- Viewing the Protected Database Chargeback Report

About the Protected Database Chargeback Report

The Protected Database Chargeback report is used for chargeback of protected databases enrolled with Recovery Appliance. It displays data in both graph and pivot table formats.

The Protected Database Chargeback report is not an out-of-the-box report. After you deploy this report, it can be accessed from the location to which it is deployed.

The report contains the following two versions:

- Recovery Appliance Protected Database Chargeback - Greatest

  Provides charge calculation based on tape storage utilization and the higher value among backup storage utilized and projected recovery window storage utilization. With this model, the consumer pays for the entire recovery window space needed up front.
- Recovery Appliance Protected Database Chargeback - Least

  Provides charge calculation based on tape storage utilization and the lower value among backup storage utilized and projected recovery window storage utilization. With this model, the consumer pays only for the utilized space.

Each version of the Protected Database Chargeback report contains the following two sections:

- Protected Database Space Chargeback on the Recovery Appliance
- Protected Database Space Chargeback on Tape

Each section is further divided into the following subsections:

- Monthly chargeback (last 12 months)

  This subsection contains the following:
  - Space breakdown
  - Budgetary breakdown
- Yearly chargeback

  This subsection contains the following:
  - Space breakdown
  - Budgetary breakdown
- Daily information (last 30 days)

  Displays the maximum current space, maximum recovery window space, and the maximum reserved space for the selected protected database in the last 30 days.

The space breakdown contains the following information:

- Current space

  Represents the amount of disk space on the Recovery Appliance currently being used by this protected database during the indicated time period
- Reserved space

Represents the minimum amount of disk space on the Recovery Appliance reserved for use by this protected database to meet its recovery window goal

- Recovery window space

  Represents the estimated space (in GB) that is needed to meet the recovery window goal for this protected database

The budgetary breakdown contains the following information:

- Chargeback amount

  Represents the fixed amount charged for the database for a given time period

- Chargeback amount delta

  Represents the difference in chargeback between the current and previous reported value

Viewing the Protected Database Chargeback Report

Use Enterprise Manager Cloud Control to access the Protected Database Chargeback report.

Before you access the report, deploy the Protected Database Chargeback report to BI Publisher using the information in My Oracle Support Note Doc ID 2247393.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=2247393.1`).

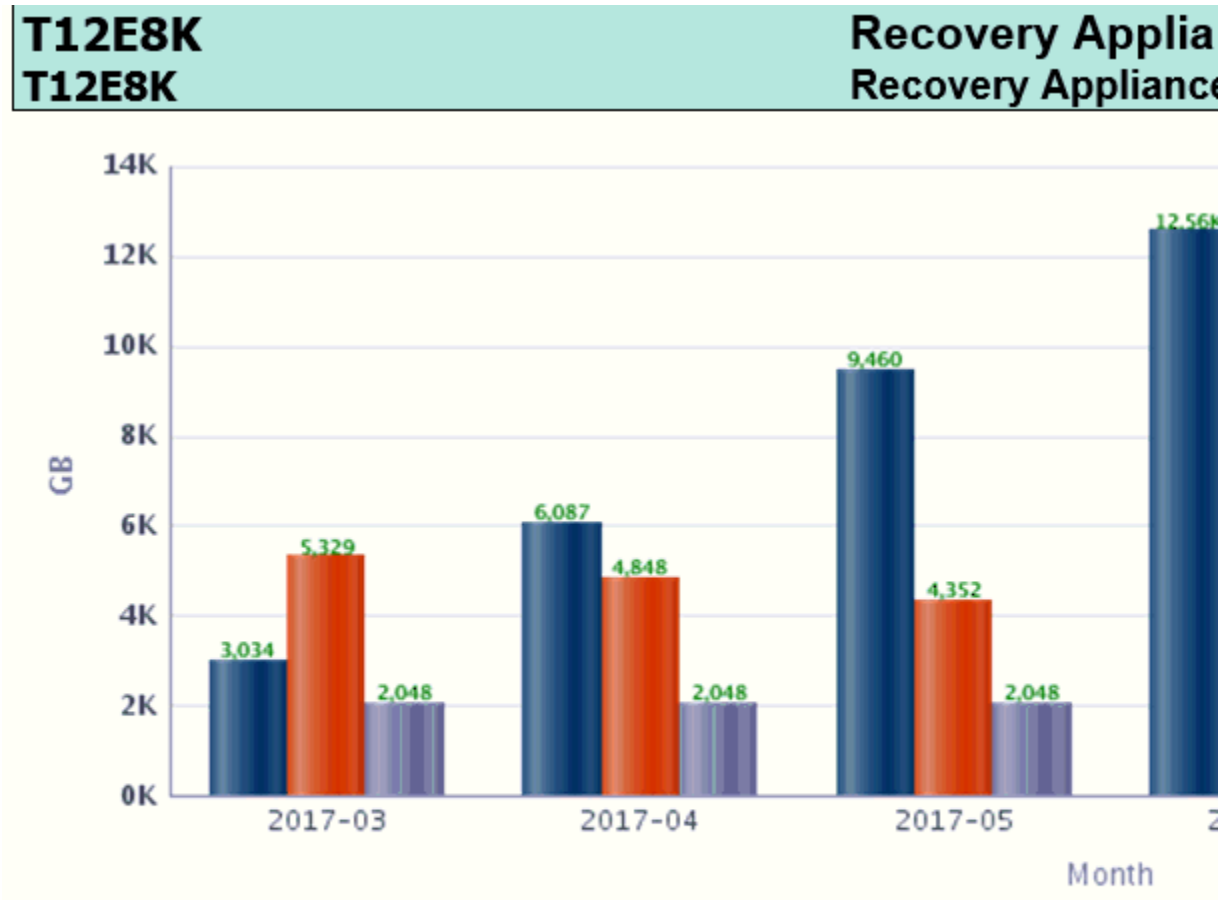
To view the protected database chargeback report:

1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. From the Enterprise menu, select Reports, and then BI Publisher Enterprise Reports.

   The BI Publisher Enterprise Reports page appears.
3. Expand the My Folders folder and then expand the ZDLRA – New BI Reports (Wave 2) folder.
4. Depending on which version of the report you want to access, click **Recovery Appliance - Protected Database Chargeback – Greatest** or **Recovery Appliance - Protected Database Chargeback – Least**.
5. Select the required values in the Recovery Appliance, Protection Policy, Protected Database, Recovery Appliance – Cost $/GB, and Tape – Cost $/GB.
6. Click **Apply** to display the report.

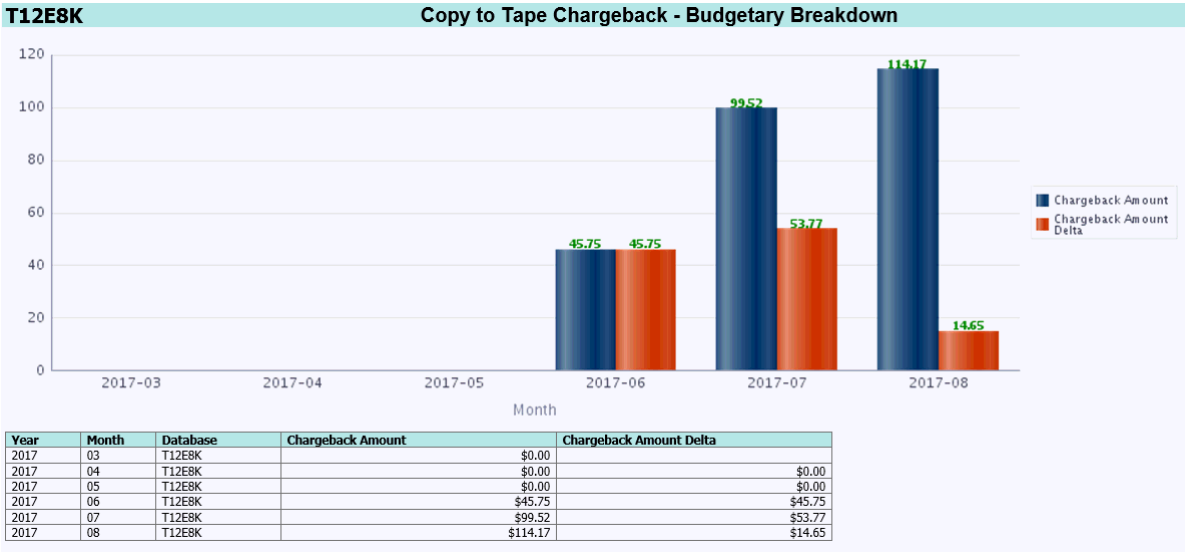
   For both Recovery Appliance and Copy to Tape, the space breakdown and the budgetary breakdown are displayed.

   The following chart shows the monthly space breakdown for the protected database `T12E8K` in the Recovery Appliance. The details tabulated for each month include the maximum current space, maximum recovery space, and the maximum reserved space. The table below the graphic displays the same information in tabular format in table below the graph. Click the column headers to sort for filter the displayed data.



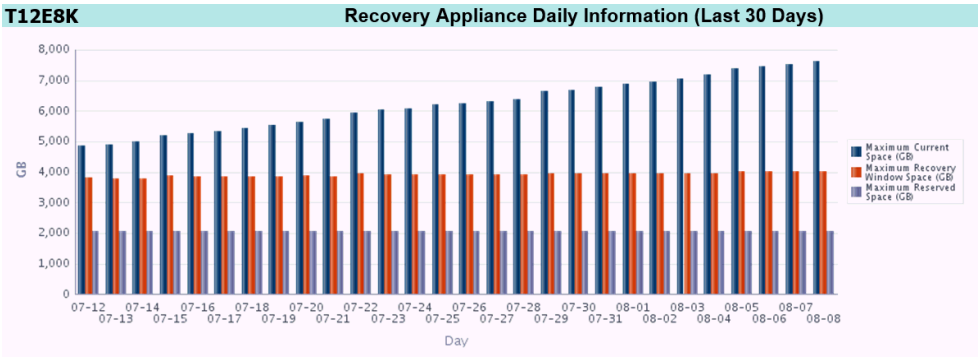
Year	Month	Database	Maximum Current Space (GB)	Maximum Recovery Windo
2017	03	T12E8K	3034.14	
2017	04	T12E8K	6087.21	
2017	05	T12E8K	9459.77	
2017	06	T12E8K	12558.35	
2017	07	T12E8K	12698.74	
2017	08	T12E8K	7606.84	

The monthly budgetary breakdown for tape is displayed below. Data, including the chargeback amount and chargeback amount delta, is displayed in both chart and tabular format.



Year	Month	Database	Chargeback Amount	Chargeback Amount Delta
2017	03	T12E8K	$0.00	
2017	04	T12E8K	$0.00	$0.00
2017	05	T12E8K	$0.00	$0.00
2017	06	T12E8K	$45.75	$45.75
2017	07	T12E8K	$99.52	$53.77
2017	08	T12E8K	$114.17	$14.65

The following image displays the daily information for the Recovery Appliance in the last 30 days. Details include the maximum current space, maximum recovery window space, and maximum reserved space.



Accessing the Active Incidents Report

This tutorial describes how to view the current active incidents for a Recovery Appliance. The data is represented in both pie chart and table format.

The incidents report can be broken down by component, protected database, or incident severity.

Deploy the incident report to BI Publisher using the information in My Oracle Support Note Doc ID 2247391.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=2247391.1`).

Assumptions

Assume that the following statements are true of your environment:

- The steps required to deploy the incidents report to BI Publisher have been performed.

  The active incidents report is not an out-of-the-box report. After you deploy this report, it can be accessed from the location to which it is deployed

To review the active incidents report:

1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. Click **Recovery Appliance Incidents**.

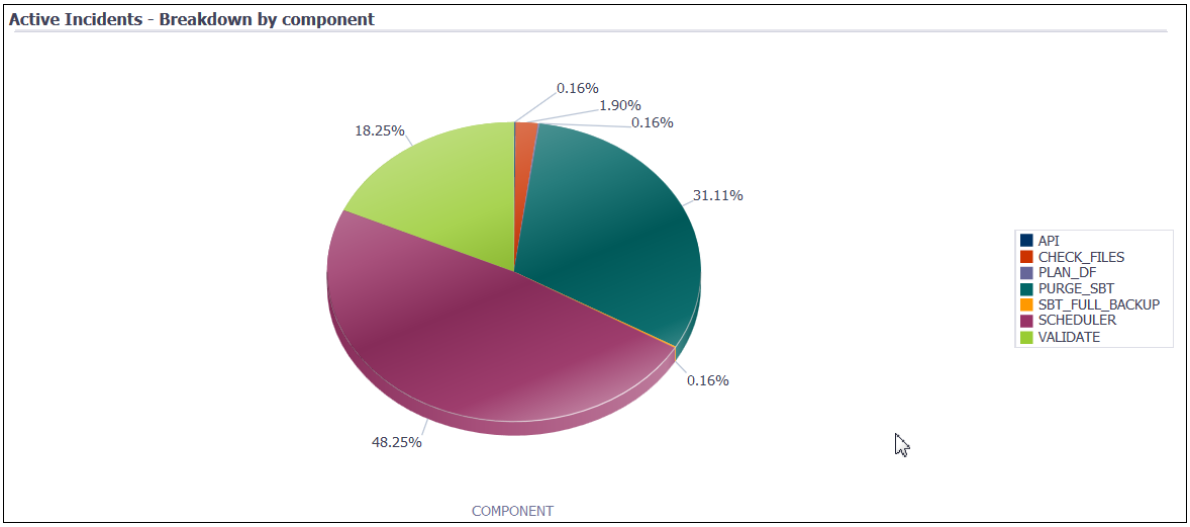
   The Recovery Appliance Incidents page is displayed.
3. From the Recovery Appliance list, select the name of the Recovery Appliance and click **Apply**.

   The active incidents report for the selected Recovery Appliance is displayed. The report contains three pie charts and a table.

   For example, select **ZDLRA Baltimore** and click **Apply**.

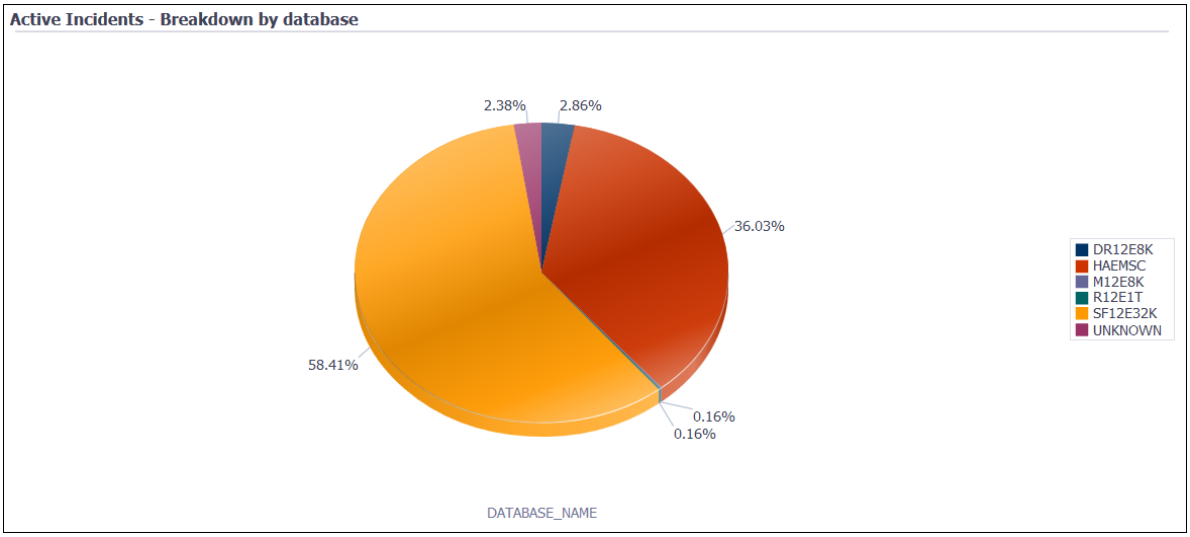
   The page refreshes displaying a report containing the following: a pie chart of active incidents by component, a pie chart of active incidents by protected database, a pie chart of active incidents by severity, and a tabular representation of the report.

   For example, the following pie chart shows the percentage of active incidents for each Recovery Appliance component:



   Click a section in the chart to display active incident details for component represented by the selected section. For example, click the section representing the `SCHEDULER` to view active incident details for the scheduler. The active incidents by database, active incidents by severity, and the tabular representation of the report are automatically updated based on your selection.
4. View the active incidents broken down by protected database.

   For example, the following chart shows the statistics of active incidents reported for six protected databases :



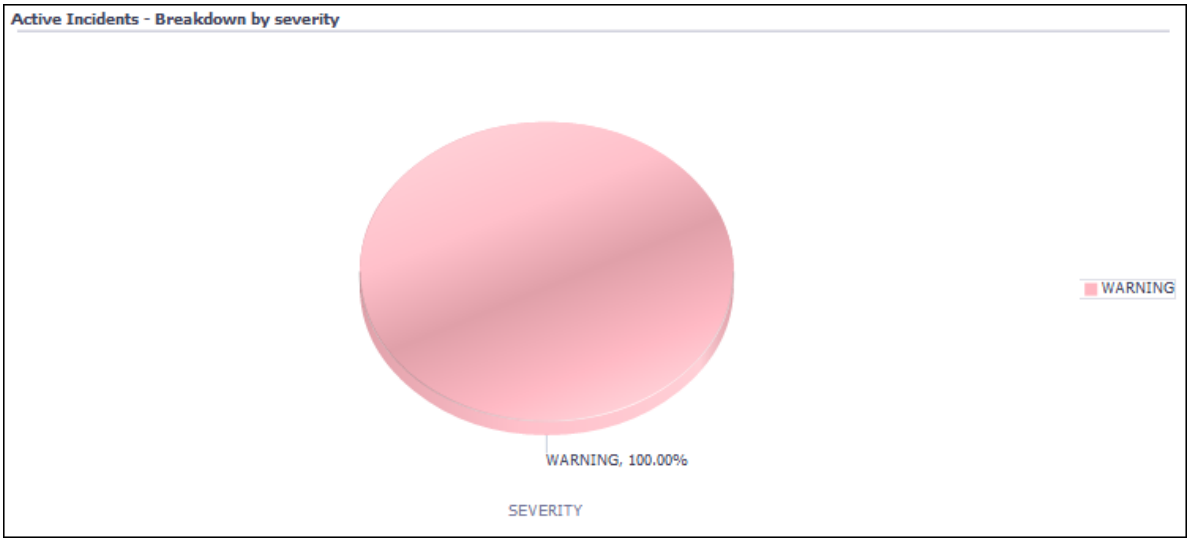
When you click the section representing a particular protected database, the data in the other pie charts and table is updated automatically based on the selection.

5. View the active incidents broken down by severity.

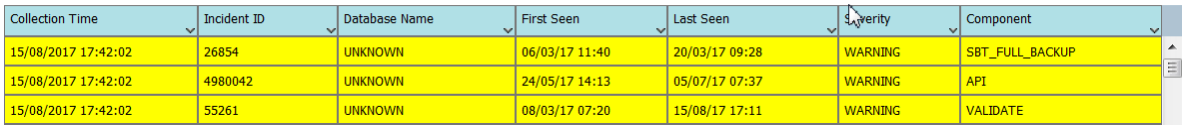
   Incident severity can be one of the following:

   - **Internal severity:** This indicates an error that needs immediate action. It is the highest category of incidents.
   - **Error severity:** This indicates that the operation ended in an error and needs attention.
   - **Warning severity:** This indicates that the operation resulted in a warning. You can review the warning for additional actions that may be need to be performed.

   The following chart shows that there are only warnings, no incidents of internal severity or errors, for the selected Recovery Appliance:



Below this chart is the tabular format of the active history report, as shown in the following example:



Collection Time	Incident ID	Database Name	First Seen	Last Seen	Severity	Component
15/08/2017 17:42:02	26854	UNKNOWN	06/03/17 11:40	20/03/17 09:28	WARNING	SBT_FULL_BACKUP
15/08/2017 17:42:02	4980042	UNKNOWN	24/05/17 14:13	05/07/17 07:37	WARNING	API
15/08/2017 17:42:02	55261	UNKNOWN	08/03/17 07:20	15/08/17 17:11	WARNING	VALIDATE

Accessing the API History Report

This tutorial describes how to view the history of API calls made to procedures in the `DBMS_RA` package. The report displays the data in pie chart and pivot table formats.

The pie chart in the API history report can be broken down by API call. The report also contains a table with the name of the API calls, data they were run, and the number of calls.

Deploy the API history report to BI Publisher using the information in My Oracle Support Note Doc ID 2247392.1 (`http://support.oracle.com/epmos/faces/DocumentDisplay?id=2247392.1`).

Assumptions

Assume that the following statements are true of your environment:

- The steps required to deploy the incidents report to BI Publisher have been performed.

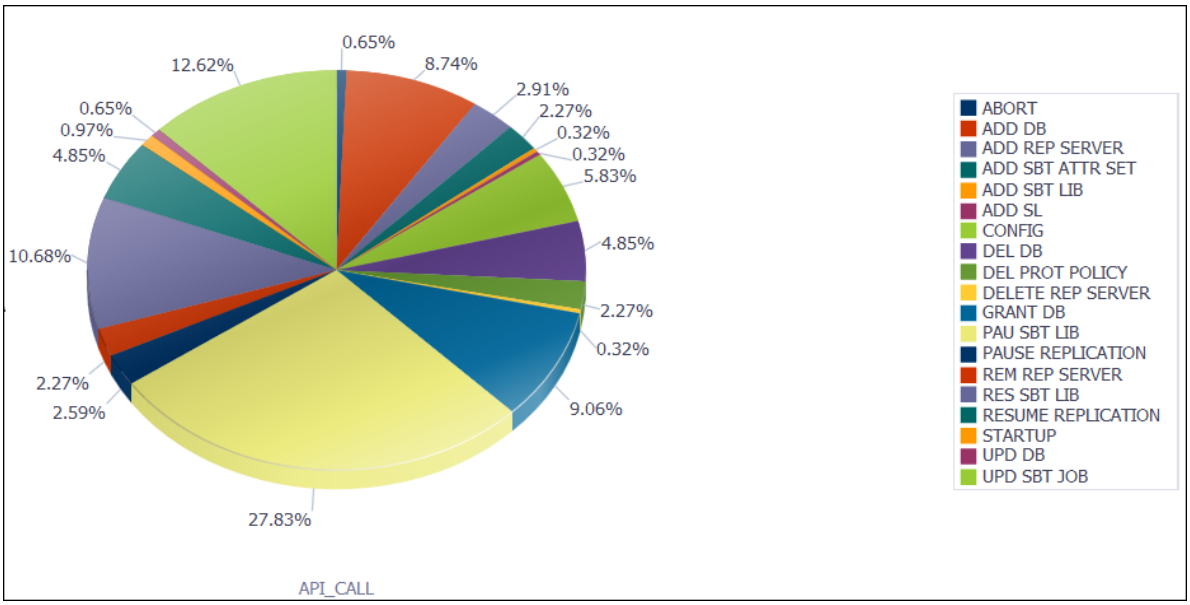
  The API history report is not an out-of-the-box report. After you deploy this report, it can be accessed from the location to which it is deployed

To review the API history report:

1. Go to the pre-created reports page, as described in "Accessing the Recovery Appliance Reports Page in Cloud Control".
2. Click **API History audit**.

   The API History Audit page is displayed.
3. From the Recovery Appliance list, select the name of the Recovery Appliance and click **Apply**.

   The API call history report for the selected Recovery Appliance is displayed. A pie chart displays the statistics for the API calls made, as in the following example:



Click a section in the chart to display the details for the API call represented by the selected section in the table below the chart. For example, click the section on the right of the chart that represents `GRANT DB` to display the history details for this API. The data in the table is automatically updated based on your selection.

4. View the API history in tabular format.

   By default, a date-wise summary with one column for each API call in the report is displayed, as shown in the following example:

	ABORT	ADD DB	ADD REP SERVER	ADD SBT ATTR SET	ADD SBT JOB	ADD SBT LIB
2017-03-03	0	18	0	12	0	
2017-03-04	0	0	0	0	0	
2017-03-06	1	11	0	0	1	
2017-03-10	0	2	0	0	0	
2017-03-14	2	0	0	0	0	
2017-03-15	0	0	0	0	0	
2017-03-20	1	1	0	0	0	
2017-03-21	0	0	0	0	0	
2017-03-23	0	0	0	0	0	
2017-03-26	0	1	0	0	0	
2017-03-27	0	0	0	0	0	
2017-03-29	0	1	0	0	0	
	0	0	1	0	0	

Command Issue Date	Command Issued
03/03/2017	add_db(db_unique_name=>'w12102e8k', protection_policy_name=>'gold', reserved_space=>'2t')
03/03/2017	add_db(db_unique_name=>'r12102e40t', protection_policy_name=>'gold', reserved_space=>'30t')
03/03/2017	add_db(db_unique_name=>'b12102e40t', protection_policy_name=>'gold', reserved_space=>'30t')
03/03/2017	add_db(db_unique_name=>'t12102e8k', protection_policy_name=>'gold', reserved_space=>'10t')
03/03/2017	add_db(db_unique_name=>'b12102e8k', protection_policy_name=>'gold', reserved_space=>'5t')
03/03/2017	add_db(db_unique_name=>'m12102e8k', protection_policy_name=>'gold', reserved_space=>'10t')
03/03/2017	add_db(db_unique_name=>'s12102e8k', protection_policy_name=>'gold', reserved_space=>'2t')
03/03/2017	add_db(db_unique_name=>'r12102e5t', protection_policy_name=>'gold', reserved_space=>'5t')
03/03/2017	add_db(db_unique_name=>'r12102e1t', protection_policy_name=>'gold', reserved_space=>'5t')
03/03/2017	add_db(db_unique_name=>'a12102mix', protection_policy_name=>'gold', reserved_space=>'4t')

Click a cell in the summary to display details for the API call and date represented by the selected cell.

Part II
Recovery Appliance Reference

Part II contains the following chapters:

12

DBMS_RA Package Reference

This chapter provides details on the DBMS_RA PL/SQL package. You use DBMS_RA subprograms to perform all Recovery Appliance administration functions.

One DBMS_RA procedure may execute at a given time. With the exception of ABORT_RECOVERY_APPLIANCE, attempts to run multiple DBMS_RA procedure in different sessions at the same time fail with an appropriate message.

The following table summarizes the available DBMS_RA subprograms.

Table 12-1 DBMS_RS Package Subprograms

Subprogram	Description
ABORT	Synonymous with ABORT_RECOVERY_APPLIANCE.
ABORT_RECOVERY_APPLIANCE	This procedure shuts down the Recovery Appliance without waiting for in-progress operations to complete.
ADD_DB	This procedure adds the specified database to the Recovery Appliance, and assigns a protection policy to the database. This procedure enables a non-protected database to attain the status of a protected database.
ADD_REPLICATION_SERVER	This procedure adds the specified replication server configuration to the specified protection policy. After the operation succeeds, the Recovery Appliance replicates backups of databases protected by this policy to the downstream Recovery Appliance.
CONFIG	This procedure updates a value in the config table.
COPY_BACKUP	This procedure copies one or more backup pieces from the Recovery Appliance to a user-specified disk or SBT destination. The Recovery Appliance copies all backup pieces matching the specified tag to the location specified with the format and template_name parameters.
COPY_BACKUP_PIECE	This procedure copies a single backup piece from the Recovery Appliance to a user-specified disk or SBT destination.
CREATE_POLLING_POLICY	This procedure creates a backup polling policy.
CREATE_PROTECTION_POLICY	This procedure creates a protection policy.
CREATE_REPLICATION_SERVER	This procedure defines a configuration for a downstream Recovery Appliance that forms part of a Recovery Appliance replication scheme.
CREATE_SBT_ATTRIBUTE_SET	This procedure creates an SBT attribute set that SBT jobs can use.

Table 12-1 (Cont.) DBMS_RS Package Subprograms

Subprogram	Description
CREATE_SBT_JOB_TEMPLATE	This procedure creates an SBT job that describes how the Recovery Appliance chooses backups for copying to tape. This form of this overloaded procedure applies to backups for all protected databases assigned to the specified protection policy.
CREATE_SBT_JOB_TEMPLATE	This procedure creates a new SBT backup job. The job describes how the Recovery Appliance chooses backups for copying to tape/cloud. This form of this overloaded procedure applies to backups for a single protected database only, whereas the previous form applies to backups of all databases assigned to a specific protection policy. With the exception of this difference, this procedure and its parameters are identical to the alternative form of this procedure.
CREATE_SBT_LIBRARY	This procedure creates metadata describing an installed media management software library. The Recovery Appliance uses the specified library to copy backups from internal storage either to tape or to other tertiary storage supported by this media manager.
CREATE_STORAGE_LOCATION	This procedure creates Recovery Appliance storage location, which is an object that describes storage to be used by the Recovery Appliance. Recovery Appliance stores all backups in named storage locations.
DELETE_DB	This procedure removes the specified protected database from the Recovery Appliance. The Recovery Appliance deletes all metadata and backups associated with this database, both from disk and SBT. Backups on tape are not affected.
DELETE_POLLING_POLICY	This procedure deletes the specified backup polling policy.
DELETE_PROTECTION_POLICY	This procedure deletes the specified protection policy.
DELETE_REPLICATION_SERVER	This procedure deletes a replication server configuration. The Recovery Appliance removes all metadata relating to the downstream Recovery Appliance.
DELETE_SBT_ATTRIBUTE_SET	This procedure deletes the specified SBT attribute set.
DELETE_SBT_JOB_TEMPLATE	This procedure deletes the specified SBT job template.
DELETE_SBT_LIBRARY	This procedure deletes the metadata describing the specified SBT library.
DELETE_STORAGE_LOCATION	This procedure deletes the specified Recovery Appliance storage location.
GRANT_DB_ACCESS	This procedure grants the necessary privileges to the specified recovery Appliance user account to enable this account to back up, restore, and access recovery catalog metadata for the specified protected database.
KEY_REKEY	This procedure rekeys encryption keys for all databases with existing encryption keys.
KEY_REKEY	This procedure rekeys encryption keys for the specified database with an existing encryption key.

Table 12-1 (Cont.) DBMS_RS Package Subprograms

Subprogram	Description
KEY_REKEY	This procedure rekeys encryption keys for all databases with existing encryption keys in the specified protection_policy
MIGRATE_TAPE_BACKUP	This procedure makes pre-migration tape backups accessible to the Recovery Appliance through the specified SBT library. You must first import metadata about the tape backups into the Recovery Appliance catalog using the RMAN `IMPORT CATALOG` command.
MOVE_BACKUP	This procedure moves one or more long-term archival backup pieces from the Recovery Appliance to a user-specified disk or SBT destination.
MOVE_BACKUP_PIECE	This procedure moves a single long-term archival backup piece from the Recovery Appliance to a user-specified disk or SBT destination.
PAUSE_REPLICATION_SERVER	This procedure pauses replication to the specified downstream Recovery Appliance.
PAUSE_SBT_LIBRARY	This procedure pauses the specified SBT library. The Recovery Appliance allows in-progress copies of backup pieces to complete. However, if backup pieces were queued for copy through this SBT library but not yet copied, then the Recovery Appliance holds them until you resume the SBT library. No new SBT jobs that run against this library can execute until you resume the library (RESUME_SBT_LIBRARY).
POPULATE_BACKUP_PIECE	This procedure pushes the specified backup piece into the delta store.
QUEUE_SBT_BACKUP_TASK	This procedure queues the backup pieces selected by the specified SBT job template for copying to tape. Typically, a scheduling utility such as Oracle Scheduler calls this procedure.
REMOVE_REPLICATION_SERVER	This procedure removes the specified replication server configuration from the specified protection policy. After the operation succeeds, the Recovery Appliance no longer replicates backups of databases protected by this policy to the downstream Recovery Appliance.
RENAME_DB	This procedure changes the name of the specified protected database in the Recovery Appliance metadata.
RESET_ERROR	This procedure modifies the specified error log entry to have the status `RESET`. Errors marked in this fashion do not cause Oracle Enterprise Manager to raise alerts. If the Recovery Appliance determines that the problem is still occurring, then errors that have been reset change to `ACTIVE` status. The primary use of this API is to reset the error status for nonrecurring errors, such as transient media failures.
RESUME_REPLICATION_SERVER	This procedure resumes replication to the specified downstream Recovery Appliance, after a previous call to PAUSE_REPLICATION_SERVER.

Table 12-1 (Cont.) DBMS_RS Package Subprograms

Subprogram	Description
RESUME_SBT_LIBRARY	This procedure resumes a paused SBT library.
REVOKE_DB_ACCESS	This procedure revokes privileges on one protected database from the specified Recovery Appliance user account.
SET_SYSTEM_DESCRIPTION	This procedure sets a descriptive name for users to apply to their Recovery Appliance. The name provided here will be seen in the RA_SERVER view.
SHUTDOWN	Synonymous with SHUTDOWN_RECOVERY_APPLIANCE.
SHUTDOWN_RECOVERY_APPLIANCE	This procedure performs a clean shutdown of the Recovery Appliance.
STARTUP	Synonymous with STARTUP_RECOVERY_APPLIANCE.
STARTUP_RECOVERY_APPLIANCE	This procedure starts the Recovery Appliance after it has been shut down or terminated.
UPDATE_DB	This procedure changes the attributes that are assigned to the specified protected database.
UPDATE_POLLING_POLICY	This procedure modifies the parameters for an existing backup polling policy.
UPDATE_PROTECTION_POLICY	This procedure modifies the parameters for an existing protection policy.
UPDATE_REPLICATION_SERVER	This procedure changes the settings for a replication server configuration.
UPDATE_SBT_ATTRIBUTE_SET	This procedure updates the parameters for the specified SBT attribute set.
UPDATE_SBT_JOB_TEMPLATE	This procedure updates the parameters for the specified SBT job.
UPDATE_SBT_LIBRARY	This procedure modifies the parameters for the specified SBT library.
UPDATE_STORAGE_LOCATION	This procedure allocates additional space for the specified storage location. You cannot reduce the amount of space used by a storage location.

ABORT

Synonymous with ABORT_RECOVERY_APPLIANCE.

Syntax

```
PROCEDURE abort;
```

ABORT_RECOVERY_APPLIANCE

This procedure shuts down the Recovery Appliance without waiting for in-progress operations to complete.

When you use this procedure, the Recovery Appliance terminates backup, restore, and background operations (such as validations, data moves, and copy-to-tape/cloud jobs) that have started but not completed. When the Recovery Appliance restarts, it automatically resumes or restarts backup operations. You must manually restart any terminated backup and restore operations. To perform a clean shutdown, use SHUTDOWN_RECOVERY_APPLIANCE.

Syntax

```
PROCEDURE abort_recovery_appliance;
```

ADD_DB

This procedure adds the specified database to the Recovery Appliance, and assigns a protection policy to the database. This procedure enables a non-protected database to attain the status of a protected database.

Enrolling a database with the Recovery Appliance involves (1) adding the protected database with ADD_DB, (2) granting access to this database to a Recovery Appliance user account (GRANT_DB_ACCESS), and (3) registering this database in the virtual private catalog (RMAN REGISTER DATABASE command). The protected database must be enrolled before Recovery Appliance can process backup and restore operations.

You cannot use this procedure to add additional databases in a physical standby configuration. Such databases will be automatically recognized as they perform recovery catalog resynchronizations.

Syntax

```
PROCEDURE add_db (
   db_unique_name IN VARCHAR2,
   protection_policy_name IN VARCHAR2,
   reserved_space IN VARCHAR2);
```

Parameters

Table 12-2 ADD_DB Parameters

Parameter	Description
db_unique_name	The DB_UNIQUE_NAME of the database to add.
protection_policy_name	The name of the protection policy to assign to the database. The protection policy must exist.

Table 12-2 (Cont.) ADD_DB Parameters

Parameter	Description
`reserved_space`	The amount of disk space that is guaranteed to be available for the protected database. The format of this value is a character string that must contain a number consisting only of the characters `0-9`, followed optionally by one of the following unit specifiers: `K`: Kilobytes `M`: Megabytes `G`: Gigabytes `T`: Terabytes `P`: Petabytes `E`: Exabytes `Z`: Zettabytes `Y`: Yottabytes If no unit is specified, then Recovery Appliance interprets the value as a number of bytes.

ADD_REPLICATION_SERVER

This procedure adds the specified replication server configuration to the specified protection policy. After the operation succeeds, the Recovery Appliance replicates backups of databases protected by this policy to the downstream Recovery Appliance.

See CREATE_REPLICATION_SERVER.

Syntax

```
PROCEDURE add_replication_server (
   replication_server_name IN VARCHAR2,
   protection_policy_name IN VARCHAR2);
```

Parameters

Table 12-3 ADD_REPLICATION_SERVER Parameters

Parameter	Description
`replication_server_name`	The name of the replication server configuration to associate with the protection policy.
`protection_policy_name`	The name of the protection policy to associate with the replication server configuration.

CONFIG

This procedure updates a value in the `config` table.

Syntax

```
PROCEDURE config(
   p_name VARCHAR2,
   p_value VARCHAR2);
```

Parameters

Table 12-4 CONFIG Parameters

Parameter	Description
p_name	The parameter to update. Possible parameters are: check_file_days The frequency with which the Recovery Appliance runs the metadata consistency check in the background. The default frequency is 3 days. crosscheck_db_days The frequency with which the Recovery Appliance performs background updates of the recovery catalog to reflect actions in tape libraries or downstream Recovery Appliances. Examples include deletions of backup pieces. The default frequency is 1 day. optimize_chunks_days The frequency with which the Recovery Appliance performs background re-ordering of blocks in the delta store to reduce disk reads required for restore operations. The default frequency is 7 days. validate_db_days The frequency with which the Recovery Appliance performs background validation of backup pieces. The default frequency is 7 days. percent_late_for_warning The percent threshold at which the Recovery Appliance posts warnings for incomplete background operations. For example, if validate_db_days is 7 and the percent_late_for_warning is 50, then the Recovery Appliance records a warning in the incident log when a database has gone 10.5 (or 7 + ((50/100)*7)) days without being validated. The fault is 100 percent. network_chunksize The message size that the Recovery Appliance uses for transferring backups between itself and protected databases. Also used for replication. The default is 128 MB. All protected databases use this value to determine the unit size in which to send or read backups. For example, if a protected database backup is 1 TB, then the SBT library sends the backup data to the Recovery Appliance in units of network_chunksize. This technique is an optimization that enables the Recovery Appliance to restart the data transfer faster if a failure occurs.
p_value	The new value for the parameter.

COPY_BACKUP

This procedure copies one or more backup pieces from the Recovery Appliance to a user-specified disk or SBT destination. The Recovery Appliance copies all backup pieces matching the specified tag to the location specified with the `format` and `template_name` parameters.

Syntax

```
PROCEDURE copy_backup (
   tag IN VARCHAR2,
   format IN VARCHAR2,
   template_name IN VARCHAR2,
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-5 COPY_BACKUP Parameters

Parameter	Description
`tag`	The tag of the backups to copy. The Recovery Appliance copies all backups matching this tag.
`format`	The naming format of the backup pieces to create. This parameter follows the same rules as the RMAN `FORMAT` parameter.
`template_name`	The media management library template. If null, then `format` points to a disk location. If not null, then the Recovery Appliance copies the backup piece to tape (or cloud), using the media pool referenced in the SBT template name as the copy destination.

Table 12-5 (Cont.) COPY_BACKUP Parameters

Parameter	Description
compression_algorithm	Specifies the compression algorithm. If compression_algorithm is specified, it will override the compression algorithm defined in template_name for this single operation. If template_name is NULL, it defines the compression algorithm for this operation. BASIC: Good compression ratios with potentially lower speed than MEDIUM. LOW: Optimized for speed with potentially lower compression raios than BASIC. MEDIUM: Recommended for most environments. Good combination of compression ratios and speed. HIGH: Best suited for operations over slower networks where the limiting factor is maximum network throughput. OFF: No compression. NULL: (default) indicates that the algorithm defined in the SBT job template should be used. 'LOW', 'MEDIUM', and 'HIGH' are Advanced Compression Options (ACO) which require additional licensing on the protected database for restore operations. OSB licenses are needed to restore backups directly from tape to DB client. OSB licenses include compression, so no ACO is needed. If 3P MMLs are used with compression, then ACO is needed on the target DB.
encryption_algorithm	Specifies the encryption algorithm If encryption_algorithm is specified, it will override the encryption algorithm defined in template_name for this single operation. If template_name is NULL, it defines the encryption algorithm for this operation. Valid value are 'AES128', 'AES192', 'AES256', 'OFF' or the constant equivalents ENC_OFF, ENC_AES128, ENC_AES192, ENC_AES256.

COPY_BACKUP_PIECE

This procedure copies a single backup piece from the Recovery Appliance to a user-specified disk or SBT destination.

Syntax

```
PROCEDURE copy_backup_piece (
   bp_key IN NUMBER,
   format IN VARCHAR2,
   template_name IN VARCHAR2,
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-6 COPY_BACKUP_PIECE Parameters

Parameter	Description
bp_key	The unique key of the backup piece to copy. Obtain this key from the RC_BACKUP_PIECE view.
format	The naming format of the backup piece to create. This parameter follows the same rules as the RMAN FORMAT parameter.
template_name	A media management library template. If null, then format points to a disk location. If not null, then the Recovery Appliance copies the backup piece to tape, using the media pool referenced in the SBT template name as the copy destination.
compression_algorithm	Specifies the compression algorithm. If compression_algorithm is specified, it will override the compression algorithm defined in template_name for this single operation. If template_name is NULL, it defines the compression algorithm for this operation. BASIC: Good compression ratios with potentially lower speed than MEDIUM. LOW: Optimized for speed with potentially lower compression raios than BASIC. MEDIUM: Recommended for most environments. Good combination of compression ratios and speed. HIGH: Best suited for operations over slower networks where the limiting factor is maximum network throughput. OFF: No compression. NULL: (default) indicates that the algorithm defined in the SBT job template should be used. 'LOW', 'MEDIUM', and 'HIGH' are Advanced Compression Options (ACO) which require additional licensing on the protected database for restore operations. OSB licenses are needed to restore backups directly from tape to DB client. OSB licenses include compression, so no ACO is needed. If 3P MMLs are used with compression, then ACO is needed on the target DB.
encryption_algorithm	Specifies the encryption algorithm If encryption_algorithm is specified, it will override the encryption algorithm defined in template_name for this single operation. If template_name is NULL, it defines the encryption algorithm for this operation. Valid value are 'AES128', 'AES192', 'AES256', 'OFF' or the constant equivalents ENC_OFF, ENC_AES128, ENC_AES192, ENC_AES256

CREATE_POLLING_POLICY

This procedure creates a backup polling policy.

A backup polling policy specifies a directory where a protected database places incoming backups or archived redo log files. The policy also specifies the frequency with which the Recovery Appliance looks for backups in the polling location.

When the Recovery Appliance discovers a file through polling, the Recovery Appliance examines the file, and then uses its contents to associate it with a protected database that is registered with the Recovery Appliance. If the Recovery Appliance cannot associate the file with any registered protected database, then the Recovery Appliance logs a warning message and ceases to process the file.

Syntax

```
PROCEDURE create_polling_policy(
   polling_policy_name IN VARCHAR2,
   polling_location IN VARCHAR2,
   polling_frequency IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
   delete_input IN BOOLEAN DEFAULT FALSE);
```

Parameters

Table 12-7 CREATE_POLLING_POLICY Parameters

Parameter	Description
`polling_policy_name`	The user-assigned name of the polling policy.
`polling_location`	The directory that the Recovery Appliance periodically examines for new backups. Do not specify this directory name in multiple polling policies.
`polling_frequency`	The frequency with which the Recovery Appliance examines the specified directory for new backups. System load may cause backup polling to occur less frequently. Specify the window as any valid `INTERVAL DAY TO SECOND` expression, such as `INTERVAL '2' DAY` (2 days), `INTERVAL '4' HOUR` (4 hours), and so on.
`delete_input`	The setting that controls deletion behavior. If `TRUE`, then the Recovery Appliance deletes files in the specified directory after copying them to a storage location. If `FALSE`, then the Recovery Appliance does not delete files that it discovers in the polling location.

CREATE_PROTECTION_POLICY

This procedure creates a protection policy.

Syntax

```
PROCEDURE create_protection_policy (
   protection_policy_name IN VARCHAR2,
   description IN VARCHAR2 DEFAULT NULL,
   storage_location_name IN VARCHAR2,
   polling_policy_name IN VARCHAR2 DEFAULT NULL,
   recovery_window_goal IN DSINTERVAL_UNCONSTRAINED,
   max_retention_window IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
   recovery_window_sbt IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
```

```
  unprotected_window IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
  guaranteed_copy IN VARCHAR2 DEFAULT 'NO',
  allow_backup_deletion IN VARCHAR2 DEFAULT 'YES',
  store_and_forward IN VARCHAR2 DEFAULT 'NO');
```

Parameters

Table 12-8 CREATE_PROTECTION_POLICY Parameters

Parameter	Description
`protection_policy_name`	The user-assigned name of the protection policy.
`description`	An optional description of the usage for the policy.
`storage_location_name`	The name of the storage location. The Recovery Appliance uses this location for actively received incoming backups, and for newly created backup files for all databases sharing this protection policy.
`polling_policy_name`	The name of the backup polling policy. The polling policy specifies the rules for how the Recovery Appliance polls for backups of protected databases that use this protection policy. If null, then no backup polling occurs for databases that use this protection policy.
`recovery_window_goal`	The recovery window goal for databases that use this protection policy. For each protected database, the Recovery Appliance attempts to ensure that the oldest backup on disk can support a point-in-time recovery to any time within the specified interval, counting backward from the current time. Specify the goal as any valid `INTERVAL DAY TO SECOND` expression, such as `INTERVAL '2' DAY` (2 days), `INTERVAL '4' HOUR` (4 hours), and so on.
`max_retention_window`	The maximum length of time that the Recovery Appliance must retain backups for databases that use this protection policy. Recovery Appliance only holds backups longer than the specified period when they are required to preserve the recovery window goal for a database. If null, then the Recovery Appliance does not purge backups unless caused by explicit user actions or space pressures within a storage location.
`recovery_window_sbt`	The recovery window for SBT backups of databases that use this protection policy. For each protected database, the Recovery Appliance keeps backups long enough on tape to guarantee that a recovery is possible to any time within the specified interval, counting backward from the current time. If this parameter is not null, then you must also create an SBT job for this protection policy, and then schedule it using a scheduling facility such as Oracle Scheduler. See CREATE_SBT_JOB_TEMPLATE. Specify the window as any valid `INTERVAL DAY TO SECOND` expression, such as `INTERVAL '2' DAY` (2 days), `INTERVAL '4' HOUR` (4 hours), and so on.

Table 12-8 (Cont.) CREATE_PROTECTION_POLICY Parameters

Parameter	Description
unprotected_window	The maximum amount of data loss that is tolerable for databases using this protection policy. When a protected database exceeds the specified amount of data loss, the Recovery Appliance posts a warning to RA_INCIDENT_LOG. The most recent time to which each protected database is recoverable is shown in the HIGH_TIME column of RA_RESTORE_RANGE. Specify the window as any valid INTERVAL DAY TO SECOND expression, such as INTERVAL '2' DAY (2 days), INTERVAL '4' HOUR (4 hours), and so on.
guaranteed_copy	The setting of the guaranteed copy feature. Specifying NO means that the Recovery Appliance always accepts new backups, even if it must delete old backups when space is low. This option prioritizes the ability to successfully process the backup currently being received over the ability to restore older backups. Specifying YES ensures that the Recovery Appliance replicates backup data or copies it to tape before removing it from Recovery Appliance storage. This option prioritizes the ability to restore older backups over the ability to successfully process the backup currently being received. If set to YES, then for each protected database the Recovery Appliance can only hold up to disk_reserve_space bytes of backup data that is not yet copied or replicated. If hardware or network errors prevent timely copying or replication, then future attempts to create new backups will fail when the Recovery Appliance reaches the disk_reserve_space limit.
allow_backup_deletion	Setting this to NO will prevent RMAN users from deleting backups on the Recovery Appliance. The default value is set to YES. NO means that the Recovery Appliance will prevent backups from being deleted by RMAN users for the databases using this protection policy. YES means that the Recovery Appliance will allow for backups to be deleted by RMAN users for the databases using this protection policy.
store_and_forward	The setting of the Backup and Redo Failover feature. This setting is used only in a protection policy defined on the alternate Recovery Appliance where the protected databases associated with this policy will redirect backups and redo in the event of an outage on the primary Recovery Appliance. Specifying YES means that the alterate Recovery Appliance does not index these redirected backups. Instead, the backups are stored as-is, and are sent to the primary Recovery Appliance when the outage is over. The backup pieces are deleted once they are replicated on the primary; support for incremental forever is turned off for this alternate ZDLRA only. The downstream ZDLRA resumes the incremental forever strategy once it receives these backups. The default is NO.

CREATE_REPLICATION_SERVER

This procedure defines a configuration for a downstream Recovery Appliance that forms part of a Recovery Appliance replication scheme.

This procedure creates metadata for the downstream Recovery Appliance, but does not replicate any backups. Use the ADD_REPLICATION_SERVER procedure to link the downstream Recovery Appliance to one or more protection policies, so that the Recovery Appliance sends backups for protected databases assigned to these policies to the downstream Recovery Appliance.

Syntax

```
PROCEDURE create_replication_server (
   replication_server_name IN VARCHAR2,
   sbt_so_name IN VARCHAR2,
   sbt_parms IN VARCHAR2 DEFAULT NULL,
   max_streams IN NUMBER DEFAULT NULL,
   catalog_user_name IN VARCHAR2,
   wallet_alias IN VARCHAR2,
   wallet_path IN VARCHAR2,
   proxy_url IN VARCHAR2 DEFAULT NULL,
   proxy_port IN NUMBER DEFAULT NULL,
   http_timeout IN NUMBER DEFAULT NULL);
```

Parameters

Table 12-9 CREATE_REPLICATION_SERVER Parameters

Parameter	Description
`replication_server_name`	The user-assigned name of the downstream Recovery Appliance. This value is converted to upper-case before storing.
`sbt_so_name`	The name and path to the Recovery Appliance Backup Module. The module is an Oracle-supplied media library that simulates an SBT device. The Recovery Appliance uses this library to communicate with the downstream Recovery Appliance.

`sbt_parms`	The name and path of a client configuration file in the form `(RA_CLIENT_CONFIG_FILE=`*file_system_location*`)`. The parentheses are mandatory. The client configuration file is a text file that must contain at a minimum the following parameters:
	`ra_host` - The host name and port of the remote Recovery Appliance in the format *host*:*port*.
	`ra_wallet` - The location of an Oracle wallet to use for authentication, and the credential to use.
	The following shows the sample contents of a client configuration file:
	`ra_host=oam2.example.com:6498`
	`ra_wallet='location=file:/u01/oracle/wallets credential_alias=repcred1'`

Table 12-9 (Cont.) CREATE_REPLICATION_SERVER Parameters

Parameter	Description
max_streams	The maximum number of simultaneous replication tasks. If null, which is the recommended setting, then the upstream Recovery Appliance determines the number of streams to use for replication based on the number of its nodes.
catalog_user_name	The name of the replication user account on the downstream Recovery Appliance. The replication user account is a database user account on the downstream Recovery Appliance that upstream Recovery Appliances use to authenticate with this downstream Recovery Appliance.
wallet_alias	The alias that identifies the credential within the wallet that the upstream Recovery Appliances uses to authenticate with the downstream Recovery Appliance.
wallet_path	The path to the local Oracle wallet (excluding the wallet file name). Path must start with `file:` .
proxy_url	The URL of any required proxy server, in the format *host*.
proxy_port	The port number of the proxy server.
http_timeout	The HTTP timeout interval, in seconds. Usually you leave this parameter set to null, to accept the system default HTTP timeout, unless directed to set it to a different value by Oracle Support.

CREATE_SBT_ATTRIBUTE_SET

This procedure creates an SBT attribute set that SBT jobs can use.

An SBT attribute set provides a grouping of attributes that control the execution of an SBT job. These attributes enable you to specify settings for the media management library, including destination media pool or media family. You can define multiple SBT attribute sets. Multiple jobs can reference a single attribute set.

Syntax

```
PROCEDURE create_sbt_attribute_set(
   lib_name IN VARCHAR2,
   attribute_set_name IN VARCHAR2,
   streams IN NUMBER DEFAULT NULL,
   poolid IN NUMBER DEFAULT NULL,
   parms IN VARCHAR2 DEFAULT NULL,
   send IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-10 CREATE_SBT_ATTRIBUTE_SET Parameters

Parameter	Description
lib_name	The name of the SBT library to associate with the attribute set.

Table 12-10 (Cont.) CREATE_SBT_ATTRIBUTE_SET Parameters

Parameter	Description
attribute_set_name	User-assigned name of the attribute set. Attribute set names must be unique.
streams	The maximum number of concurrent streams that the Recovery Appliance uses for automated backups. The number of concurrent streams never exceeds the limits set by the drives and restore_drives attributes of the SBT library. If streams is null, then the Recovery Appliance uses all available drives.
poolid	The media pool number to use as the destination for backup copies. This parameter accepts a value in the same format as the POOL parameter of the RMAN BACKUP command.
parms	The media management library-specific parameter string for the backup copy operation. The string has the same format as the PARMS option of the RMAN ALLOCATE CHANNEL command. During SBT backup operations for this attribute, the Recovery Appliance merges the value of this parameter with the PARMS parameter specified in the CREATE_SBT_LIBRARY procedure.
send	The string that the Recovery Appliance uses to send additional media management library-specific parameters for the backup copy operation. The string has the same format as the SEND option of the RMAN ALLOCATE CHANNEL command. During backup operations for this attribute, the Recovery Appliance merges the value of this parameter with the SEND parameter specified in the CREATE_SBT_LIBRARY procedure.

CREATE_SBT_JOB_TEMPLATE

This procedure creates an SBT job that describes how the Recovery Appliance chooses backups for copying to tape. This form of this overloaded procedure applies to backups for all protected databases assigned to the specified protection policy.

After you create an SBT backup job, you must schedule it with a scheduling facility such as Oracle Scheduler. See QUEUE_SBT_BACKUP_TASK.

Syntax

```
PROCEDURE create_sbt_job_template (
   template_name IN VARCHAR2,
   protection_policy_name IN VARCHAR2,
   attribute_set_name IN VARCHAR2,
   backup_type IN VARCHAR2,
   full_template_name IN VARCHAR2 DEFAULT NULL,
   from_tag IN VARCHAR2 DEFAULT NULL,
   priority IN NUMBER DEFAULT SBT_PRIORITY_MEDIUM,
   copies IN NUMBER DEFAULT 1,
   window IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-11 CREATE_SBT_JOB_TEMPLATE Parameters

Parameter	Description
`template_name`	The user-assigned name of this SBT job template.
`protection_policy_name`	The name of the protection policy to which this SBT job applies. Backups for all protected databases assigned to this protection policy are eligible for copying.
`attribute_set_name`	The name of the SBT attribute set to use for this SBT job.
`backup_type`	The types of backups that this SBT job chooses for copying to tape. The string must be a comma-separated list of the following types: `ALL`: Shorthand for `FULL`, `INCR`, `ARCH` `INCR`: Copies all incremental logs that have not yet been copied to tape, since the most recent full backup. `ARCH`: Copies all archived redo log backups that have not yet been copied to tape, since the most recent full backup. `FULL`: Copies the most recent virtual level 0 backup, if it has not already been copied, to tape. The backup can either be a virtual level 0 backup that is based on the most recent level 0 backup received or a virtual level 0 backup that is based on the most recent level 1 backup received, whichever is more recent.
`full_template_name`	The full name of this SBT job template. This applies only to `INCR` and `ARCH` backup types. The full name links full backups with the incremental backups and archived redo log files needed to recover them. If only one full backup template exists for the specified tape library, then this parameter defaults to the name of this template, which means it does not need to be specified. If multiple full backup templates exist, then you must specify the full template name. The specified `FULL` template name must belong to the same SBT library as the `INCR` or `ARCH` job. If `backup_type` is set to `FULL` or `ALL`, then `full_template_name` is the same as `template_name`.
`from_tag`	The tag name. If specified, then the Recovery Appliance only considers backups using this tag for copying to tape. Refer to *"Oracle Database Backup and Recovery Reference"* for the correct format for TAG string.
`priority`	The priority of this job for tape resource usage. Lower priority values take precedence over higher values. `0` is the highest possible priority. You can use any number that is greater than or equal to `0`. The pre-defined values are as follows: `SBT_PRIORITY_LOW` maps to 1000 `SBT_PRIORITY_MEDIUM` maps to 100 `SBT_PRIORITY_HIGH` maps to 10 `SBT_PRIORITY_CRITICAL` maps to 1 The default priority is `SBT_PRIORITY_MEDIUM`. Restore jobs by default have `SBT_PRIORITY_CRITICAL` priority.

Table 12-11 (Cont.) CREATE_SBT_JOB_TEMPLATE Parameters

Parameter	Description
copies	The number of distinct copies of each backup that this SBT job creates. Valid values range from 1 (default) to 4.
window	The window of time in which this job can copy backups to tape. Copy tasks that are not able to start within the specified window must wait until the next scheduled job execution.
compression_algorithm	Specifies the compression algorithm. If compression_algorithm is specified, it will override the compression algorithm defined in template_name for this single operation. If template_name is NULL, it defines the compression algorithm for this operation. BASIC: Good compression ratios with potentially lower speed than MEDIUM. LOW: Optimized for speed with potentially lower compression raios than BASIC. MEDIUM: Recommended for most environments. Good combination of compression ratios and speed. HIGH: Best suited for operations over slower networks where the limiting factor is maximum network throughput. OFF: No compression. NULL: (default) indicates that the algorithm defined in the SBT job template should be used. 'LOW', 'MEDIUM', and 'HIGH' are Advanced Compression Options (ACO) which require additional licensing on the protected database for restore operations. OSB licenses are needed to restore backups directly from tape to DB client. OSB licenses include compression, so no ACO is needed. If 3P MMLs are used with compression, then ACO is needed on the target DB.
encryption_algorithm	Encryption algorithm to use for tape jobs. Valid value are 'AES128', 'AES192', 'AES256', 'OFF', or the constant equivalents ENC_OFF, ENC_AES128, ENC_AES192, ENC_AES256

CREATE_SBT_JOB_TEMPLATE

This procedure creates a new SBT backup job. The job describes how the Recovery Appliance chooses backups for copying to tape/cloud. This form of this overloaded procedure applies to backups for a single protected database only, whereas the previous form applies to backups of all databases assigned to a specific protection policy. With the exception of this difference, this procedure and its parameters are identical to the alternative form of this procedure.

Syntax

```
PROCEDURE create_sbt_job_template (
   template_name IN VARCHAR2,
   db_unique_name IN VARCHAR2,
   attribute_set_name IN VARCHAR2,
```

```
   backup_type IN VARCHAR2,
   full_template_name IN VARCHAR2 DEFAULT NULL,
   from_tag IN VARCHAR2 DEFAULT NULL,
   priority IN NUMBER DEFAULT SBT_PRIORITY_MEDIUM,
   copies IN NUMBER DEFAULT 1,
   window IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-12 CREATE_SBT_JOB_TEMPLATE Parameters

Parameter	Description
db_unique_name	The DB_UNIQUE_NAME of the protected database to which this SBT job applies. This SBT job copies only backups that belong to the specified database.

CREATE_SBT_LIBRARY

This procedure creates metadata describing an installed media management software library. The Recovery Appliance uses the specified library to copy backups from internal storage either to tape or to other tertiary storage supported by this media manager.

Syntax

```
PROCEDURE create_sbt_library (
   lib_name IN VARCHAR2,
   drives IN NUMBER,
   restore_drives IN NUMBER DEFAULT 0,
   parms IN VARCHAR2 DEFAULT NULL,
   send IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-13 CREATE_SBT_LIBRARY Parameters

Parameter	Description
lib_name	The user-specified name that the Recovery Appliance uses to refer to this SBT library.
drives	The maximum number of tape drives that this SBT library can access. The Recovery Appliance never uses more than the specified number of concurrent streams when accessing this library.

Table 12-13 (Cont.) CREATE_SBT_LIBRARY Parameters

Parameter	Description
restore_drives	The number of tape drives that the Recovery Appliance reserves for restore operations. If specified, then the Recovery Appliance uses a maximum of drives - restore_drives drives for backup operations, which ensures that the Recovery Appliance always has the specified number of drives available for restore operations. If not specified, then the Recovery Appliance can use all available drives for backups, which means that a restore operation might have to wait for a drive to become free.
parms	The library-specific parameter string that the Recovery Appliance uses to access this SBT library. This string has the same format as the PARMS option of the RMAN ALLOCATE CHANNEL command. The string usually contains the SBT_LIBRARY parameter.
send	The parameter string that the Recovery Appliance uses to send additional library-specific parameters to this SBT library. This string has the same format as the SEND option of the RMAN ALLOCATE CHANNEL command.

CREATE_STORAGE_LOCATION

This procedure creates Recovery Appliance storage location, which is an object that describes storage to be used by the Recovery Appliance. Recovery Appliance stores all backups in named storage locations.

The Recovery Appliance allocates storage locations from the Oracle ASM disk groups that reside in the Recovery Appliance.

The newly created storage location does not store any files until you use a protection policy to associate the location with a protected database.

Syntax

```
PROCEDURE create_storage_location (
   storage_location_name IN VARCHAR2,
   storage_location_dests IN VARCHAR2);
```

Parameters

Table 12-14 CREATE_STORAGE_LOCATION Parameters

Parameter	Description
storage_location_name	The user-assigned name of this storage location.

Table 12-14 (Cont.) CREATE_STORAGE_LOCATION Parameters

Parameter	Description
storage_location_dests	A comma-delimited list of Oracle ASM disk groups from which space is to be allocated for this storage location. The disk groups must exist. The specified disk groups must have the same allocation unit size. Multiple storage locations cannot use the same Oracle ASM disk group. This parameter accepts a string in the following format: *storage_location_spec* [, *storage_location_spec* ...] Each *storage_location_spec* has the following format: *storage_location_name* [(*size* [K | M | G | T | P | E | Z | Y])] *storage_location_name* is the name of an Oracle ASM disk group. Optionally, you can follow the disk group name with the amount of space to be allocated from this disk group for this storage location. If no size is specified, then the Recovery Appliance allocates all of the free space in the disk group to this storage location. The following example specifies a storage location that uses all the free space in disk group DG1: +DG1 The following example specifies a storage location that uses 20 terabytes of space from disk group DG1, all free space in disk group DG2, and 30 terabytes of space from disk group DG3: +DG1(20T),+DG2,+DG3(30T)

DELETE_DB

This procedure removes the specified protected database from the Recovery Appliance. The Recovery Appliance deletes all metadata and backups associated with this database, both from disk and SBT. Backups on tape are not affected.

If the Recovery Appliance cannot delete the SBT backups owned by this database because of errors, then the DELETE_DB operation fails. If SBT errors occur, then the specified database is not completely removed from the Recovery Appliance. The Recovery Appliance logs errors that occur during the DELETE_DB procedure in the RA_INCIDENT_LOG view. If the wait parameter is specified as TRUE, then the Recovery Appliance also raises these errors in the session in which DELETE_DB is called. If you diagnose the SBT errors and fix the problem, then you can run DELETE_DB again.

Syntax

```
PROCEDURE delete_db (
   db_unique_name IN VARCHAR2,
   wait IN BOOLEAN DEFAULT TRUE);
```

Parameters

Table 12-15 DELETE_DB Parameters

Parameter	Description
db_unique_name	The DB_UNIQUE_NAME of the database to be removed.
wait	The wait behavior of the procedure. If TRUE, then the procedure will not return until the backups and metadata for the specified database are completely removed from the Recovery Appliance. If FALSE, then the procedure returns immediately, and the database deletion operation continues in the background.

DELETE_POLLING_POLICY

This procedure deletes the specified backup polling policy.

Syntax

```
PROCEDURE delete_polling_policy (
   polling_policy_name IN VARCHAR2);
```

Parameters

Table 12-16 DELETE_POLLING_POLICY Parameters

Parameter	Description
polling_policy_name	The name of the backup polling policy to delete.

DELETE_PROTECTION_POLICY

This procedure deletes the specified protection policy.

The specified policy must not be associated with any database.

Syntax

```
PROCEDURE delete_protection_policy (
   protection_policy_name IN VARCHAR2);
```

Parameters

Table 12-17 DELETE_PROTECTION_POLICY Parameters

Parameter	Description
protection_policy_name	The name of the protection policy to delete.

DELETE_REPLICATION_SERVER

This procedure deletes a replication server configuration. The Recovery Appliance removes all metadata relating to the downstream Recovery Appliance.

Syntax

```
PROCEDURE delete_replication_server (
   replication_server_name IN VARCHAR2,
   force IN BOOLEAN DEFAULT FALSE);
```

Parameters

Table 12-18 DELETE_REPLICATION_SERVER Parameters

Parameter	Description
`replication_server_name`	The name of the replication server configuration to delete.
`force`	The deletion behavior when a protection policy is associated with the configuration. If FALSE, and if the replication server configuration is still associated with a protection policy, then the deletion fails. In this case, you must first call REMOVE_REPLICATION_SERVER. If TRUE, then `delete_replication_server` first removes the replication server configuration from the protection policy.

DELETE_SBT_ATTRIBUTE_SET

This procedure deletes the specified SBT attribute set.

Syntax

```
PROCEDURE delete_sbt_attribute_set(
   attribute_set_name IN VARCHAR2);
```

Parameters

Table 12-19 DELETE_SBT_ATTRIBUTE_SET Parameters

Parameter	Description
`attribute_set_name`	The name of the SBT attribute set to delete.

DELETE_SBT_JOB_TEMPLATE

This procedure deletes the specified SBT job template.

Syntax

```
PROCEDURE delete_sbt_job_template (
   template_name IN VARCHAR2);
```

Parameters

Table 12-20 DELETE_SBT_JOB_TEMPLATE Parameters

Parameter	Description
template_name	The name of the SBT job to delete. The Recovery Appliance removes tasks belonging to this job from the task queue but does not terminate any executing task.

DELETE_SBT_LIBRARY

This procedure deletes the metadata describing the specified SBT library.

The Recovery Appliance only removes the SBT library object, and does not uninstall the media management software.

This procedure deletes any SBT jobs and attributes created for this SBT library.

Syntax

```
PROCEDURE delete_sbt_library (
   lib_name IN VARCHAR2);
```

Parameters

Table 12-21 DELETE_SBT_LIBRARY Parameters

Parameter	Description
lib_name	The name of the SBT library to delete.

DELETE_STORAGE_LOCATION

This procedure deletes the specified Recovery Appliance storage location.

The specified storage location must be empty and must not be associated with any protection policy. No storage movement may be in progress for this storage location. If any storage movement is in progress, or if any files exist in this storage location, then this procedure fails with an error.

Syntax

```
PROCEDURE delete_storage_location (
   storage_location_name IN VARCHAR2);
```

Parameters

Table 12-22 DELETE_STORAGE_LOCATION Parameters

Parameter	Description
storage_location_name	The name of the Recovery Appliance storage location to delete.

ESTIMATE_SPACE

This procedure estimates the amount of storage in GB required for recovery of a given database and a desired recovery window.

Syntax

```
FUNCTION estimate_space (
   db_unique_name IN VARCHAR2,
   target_window IN DSINTERVAL_UNCONSTRAINED) RETURN NUMBER;
```

Parameters

Table 12-23 ESTIMATE_SPACE Parameters

Parameter	Description
db_unique_name	The name of the database needing the storage estimate.
target_window	The desired recovery window for the database. Specify the goal as any valid INTERVAL DAY TO SECOND expression, such as INTERVAL '2' DAY (2 days), INTERVAL '4' HOUR (4 hours), and so on.

GRANT_DB_ACCESS

This procedure grants the necessary privileges to the specified recovery Appliance user account to enable this account to back up, restore, and access recovery catalog metadata for the specified protected database.

Syntax

```
PROCEDURE grant_db_access (
   username IN VARCHAR2,
   db_unique_name IN VARCHAR2);
```

Parameters

Table 12-24 GRANT_DB_ACCESS Parameters

Parameter	Description
username	The name of the Recovery Appliance user account.

Table 12-24 (Cont.) GRANT_DB_ACCESS Parameters

Parameter	Description
db_unique_name	The protected database for which the privilege is being granted.

KEY_REKEY

This procedure rekeys encryption keys for all databases with existing encryption keys.

Syntax

```
PROCEDURE key_rekey;
```

KEY_REKEY

This procedure rekeys encryption keys for the specified database with an existing encryption key.

Syntax

```
PROCEDURE key_rekey (
   db_unique_name IN VARCHAR2);
```

Parameters

Table 12-25 KEY_REKEY Parameters

Parameter	Description
db_unique_name	The DB_UNIQUE_NAME of the database to generate a new encryption key. Note: this routine will not create a new key, only rekey an existing key

KEY_REKEY

This procedure rekeys encryption keys for all databases with existing encryption keys in the specified protection_policy

Syntax

```
PROCEDURE key_rekey (
   protection_policy_name IN VARCHAR2);
```

Parameters

Table 12-26 KEY_REKEY Parameters

Parameter	Description
protection_policy_name	Generate new encryption keys for databases that are part of this protection policy.

MIGRATE_TAPE_BACKUP

This procedure makes pre-migration tape backups accessible to the Recovery Appliance through the specified SBT library. You must first import metadata about the tape backups into the Recovery Appliance catalog using the RMAN IMPORT CATALOG command.

This procedure performs the metadata adjustments required to access pre-existing tape backups, but does not physically move backups. The pre-existing backups must already be accessible by the specified SBT library.

Syntax

```
PROCEDURE migrate_tape_backup(
   db_unique_name IN VARCHAR2,
   sbt_lib_name IN VARCHAR2);
```

Parameters

Table 12-27 MIGRATE_TAPE_BACKUP Parameters

Parameter	Description
db_unique_name	The comma-delimited list of protected databases whose backups are to be migrated. You must already have registered each db_unique_name with the Recovery Appliance catalog, and have added it to the Recovery Appliance with ADD_DB.
sbt_lib_name	The SBT library that the Recovery Appliance uses to access existing tape backups for the specified protected database. See CREATE_SBT_LIBRARY.

MOVE_BACKUP

This procedure moves one or more long-term archival backup pieces from the Recovery Appliance to a user-specified disk or SBT destination.

The Recovery Appliance copies all backup pieces matching the specified tag to the location specified with the format and template_name parameters. After the Recovery Appliance copies each backup piece successfully, the Recovery Appliance deletes the backup piece from its original location.

Syntax

```
PROCEDURE move_backup (
   tag IN VARCHAR2,
   format IN VARCHAR2,
   template_name IN VARCHAR2,
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-28 MOVE_BACKUP Parameters

Parameter	Description
`tag`	The tag of backups to copy. The Recovery Appliance removes all backups matching this tag.
`format`	The naming format of the backup pieces to create. This parameter follows the same rules as the RMAN `FORMAT` parameter.
`template_name`	A media management library template. If null, then `format` points to a disk location. If not null, then the Recovery Appliance copies the backup piece to tape, using the media pool referenced in the SBT template name as the copy destination.
`compression_algorithm`	Specifies the compression algorithm If `compression_algorithm` is specified, it will override the compression algorithm defined in `template_name` for this single operation. If `template_name` is NULL, it defines the compression algorithm for this operation. Valid value are 'BASIC', 'LOW', 'MEDIUM', 'HIGH', 'OFF'
`encryption_algorithm`	Specifies the encryption algorithm If `encryption_algorithm` is specified, it will override the encryption algorithm defined in `template_name` for this single operation. If `template_name` is NULL, it defines the encryption algorithm for this operation. Valid value are 'AES128', 'AES192', 'AES256', 'OFF' or the constant equivalents ENC_OFF, ENC_AES128, ENC_AES192, ENC_AES256

MOVE_BACKUP_PIECE

This procedure moves a single long-term archival backup piece from the Recovery Appliance to a user-specified disk or SBT destination.

The Recovery Appliance copies the specified backup piece to the location specified with the `format` and `template_name` parameters. After the Recovery Appliance copies the backup piece successfully, the Recovery Appliance deletes the backup piece from its original location.

Syntax

```
PROCEDURE move_backup_piece (
   bp_key IN NUMBER,
   format IN VARCHAR2,
   template_name IN VARCHAR2,
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-29 MOVE_BACKUP_PIECE Parameters

Parameter	Description
bp_key	The unique key of the backup piece to move. Obtain this key from the RC_BACKUP_PIECE view.
format	The naming format of the backup pieces to create. This parameter follows the same rules as the RMAN FORMAT parameter.
template_name	A media management library template. If null, then format points to a disk location. If not null, then the Recovery Appliance copies the backup piece to tape, using the media pool referenced in the SBT template name as the copy destination.
compression_algorithm	Specifies the compression algorithm. If compression_algorithm is specified, it will override the compression algorithm defined in template_name for this single operation. If template_name is NULL, it defines the compression algorithm for this operation. BASIC: Good compression ratios with potentially lower speed than MEDIUM. LOW: Optimized for speed with potentially lower compression raios than BASIC. MEDIUM: Recommended for most environments. Good combination of compression ratios and speed. HIGH: Best suited for operations over slower networks where the limiting factor is maximum network throughput. OFF: No compression. NULL: (default) indicates that the algorithm defined in the SBT job template should be used. 'LOW', 'MEDIUM', and 'HIGH' are Advanced Compression Options (ACO) which require additional licensing on the protected database for restore operations. OSB licenses are needed to restore backups directly from tape to DB client. OSB licenses include compression, so no ACO is needed. If 3P MMLs are used with compression, then ACO is needed on the target DB.

Table 12-29 (Cont.) MOVE_BACKUP_PIECE Parameters

Parameter	Description
encryption_algorithm	Specifies the encryption algorithm If encryption_algorithm is specified, it will override the encryption algorithm defined in template_name for this single operation. If template_name is NULL, it defines the encryption algorithm for this operation. Valid value are 'AES128', 'AES192', 'AES256', 'OFF' or the constant equivalents ENC_OFF, ENC_AES128, ENC_AES192, ENC_AES256

PAUSE_REPLICATION_SERVER

This procedure pauses replication to the specified downstream Recovery Appliance.

The Recovery Appliance permits in-progress replication of backup pieces to complete. If the Recovery Appliance queued backup pieces for replication through this replication server configuration but did not replicate them, then the Recovery Appliance holds the backup pieces until you call RESUME_REPLICATION_SERVER. No replication tasks that run against this Recovery Appliance can execute until you resume replication to the downstream Recovery Appliance.

Syntax

```
PROCEDURE pause_replication_server  (
   replication_server_name IN VARCHAR2);
```

Parameters

Table 12-30 PAUSE_REPLICATION_SERVER Parameters

Parameter	Description
replication_server_name	The name of the downstream Recovery Appliance.

PAUSE_SBT_LIBRARY

This procedure pauses the specified SBT library. The Recovery Appliance allows in-progress copies of backup pieces to complete. However, if backup pieces were queued for copy through this SBT library but not yet copied, then the Recovery Appliance holds them until you resume the SBT library. No new SBT jobs that run against this library can execute until you resume the library (RESUME_SBT_LIBRARY).

Query the RA_SBT_LIBRARY view for a list of existing SBT libraries.

Syntax

```
PROCEDURE pause_sbt_library(
   lib_name IN VARCHAR2);
```

Parameters

Table 12-31 PAUSE_SBT_LIBRARY Parameters

Parameter	Description
lib_name	Name of the SBT library to pause.

POPULATE_BACKUP_PIECE

This procedure pushes the specified backup piece into the delta store.

Use this procedure to initially populate the delta store or to correct corruption in the delta store. The delta store is backup data that supports an incremental-forever backup solution. Only incremental backups (not KEEP backups) can become part of the delta store.

Syntax

```
PROCEDURE populate_backup_piece(
   backup_piece_key IN NUMBER);
```

Parameters

Table 12-32 POPULATE_BACKUP_PIECE Parameters

Parameter	Description
backup_piece_key	Either the backup piece key provided by the Recovery Appliance when it detected a corruption, or the backup piece to insert into the delta store. If the key represents a virtual backup piece, then the Recovery Appliance searches for a backup piece to resolve corruption in the delta store. If the key does not represent a virtual backup, then the Recovery Appliance inserts this backup piece into the delta store. This backup must be an incremental backup that is not a KEEP backup.

QUEUE_SBT_BACKUP_TASK

This procedure queues the backup pieces selected by the specified SBT job template for copying to tape. Typically, a scheduling utility such as Oracle Scheduler calls this procedure.

Syntax

```
PROCEDURE queue_sbt_backup_task(
   template_name IN VARCHAR2,
   format IN VARCHAR2 DEFAULT NULL,
   autobackup_prefix IN VARCHAR2 DEFAULT NULL,
   tag IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-33 QUEUE_SBT_BACKUP_TASK Parameters

Parameter	Description
`template_name`	The name of the SBT job template that specifies the backup pieces to copy to tape.
`format`	The naming format of the backup pieces to create. This parameter follows the same rules as the RMAN `FORMAT` parameter
`autobackup_prefix`	The original autobackup names will be prefixed with this autobackup_prefix.
`tag`	User specified tag for backups to be copied See CREATE_SBT_JOB_TEMPLATE.

REMOVE_REPLICATION_SERVER

This procedure removes the specified replication server configuration from the specified protection policy. After the operation succeeds, the Recovery Appliance no longer replicates backups of databases protected by this policy to the downstream Recovery Appliance.

Syntax

```
PROCEDURE remove_replication_server (
   replication_server_name IN VARCHAR2,
   protection_policy_name IN VARCHAR2);
```

Parameters

Table 12-34 REMOVE_REPLICATION_SERVER Parameters

Parameter	Description
`replication_server_name`	The name of the replication server configuration to remove.
`protection_policy_name`	The name of the protection policy from which the specified replication server configuration is to be removed.

RENAME_DB

This procedure changes the name of the specified protected database in the Recovery Appliance metadata.

Use this procedure when the `DB_UNIQUE_NAME` for a protected database changes, so that the Recovery Appliance metadata reflects the correct name.

Syntax

```
PROCEDURE rename_db (
```

```
   db_unique_name_old IN VARCHAR2,
   db_unique_name_new IN VARCHAR2);
```

Parameters

Table 12-35 RENAME_DB Parameters

Parameter	Description
db_unique_name_old	The DB_UNIQUE_NAME to change.
db_unique_name_new	The new DB_UNIQUE_NAME.

RESET_ERROR

This procedure modifies the specified error log entry to have the status RESET. Errors marked in this fashion do not cause Oracle Enterprise Manager to raise alerts. If the Recovery Appliance determines that the problem is still occurring, then errors that have been reset change to ACTIVE status. The primary use of this API is to reset the error status for nonrecurring errors, such as transient media failures.

Syntax

```
PROCEDURE reset_error(
   incident# IN NUMBER);
```

Parameters

Table 12-36 RESET_ERROR Parameters

Parameter	Description
incident#	The unique identifier of the error log entry to reset. Obtain the identifier from the RA_INCIDENT_LOG view.

RESUME_REPLICATION_SERVER

This procedure resumes replication to the specified downstream Recovery Appliance, after a previous call to PAUSE_REPLICATION_SERVER.

Syntax

```
PROCEDURE resume_replication_server (
   replication_server_name IN VARCHAR2);
```

Parameters

Table 12-37 RESUME_REPLICATION_SERVER Parameters

Parameter	Description
replication_server_name	The name of the downstream Recovery Appliance.

RESUME_SBT_LIBRARY

This procedure resumes a paused SBT library.

Query the RA_SBT_LIBRARY to determine which SBT libraries are paused (see PAUSE_SBT_LIBRARY).

Syntax

```
PROCEDURE resume_sbt_library(
   lib_name IN VARCHAR2);
```

Parameters

Table 12-38 RESUME_SBT_LIBRARY Parameters

Parameter	Description
lib_name	Name of the SBT library to resume.

REVOKE_DB_ACCESS

This procedure revokes privileges on one protected database from the specified Recovery Appliance user account.

Syntax

```
PROCEDURE revoke_db_access (
   username IN VARCHAR2,
   db_unique_name IN VARCHAR2);
```

Parameters

Table 12-39 REVOKE_DB_ACCESS Parameters

Parameter	Description
username	The name of the user account from which to revoke the privilege.
db_unique_name	The protected database for which the privilege is being revoked.

SET_SYSTEM_DESCRIPTION

This procedure sets a descriptive name for users to apply to their Recovery Appliance. The name provided here will be seen in the RA_SERVER view.

Syntax

```
PROCEDURE set_system_description(
   sys_desc VARCHAR2);
```

Parameters

Table 12-40 SET_SYSTEM_DESCRIPTION Parameters

Parameter	Description
sys_desc	A descriptive name for this Recovery Applicance.

SHUTDOWN

Synonymous with SHUTDOWN_RECOVERY_APPLIANCE.

Syntax

```
PROCEDURE shutdown;
```

SHUTDOWN_RECOVERY_APPLIANCE

This procedure performs a clean shutdown of the Recovery Appliance.

This procedure permits in-progress operations to complete before shutting down. The shutdown can take some time. If an immediate shutdown is required, then use ABORT_RECOVERY_APPLIANCE.

Syntax

```
PROCEDURE shutdown_recovery_appliance;
```

STARTUP

Synonymous with STARTUP_RECOVERY_APPLIANCE.

Syntax

```
PROCEDURE startup;
```

STARTUP_RECOVERY_APPLIANCE

This procedure starts the Recovery Appliance after it has been shut down or terminated.

The Recovery Appliance can process backup and restore requests only when it is started.

If the Recovery Appliance was started with STARTUP_RECOVERY_APPLIANCE, and if any instance of the Recovery Appliance metadata database is restarted, then a database startup trigger automatically restarts the Recovery Appliance. The only exception is when the metadata database is restarted with the RESETLOGS option, which requires you to run the startup_recovery_appliance procedure to repair corrupt metadata.

Syntax

```
PROCEDURE startup_recovery_appliance;
```

UPDATE_DB

This procedure changes the attributes that are assigned to the specified protected database.

Syntax

```
PROCEDURE update_db (
   db_unique_name IN VARCHAR2,
   protection_policy_name IN VARCHAR2 DEFAULT NULL,
   reserved_space IN VARCHAR2 DEFAULT NULL,
   db_timezone IN VARCHAR2 DEFAULT NULL,
   incarnations IN VARCHAR2 DEFAULT 'CURRENT');
```

Parameters

Table 12-41 UPDATE_DB Parameters

Parameter	Description
db_unique_name	The DB_UNIQUE_NAME of the database.
protection_policy_name	The name of the protection policy to assign to the database. The protection policy must exist. The new protection policy controls new storage operations. If the old and new protection policies specify different storage locations, the Recovery Appliance starts a background task to move data from the old storage location to the new location. Recovery Appliance only moves backups that are not obsolete. If a move between storage locations is required, then the RA_DATABASE view does not show the new protection policy until the move has started. If the Recovery Appliance must perform higher priority work, then the Recovery Appliance may not start the move for several hours.
reserved_space	See ADD_DB.
db_timezone	The time zone where this database is located. By default, protected databases are assigned to the same time zone as the Recovery Appliance. If the protected database is in a different time zone, then use this procedure to assign the database to the correct time zone.

Table 12-41 (Cont.) UPDATE_DB Parameters

Parameter	Description
incarnations	Comma-delimited list of keys for all previous database incarnations to update. By default, this procedure updates the current incarnation. If this list contains the current incarnation key, then the procedure updates the row corresponding to the current incarnation in the dbinc and node tables only if the protected database administrator has not specified the time zone in the parameter file. If their entries in the dbinc table are valid, then this procedure updates the entries for all other incarnations in the list.

UPDATE_POLLING_POLICY

This procedure modifies the parameters for an existing backup polling policy.

Parameters that are NULL retain their existing values.

Syntax

```
PROCEDURE update_polling_policy (
   polling_policy_name IN VARCHAR2,
   polling_location IN VARCHAR2 DEFAULT NULL,
   polling_frequency IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
   delete_input IN BOOLEAN DEFAULT NULL);
```

Parameters

Table 12-42 UPDATE_POLLING_POLICY Parameters

Parameter	Description
polling_policy_name	The name of the backup polling policy to update.
polling_location	See CREATE_POLLING_POLICY.
polling_frequency	See CREATE_POLLING_POLICY.
delete_input	See CREATE_POLLING_POLICY.

UPDATE_PROTECTION_POLICY

This procedure modifies the parameters for an existing protection policy.

If a parameter is NULL, its value remains unchanged, except as noted below.

Syntax

```
PROCEDURE update_protection_policy (
   protection_policy_name IN VARCHAR2,
   description IN VARCHAR2 DEFAULT NULL,
   storage_location_name IN VARCHAR2 DEFAULT NULL,
   polling_policy_name IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p1'),
```

```
   recovery_window_goal IN DSINTERVAL_UNCONSTRAINED DEFAULT NULL,
   max_retention_window IN DSINTERVAL_UNCONSTRAINED DEFAULT
dbms_ra_misc.intervalnull('p3'),
   recovery_window_sbt IN DSINTERVAL_UNCONSTRAINED DEFAULT
dbms_ra_misc.intervalnull('p2'),
   unprotected_window IN DSINTERVAL_UNCONSTRAINED DEFAULT
dbms_ra_misc.intervalnull('p4'),
   guaranteed_copy IN VARCHAR2 DEFAULT NULL,
   allow_backup_deletion IN VARCHAR2 DEFAULT NULL,
   store_and_forward IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-43 UPDATE_PROTECTION_POLICY Parameters

Parameter	Description
protection_policy_name	The name of the protection policy to update.
description	See CREATE_PROTECTION_POLICY.
storage_location_name	See CREATE_PROTECTION_POLICY. If you change the storage location for this protection policy, then the Recovery Appliance starts background jobs to move data from the old storage location to the new storage location.
polling_policy_name	See CREATE_PROTECTION_POLICY. If you do not specify this parameter, then the policy retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
recovery_window_goal	See CREATE_PROTECTION_POLICY.
max_retention_window	See CREATE_PROTECTION_POLICY. If this parameter is not specified, its old value is retained. If specified, including being specified as NULL, the new value is set.
recovery_window_sbt	See CREATE_PROTECTION_POLICY. If you do not specify this parameter, then the policy retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
unprotected_window	If you do not specify this parameter, then the policy retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value. See CREATE_PROTECTION_POLICY.
guaranteed_copy	See CREATE_PROTECTION_POLICY.
allow_backup_deletion	See CREATE_PROTECTION_POLICY.
store_and_forward	See CREATE_PROTECTION_POLICY.

UPDATE_REPLICATION_SERVER

This procedure changes the settings for a replication server configuration.

Note the following restrictions for changing replication server parameters:

The configuration does not retain the `sbt_parms` string from the original CREATE_REPLICATION_SERVER call. If you change any parameter other than `max_streams`, then you must pass in this value.

Changing any setting other than `max_streams` requires replication to be paused, which you can achieve by calling PAUSE_REPLICATION_SERVER.

Parameters other than `sbt_parms` whose values are null remain unchanged, except as noted in the following parameter descriptions.

Syntax

```
PROCEDURE update_replication_server (
   replication_server_name IN VARCHAR2,
   sbt_so_name IN VARCHAR2 DEFAULT NULL,
   sbt_parms IN VARCHAR2 DEFAULT NULL,
   max_streams IN NUMBER DEFAULT dbms_ra_misc.number2null('p4'),
   catalog_user_name IN VARCHAR2 DEFAULT NULL,
   wallet_alias IN VARCHAR2 DEFAULT NULL,
   wallet_path IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p1'),
   proxy_url IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p2'),
   proxy_port IN NUMBER DEFAULT dbms_ra_misc.number2null('p3'),
   http_timeout IN NUMBER DEFAULT NULL);
```

Parameters

Table 12-44 UPDATE_REPLICATION_SERVER Parameters

Parameter	Description
`replication_server_name`	The name of the replication server configuration to update. This value is converted to upper-case before storing.
`sbt_so_name`	See CREATE_REPLICATION_SERVER.
`sbt_parms`	See CREATE_REPLICATION_SERVER.
`max_streams`	See CREATE_REPLICATION_SERVER. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
`catalog_user_name`	See CREATE_REPLICATION_SERVER.
`wallet_alias`	See CREATE_REPLICATION_SERVER.
`wallet_path`	See CREATE_REPLICATION_SERVER. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value. Path must start with `file:` .
`proxy_url`	See CREATE_REPLICATION_SERVER. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.

Table 12-44 (Cont.) UPDATE_REPLICATION_SERVER Parameters

Parameter	Description
proxy_port	See CREATE_REPLICATION_SERVER. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
http_timeout	See CREATE_REPLICATION_SERVER.

UPDATE_SBT_ATTRIBUTE_SET

This procedure updates the parameters for the specified SBT attribute set.

If a parameter is null, then its value remains unchanged, except as noted in the following parameter descriptions.

Syntax

```
PROCEDURE update_sbt_attribute_set(
   attribute_set_name IN VARCHAR2,
   streams IN NUMBER DEFAULT dbms_ra_misc.number2null('p1'),
   poolid IN NUMBER DEFAULT NULL,
   parms IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p2'),
   send IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p3'));
```

Parameters

Table 12-45 UPDATE_SBT_ATTRIBUTE_SET Parameters

Parameter	Description
attribute_set_name	The name of the SBT attribute set to update.
streams	See CREATE_SBT_ATTRIBUTE_SET. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
poolid	See CREATE_SBT_ATTRIBUTE_SET.
parms	See CREATE_SBT_ATTRIBUTE_SET. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
send	See CREATE_SBT_ATTRIBUTE_SET. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.

UPDATE_SBT_JOB_TEMPLATE

This procedure updates the parameters for the specified SBT job.

If a parameter is null, then its value remains unchanged, except as noted in the following parameter descriptions.

Syntax

```
PROCEDURE update_sbt_job_template (
   template_name IN VARCHAR2,
   attribute_set_name IN VARCHAR2 DEFAULT NULL,
   backup_type IN VARCHAR2 DEFAULT NULL,
   from_tag IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p1'),
   priority IN NUMBER DEFAULT NULL,
   copies IN NUMBER DEFAULT NULL,
   window IN DSINTERVAL_UNCONSTRAINED DEFAULT dbms_ra_misc.intervalnull('p2'),
   compression_algorithm IN VARCHAR2 DEFAULT NULL,
   encryption_algorithm IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-46 UPDATE_SBT_JOB_TEMPLATE Parameters

Parameter	Description
`template_name`	The name of the SBT job template to update.
`attribute_set_name`	See CREATE_SBT_JOB_TEMPLATE.
`backup_type`	See CREATE_SBT_JOB_TEMPLATE.
`from_tag`	See CREATE_SBT_JOB_TEMPLATE. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
`priority`	See CREATE_SBT_JOB_TEMPLATE.
`copies`	See CREATE_SBT_JOB_TEMPLATE.
`window`	See CREATE_SBT_JOB_TEMPLATE. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
`compression_algorithm`	see CREATE_SBT_JOB_TEMPLATE. If you do not specify this parameter or if you specify `NULL`, then the Recovery Appliance retains the existing value. If you specficy a value, then that becomes the new setting. Specify `OFF` to remove the compression algorithm from this template. This changes the value of `COMPRESSION_ALGORITHM` to `NONE`.
`encryption_algorithm`	see CREATE_SBT_JOB_TEMPLATE.

UPDATE_SBT_LIBRARY

This procedure modifies the parameters for the specified SBT library.

If a parameter is null, then its value remains unchanged, except as noted in the `parms` and `send` descriptions.

Syntax

```
PROCEDURE update_sbt_library (
   lib_name IN VARCHAR2,
   drives IN NUMBER DEFAULT NULL,
   restore_drives IN NUMBER DEFAULT NULL,
   parms IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p1'),
   send IN VARCHAR2 DEFAULT dbms_ra_misc.varchar2null('p2'));
```

Parameters

Table 12-47 UPDATE_SBT_LIBRARY Parameters

Parameter	Description
`lib_name`	The name of the SBT library whose parameters are to be modified.
`drives`	See CREATE_SBT_LIBRARY.
`restore_drives`	See CREATE_SBT_LIBRARY.
`parms`	See CREATE_SBT_LIBRARY. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.
`send`	See CREATE_SBT_LIBRARY. If you do not specify this parameter, then the Recovery Appliance retains the existing value. If you specify a value (including null), then the Recovery Appliance sets the new value.

UPDATE_STORAGE_LOCATION

This procedure allocates additional space for the specified storage location. You cannot reduce the amount of space used by a storage location.

Syntax

```
PROCEDURE update_storage_location (
   storage_location_name IN VARCHAR2,
   storage_location_dests IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 12-48 UPDATE_STORAGE_LOCATION Parameters

Parameter	Description
`storage_location_name`	The name of the storage location to update.
`storage_location_dests`	This parameter accepts a string in the same format as the corresponding parameter to CREATE_STORAGE_LOCATION. This parameter is optional and can contain any or none of the original disk groups. For each disk group specified in the `storage_location_dests` parameter, this procedure adds the disk group to this storage location (unless it is already part of the location), and adds the requested amount of additional space to the storage location. If no size was specified, then the Recovery Appliance allocates all of the free space in the disk group to this storage location. If a size is specified, and if it is greater than the amount of space already allocated from this disk group, then the Recovery Appliance allocates the specified amount of space to this storage location. Any new disk groups that you add to this storage location must have the same allocation unit size as the disk groups already allocated to this storage location. All existing disk groups that were previously added to this storage location remain allocated to this location, even if you do not explicitly specify them while running this procedure. In addition to processing any change to storage destinations, this procedure reexamines any disk groups specified without an explicit size when the storage location was previously created or updated. If additional free space is available in any of the reexamined disk groups, then the Recovery Appliance allocates this space to the storage location.

13
Recovery Appliance View Reference

Describes the available Recovery Appliance views.

Summary of Recovery Appliance Views

Table 13-1 Recovery Appliance Views

Recovery Appliance View	Description
RA_ACTIVE_SESSION	This view lists information about active client sessions currently running in the Recovery Appliance.
RA_API_HISTORY	This view describes the history of user-issued API commands.
RA_CONFIG	This view lists the user configuration settings.
RA_DATABASE	This view lists the databases protected by this Recovery Appliance.
RA_DATABASE_STORAGE_USAGE	This view lists the storage usage for each protected database.
RA_DATABASE_SYNONYM	This view lists the protected databases and their equivalent names.
RA_DB_ACCESS	This view describes which Recovery Appliance user accounts have access to which protected databases.
RA_DISK_RESTORE_RANGE	The restore range of each protected database from disk backups on this Recovery Appliance.
RA_EM_SBT_JOB_TEMPLATE	This view lists defined SBT jobs and their statuses for Oracle Enterprise Manager.
RA_ENCRYPTION_INFO	This view describes the historical encryption key information.
RA_INCIDENT_LOG	This view describes the Recovery Appliance incidents.
RA_INCOMING_BACKUP_PIECES	This view describes the backup pieces being received by the Recovery Appliance.
RA_POLLING_FILES	This view lists the files backed up by Recovery Appliance from the polling location.
RA_POLLING_POLICY	This view lists the defined backup polling policies.
RA_PROTECTION_POLICY	This view lists the protection policies defined for this Recovery Appliance.
RA_PURGING_QUEUE	This view describes the order in which protected databases will have their oldest backups deleted when space is low.
RA_REPLICATION_SERVER	This view lists the replication server configurations.
RA_RESTORE_RANGE	This view describes the restore range of each protected database from all backups on this Recovery Appliance.
RA_SBT_ATTRIBUTE_SET	This view describes the defined SBT attribute set.

Table 13-1 (Cont.) Recovery Appliance Views

Recovery Appliance View	Description
RA_SBT_JOB	This view describes the defined SBT job templates.
RA_SBT_LIBRARY	This view lists the defined SBT libraries.
RA_SBT_RESTORE_RANGE	This view describes the restore range of each database from SBT backups on the Recovery Appliance.
RA_SBT_TASK	This view lists the queued background SBT tasks and their run statuses.
RA_SBT_TEMPLATE_MDF	This view lists missing level 0 data file backups for each SBT template.
RA_SERVER	This view describes the current settings for the Recovery Appliance.
RA_STORAGE_HISTOGRAM	This view describes the storage allocation history for recent time periods.
RA_STORAGE_LOCATION	This view lists defined Recovery Appliance storage locations and their allocations.
RA_TASK	This view lists queued background tasks and their run statuses.
RA_TIMER_TASK	This view describes timer process tasks and their planned executions.
RA_TIME_USAGE	This view describes the Recovery Appliance elapsed and idle time for the last 30 days.

RA_ACTIVE_SESSION

This view lists information about active client sessions currently running in the Recovery Appliance.

Column	Data Type	NULL	Description
INST_ID	NUMBER		The Recovery Appliance instance number where this session is running.
INSTANCE_NAME	VARCHAR2(16)		The Recovery Appliance instance name where this session is running.
HOST_NAME	VARCHAR2(64)		The Recovery Appliance host name where this session is running.
SID	NUMBER		The session ID for the active session.
SERIAL#	NUMBER		The session serial number, which uniquely identifies the objects in a session.
SPID	VARCHAR2(24)		The operating system process identifier.
DB_KEY	NUMBER		The primary key for this database in the recovery catalog.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique database name.
SBT_SID	VARCHAR2(64)		The SBT session identifier.
CLIENT_IDENTIFIER	VARCHAR2(64)		The client Identifier of the session.

Column	Data Type	NULL	Description
MODULE	VARCHAR2(64)		The name of the module that is currently executing.
ACTION	VARCHAR2(64)		The name of the action that is currently executing.
SQL_ID	VARCHAR2(13)		The SQL identifier of the SQL statement that is currently being executed.
EVENT	VARCHAR2(64)		The resource or event for which the session is waiting.
P1	NUMBER		First wait event parameter
P2	NUMBER		The second wait event parameter.
P3	NUMBER		The third wait event parameter.
WAIT_TIME	NUMBER		The wait time in hundredths of a second. See description of V$SESSION.WAIT_TIME for more information.
SECONDS_IN_WAIT	NUMBER		The wait time (in seconds). If the session is currently waiting, then the value is the amount of time waited for the current wait. If the session is not in a wait, then the value is the amount of time since the start of the most recent wait.
STATE	VARCHAR2(19)		The state of the wait event: WAITING, WAITED UNKNOWN TIME, WAITED SHORT TIME, WAITED KNOWN TIME. See description of V$SESSION.STATE for more information.
TASK_ID	NUMBER		The task identifier.
TASK_TYPE	VARCHAR2(30)		The task type.
PRIORITY	NUMBER		The task priority.
TASK_STATE	VARCHAR2(13)		The processing state for the task: EXECUTABLE, RUNNING, COMPLETED, TASK_WAIT, FAILED, and so on.
JOB_NAME	VARCHAR2(128)		The DBMS_SCHEDULER job name.

RA_API_HISTORY

This view describes the history of user-issued API commands.

Column	Data Type	NULL	Description
RESULTS	VARCHAR2(1000)		The results from running this command: SUCCESS or FAIL.
EXECUTE_TIME	TIMESTAMP(6) WITH TIME ZONE		The time at which the command started.
TASK_NAME	VARCHAR2(30)		The name of the task.
COMMAND_ISSUED	VARCHAR2(4000)		The full command as submitted by the user.
ELAPSED_SECONDS	NUMBER		The elapsed run time (in seconds) for the task.

RA_CONFIG

This view lists the user configuration settings.

Column	Data Type	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	The name of the configuration variable. See "RA_CONFIG" for variable definitions and default values.
VALUE	VARCHAR2(100)		The value of the configuration variable

RA_DATABASE

This view lists the databases protected by this Recovery Appliance.

Column	Data Type	NULL	Description
DB_UNIQUE_NAME	VARCHAR2(32)		The unique name of this protected database.
DB_KEY	NUMBER		The primary key for this database in the Recovery Appliance metadata database.
DELETING	VARCHAR2(7)		YES if this database is currently being deleted.
DBID	NUMBER		The DBID for this protected database.
CREATION_TIME	TIMESTAMP(6) WITH TIME ZONE		The time when this database was added to the Recovery Appliance.
POLICY_NAME	VARCHAR2(128)		The name of the protection policy used by this database.
STORAGE_LOCATION	VARCHAR2(128)		The name of the Recovery Appliance storage location used by this protected database.
RECOVERY_WINDOW_GOAL	INTERVAL DAY(9) TO SECOND(6)		The recovery window goal for backups on disk, as specified in the protection policy.
MAX_RETENTION_WINDOW	INTERVAL DAY(9) TO SECOND(6)		The maximum amount of time to retain disk backups. The Recovery Appliance deletes disk backups when they are older than this window. However, backups may be retained longer if deleting them would negatively affect the recovery_window_goal requirement.
RECOVERY_WINDOW_SBT	INTERVAL DAY(9) TO SECOND(6)		The recovery window for backups on tape, as specified in the protection policy.
TIMEZONE	VARCHAR2(64)		The time zone offset of the protected database.
SPACE_USAGE	NUMBER		The amount of disk space (in GB) currently used by this protected database.
KEEP_SPACE	NUMBER		The space used to hold KEEP backups for the database. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
DISK_RESERVED_SPACE	NUMBER		The amount of disk space (in GB) reserved for the exclusive use of this database

Column	Data Type	NULL	Description
GUARANTEED_COPY	VARCHAR2(3)		The status of the guaranteed copy setting: YES means that the Recovery Appliance replicates backups or copies them to tape before deleting them; NO means that the Recovery Appliance accepts new backups even if old backups must be purged because free space is low.
CUMULATIVE_USAGE	NUMBER		The cumulative amount of disk space (in GB) allocated for all backups received for this database.
REPLICATION_USAGE	NUMBER		The cumulative amount of disk space (in GB) replicated for this protected database.
SBT_USAGE	NUMBER		The cumulative amount of disk space (in GB) sent to SBT from this protected database.
REPLICATION_SETUP_STATUS	VARCHAR2(7)		The status of the setup for the downstream replication appliance for this database.
LAST_OPTIMIZE	TIMESTAMP(6) WITH TIME ZONE		The time when the most recent data placement optimization was completed.
LAST_VALIDATE	TIMESTAMP(6) WITH TIME ZONE		The time when the most recent validation of backup data was completed.
LAST_CROSSCHECK	TIMESTAMP(6) WITH TIME ZONE		The time when the most recent crosscheck of backup data was completed.
STORAGE_LOCATION_COUNT	NUMBER		The number of storage locations used by this database. If greater than one, then a storage location movement operation is in progress for this database.
STORAGE_MOVEMENT_PHASE	VARCHAR2(18)		The phase of the storage location movement operation for this protected database.
SIZE_ESTIMATE	NUMBER		The estimated space (in GB) consumed by the entire protected database.
RECOVERY_WINDOW_SPACE	NUMBER		The estimated space (in GB) that is needed to meet the recovery window goal.
RESTORE_WINDOW	INTERVAL DAY(9) TO SECOND(9)		The time range used to compute the value of RECOVERY_WINDOW_SPACE. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
DEDUPLICATION_FACTOR	NUMBER		The ratio of the total size of virtual full backups to the actual consumed space on the appliance for this protected database.
MINIMUM_RECOVERY_NEEDED	INTERVAL DAY(9) TO SECOND(9)		The minimum interval needed to restore the protected database to the present.
UNPROTECTED_WINDOW_THRESHOLD	INTERVAL DAY(9) TO SECOND(6)		The user-specified maximum amount of data loss for protected databases that are subject to a protection policy. The Recovery Appliance generates an alert if the unprotected window of this database exceeds this value.

Column	Data Type	NULL	Description
UNPROTECTED_WINDOW	INTERVAL DAY(9) TO SECOND(9)		The point beyond which recovery is impossible unless additional redo is available.
NZDL_ACTIVE	VARCHAR2(3)		YES if real-time redo transport is active. NO if redo has not recently been received.
ALLOW_BACKUP_DELETION	VARCHAR2(3)		The setting that controls whether RMAN backups for databases that use this protection policy can be deleted: NO means that the Recovery Appliance does not allow deletion of these backups; YES means that the Recovery Appliance allows deletion of these backups. Note that this parameter is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.7 and later.
STORE_AND_FORWARD	VARCHAR2(3)		The status of the Backup and Redo Failover setting: YES means that the Recovery Appliance applies the Backup and Redo Failover strategy to backups from the databases associated with this protection policy; NO means that the Backup and Redo Failover feature is not enabled and the Recovery Appliance applies the normal incremental-forever backup strategy instead. Note that this setting is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

RA_DATABASE_STORAGE_USAGE

This view lists the storage usage for each protected database.

Column	Data Type	NULL	Description
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database.
DB_KEY	NUMBER		The primary key for this protected database in the Recovery Appliance metadata database.
STORAGE_LOCATION	VARCHAR2(128)	NOT NULL	The name of the Recovery Appliance storage location used by this protected database.
USED_SPACE	NUMBER		The amount of space (in GB) used by this database in its Recovery Appliance storage locations. Backups for a protected database typically reside in only one storage location, but can reside in two locations when a movement operation is in progress.

RA_DATABASE_SYNONYM

This view lists the protected databases and their equivalent names.

Column	Data Type	NULL	Description
DB_UNIQUE_NAME	VARCHAR2(512)		The unique name of the protected database.
DBID	NUMBER		The DBID for all protected databases that are equivalent to this database.

RA_DB_ACCESS

This view describes which Recovery Appliance user accounts have access to which protected databases.

Column	Data Type	NULL	Description
USERNAME	VARCHAR2(128)	NOT NULL	The name of the Recovery Appliance user account.
DB_UNIQUE_NAME	VARCHAR2(32)		The unique name of the protected database accessed by the Recovery Appliance user account.
DB_KEY	NUMBER		The primary key for the protected database accessed by the Recovery Appliance user account.

RA_DISK_RESTORE_RANGE

The restore range of each protected database from disk backups on this Recovery Appliance.

Column	Data Type	NULL	Description
DB_KEY	NUMBER		The primary key of the protected database.
LOW_TIME	DATE		The earliest time to which the protected database can be restored.
HIGH_TIME	DATE		The latest time to which the protected database can be restored.
LOW_SCN	NUMBER		The lowest SCN to which the protected database can be restored.
HIGH_SCN	NUMBER		The highest SCN to which the protected database can be restored.
LOW_DBINC_KEY	NUMBER		The primary key for the incarnation of the target database to which LOW_SCN belongs.
HIGH_DBINC_KEY	NUMBER		The primary key for the incarnation of the target database to which HIGH_SCN belongs.
LAST_UPDATED	DATE		The time that the restore range for this protected database was updated.

RA_EM_SBT_JOB_TEMPLATE

This view lists defined SBT jobs and their statuses for Oracle Enterprise Manager.

Column	Data Type	NULL	Description
TEMPLATE_NAME	VARCHAR2(128)	NOT NULL	The name of the SBT job template.
FULL_TEMPLATE_NAME	VARCHAR2(128)	NOT NULL	The full name of the SBT job template.
POLICY_NAME	VARCHAR2(128)		The protection policy specifying the protected databases whose backups the Recovery Appliance considers eligible for copying.
DB_UNIQUE_NAME	VARCHAR2(512)		The unique name of the protected database whose backups the Recovery Appliance considers eligible for copying.
ATTRIBUTE_SET_NAME	VARCHAR2(128)	NOT NULL	The name of the SBT attribute set.
LIB_NAME	VARCHAR2(128)	NOT NULL	The name of the SBT library.
BACKUP_TYPE	VARCHAR2(16)		The types of backups to be copied to tape by this job: ALL, FULL, INCR, ARCH, or TAPE_RESERVE.
PRIORITY	NUMBER		The priority for scheduling this job.
COPIES	NUMBER	NOT NULL	The number of copies to be created on tape.
WINDOW	INTERVAL DAY(2) TO SECOND(6)		The time allotted for copy tasks to start for this job.
FROM_TAG	VARCHAR2(32)		The tag for the backup to be copied to tape by this job.
ERROR_TEXT	VARCHAR2(4000)		The error text for the task that failed.
ERROR_LAST_SEEN	TIMESTAMP(6) WITH TIME ZONE		The timestamp when the Recovery Appliance most recently detected the error.
EXECUTABLE	NUMBER		The number of tasks in an executable state.
RUNNING	NUMBER		The number of tasks that are running or retrying.
COMPLETED	NUMBER		The number of completed tasks.
COMPLETION_TIME	TIMESTAMP(6) WITH TIME ZONE		The time of the most recent completed task.
STATUS	VARCHAR2(5)		The status of the SBT library: READY, PAUSE, or ERROR.
BYTES	NUMBER		The number of bytes read or written so far.
COMPRESSION_ALGORITHM	VARCHAR2(6)		The compression algorithm used by this job: NONE, BASIC, LOW, MEDIUM, or HIGH. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

RA_ENCRYPTION_INFO

This view describes the historical encryption key information.

Column	Data Type	NULL	Description
ENCINFO_KEY	NUMBER	NOT NULL	The key of this encryption info record in the Recovery Appliance metadata database.

Column	Data Type	NULL	Description
DB_UNIQUE_NAME	VARCHAR2(32)	NOT NULL	The unique name of the database for this encryption info record..
DBID	NUMBER	NOT NULL	
CREATE_TIME	TIMESTAMP(6) WITH TIME ZONE		The time at which the key was created..
ENCRYPTION_TAG	VARCHAR2(128)	NOT NULL	The tag associated with this encryption key..
ENCRYPTION_KEYID	VARCHAR2(78)	NOT NULL	The encryption id associated with this encryption key.

RA_INCIDENT_LOG

This view describes the Recovery Appliance incidents.

Column	Data Type	NULL	Description
INCIDENT_ID	NUMBER		The unique ID for the incident.
ERROR_CODE	NUMBER		The Oracle error code for the message describing the incident.
PARAMETER	VARCHAR2(1000)		The parameter qualifying the scope of the error code.
ERROR_TEXT	VARCHAR2(4000)		The text of the message for the last detection of this error condition.
SL_KEY	NUMBER		Primary key of the storage location (if any) involved in this incident
SL_NAME	VARCHAR2(128)		The primary key of the Recovery Appliance storage location (if any) involved in this incident.
DB_KEY	NUMBER		The primary key of the protected database (if any) involved in this incident.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database (if any) involved in this incident.
TASK_ID	NUMBER		The ID for the task (if any) in which this incident was detected.
STATUS	VARCHAR2(6)		The status of this incident: ACTIVE, FIXED, or RESET.
COMPONENT	VARCHAR2(30)		The component of the Recovery Appliance detecting this incident.
SEVERITY	VARCHAR2(47)		The importance of this incident to the smooth operation of the Recovery Appliance.
FIRST_SEEN	TIMESTAMP(6) WITH TIME ZONE	NOT NULL	The timestamp when the Recovery Appliance first detected the incident.
LAST_SEEN	TIMESTAMP(6) WITH TIME ZONE	NOT NULL	The timestamp when the Recovery Appliance most recently detected the incident.
SEEN_COUNT	NUMBER	NOT NULL	The number of times that the Recovery Appliance detected the incident.

RA_INCOMING_BACKUP_PIECES

This view describes the backup pieces being received by the Recovery Appliance.

Column	Data Type	NULL	Description
SL_KEY	NUMBER		The primary key of the Recovery Appliance storage location storing this backup piece.
SL_NAME	VARCHAR2(128)		The name of the Recovery Appliance storage location storing this backup piece.
DB_KEY	NUMBER		The primary key of the protected database creating this backup piece.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database creating this backup piece.
HANDLE	VARCHAR2(1024)		The handle assigned to this backup piece.
CURRENT_SIZE	NUMBER		The size (in GB) currently allocated for this backup piece.
START_TIME	TIMESTAMP(6) WITH TIME ZONE		The time when the backup piece was first seen by the Recovery Appliance. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
LAST_UPDATE	TIMESTAMP(6) WITH TIME ZONE		The time when the backup piece was completely received.

RA_POLLING_FILES

This view describes the set of files the Recovery Appliance backed up from the configured polling location.

Note that this view is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

Column	Data Type	NULL	Description
POLL_NAME	VARCHAR2(128)		The name of the polling policy.
POLL_KEY	NUMBER	NOT NULL	The unique identifier of the polling policy.
FILE_NAME	VARCHAR2(512)		The name of the file discovered in the configured polling location.
FILE_SIZE	NUMBER		The size of the file the last time it was scanned in the polling location.
STATUS	VARCHAR2(24)		The current state of processing of the file.

RA_POLLING_POLICY

This view lists the defined backup polling policies.

Column	Data Type	NULL	Description
POLLING_NAME	VARCHAR2(128)		The name of this backup polling policy.
POLLING_KEY	NUMBER	NOT NULL	The primary key for this backup polling policy in the recovery catalog.
DEST	VARCHAR2(4000)	NOT NULL	The file system directory corresponding to the backup polling location.
FREQUENCY	INTERVAL DAY(9) TO SECOND(6)		The interval at which the Recovery Appliance scans the backup polling location for new files.
NEXT_EXECUTE	TIMESTAMP(6) WITH TIME ZONE		The next time when the polling location will be scanned. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
DELETE_INPUT	VARCHAR2(5)		The deletion policy for the polling location: TRUE to delete files as they are accepted; FALSE to keep all files.

RA_PROTECTION_POLICY

This view lists the protection policies defined for this Recovery Appliance.

Column	Data Type	NULL	Description
POLICY_NAME	VARCHAR2(128)	NOT NULL	The user-created name of the protection policy.
DESCRIPTION	VARCHAR2(128)		The protection policy description.
PROT_KEY	NUMBER	NOT NULL	The primary key for this protection policy in the Recovery Appliance metadata database.
SL_NAME	VARCHAR2(128)	NOT NULL	The name of the Recovery Appliance storage location used by this protection policy.
SL_KEY	NUMBER	NOT NULL	The primary key of the Recovery Appliance storage location used by this protection policy.
POLLING_NAME	VARCHAR2(128)		The name of the backup polling policy assigned to this protection policy.
RECOVERY_WINDOW_GOAL	INTERVAL DAY(9) TO SECOND(6)		The recovery window goal for restoring backups stored on disk.
MAX_RETENTION_WINDOW	INTERVAL DAY(9) TO SECOND(6)		The maximum amount of time that the Recovery Appliance must retain disk backups.
RECOVERY_WINDOW_SBT	INTERVAL DAY(9) TO SECOND(6)		The recovery window for restoring backups stored on tape.
UNPROTECTED_WINDOW	INTERVAL DAY(9) TO SECOND(6)		The point beyond which recovery is not possible unless additional redo is available.

Column	Data Type	NULL	Description
GUARANTEED_COPY	VARCHAR2(3)		The status of the guaranteed copy setting: YES means that the Recovery Appliance replicates backups or copies them to tape before deleting them; NO means that the Recovery Appliance accepts new backups even if old backups must be purged because free space is low.
REPLICATION_SERVER_LIST	VARCHAR2(4000)		The list of replication server configurations associated with this protection policy.
ALLOW_BACKUP_DELETION	VARCHAR2(3)		The setting that controls whether RMAN backups for databases that use this protection policy can be deleted: NO means that the Recovery Appliance does not allow deletion of these backups; YES means that the Recovery Appliance allows deletion of these backups. Note that this parameter is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.7 and later.
STORE_AND_FORWARD	VARCHAR2(3)		The status of the Backup and Redo Failover setting: YES means that the Recovery Appliance applies the Backup and Redo Failover strategy to backups from the databases associated with this protection policy; NO means that the Backup and Redo Failover feature is not enabled and the Recovery Appliance applies the normal incremental-forever backup strategy instead. Note that this setting is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

RA_PURGING_QUEUE

This view describes the order in which protected databases will have their oldest backups deleted when space is low.

Column	Data Type	NULL	Description
SL_NAME	VARCHAR2(128)	NOT NULL	The Recovery Appliance storage location name.
SL_KEY	NUMBER		The primary key for this Recovery Appliance storage location in the recovery catalog.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database whose backups the Recovery Appliance will purge.
DB_KEY	NUMBER	NOT NULL	The primary key for the protected database whose backups the Recovery Appliance will purge.
PURGE_ORDER	NUMBER		The order in which this protected database is eligible for purging.

Column	Data Type	NULL	Description
NEW_RECOVERY_WINDOW	INTERVAL DAY(9) TO SECOND(6)		The recovery window goal for this protected database after a purge.
NEW_PCT_RECOVERY	NUMBER		The percentage of the recovery window goal remaining for this protected database after a purge.
PCT_STORAGE	NUMBER		The percentage of reserved space being consumed by this protected database.

RA_REPLICATION_SERVER

This view lists the replication server configurations.

Column	Data Type	NULL	Description
REPLICATION_SERVER_NAME	VARCHAR2(128)	NOT NULL	The user-assigned name of the replication server configuration.
REPLICATION_SERVER_STATE	VARCHAR2(21)		The replication status of the downstream Recovery Appliance: AVAILABLE, CREATING, DELETING, TESTING COMMUNICATION, or null.
PROTECTION_POLICY	VARCHAR2(128)		The protection policy associated with this replication server configuration.
REP_SERVER_CONNECT_NAME	VARCHAR2(128)	NOT NULL	The user name used to connect to the downstream Recovery Appliance.
PROXY_HTTP_ADDRESS	VARCHAR2(519)		The address of the proxy server in the form *proxy_server_http_address:port_of_proxy_server*.
PROXY_TIMEOUT	NUMBER		The timeout period for the proxy server connection.
SBT_LIBRARY_NAME	VARCHAR2(128)	NOT NULL	The name of the SBT library with which this replication server configuration is associated.
SBT_LIBRARY_PARMS	VARCHAR2(1024)		The SBT library parameters.
ATTRIBUTE_NAME	VARCHAR2(128)	NOT NULL	The SBT attribute set name.
ATTRIBUTE_PARMS	VARCHAR2(1024)		The SBT parameters passed while allocating the RMAN channel.
WALLET_PATH	VARCHAR2(512)		The path to the local Oracle wallet (excluding the wallet file name).
WALLET_ALIAS	VARCHAR2(512)	NOT NULL	The alias that identifies the Oracle wallet credentials that this Recovery Appliance uses to log in to the downstream Recovery Appliance.
SERVER_HOST	CLOB	NOT NULL	The server name or address of the downstream Recovery Appliance.
MAX_STREAMS	NUMBER		The maximum number of simultaneous replication tasks. Each replication task processes a single backup piece. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

RA_RESTORE_RANGE

This view describes the restore range of each protected database from all backups on this Recovery Appliance.

Column	Data Type	NULL	Description
DB_KEY	NUMBER		The primary key of the protected database.
LOW_TIME	DATE		The earliest time to which the protected database can be restored.
HIGH_TIME	DATE		The latest time to which the protected database can be restored.
LOW_SCN	NUMBER		The lowest SCN to which the database can be restored.
HIGH_SCN	NUMBER		The highest SCN to which the protected database can be restored.
LOW_DBINC_KEY	NUMBER		The primary key for the incarnation of the target database to which the low SCN belongs.
HIGH_DBINC_KEY	NUMBER		The primary key for the incarnation of the target database to which the high SCN belongs.
LAST_UPDATED	DATE		The time that the restore range for this database was updated.

RA_SBT_ATTRIBUTE_SET

This view describes the defined SBT attribute set.

Column	Data Type	NULL	Description
ATTRIBUTE_SET_KEY	NUMBER	NOT NULL	The key of this SBT attribute set in the Recovery Appliance metadata database.
ATTRIBUTE_SET_NAME	VARCHAR2(128)	NOT NULL	The SBT attribute set name.
LIB_NAME	VARCHAR2(128)	NOT NULL	The name of the SBT library object with which this attribute set is associated.
STREAMS	NUMBER		The number of parallel streams available for jobs that run with this attribute set.
POOLID	NUMBER		The media pool identifier.
PARMS	VARCHAR2(1024)		The SBT parameters passed while allocating the RMAN channel.
SEND	VARCHAR2(1024)		The SEND command string passed to the allocated channel.

RA_SBT_JOB

This view describes the defined SBT job templates.

Column	Data Type	NULL	Description
TEMPLATE_KEY	NUMBER	NOT NULL	The key of this SBT job template in the Recovery Appliance metadata database.
TEMPLATE_NAME	VARCHAR2(128)	NOT NULL	The SBT job template name.
ATTRIBUTE_SET_NAME	VARCHAR2(128)	NOT NULL	The SBT attribute set name.
LIB_NAME	VARCHAR2(128)	NOT NULL	The SBT library name.
POLICY_NAME	VARCHAR2(128)		The protection policy specifying databases whose backups the Recovery Appliance considers eligible for copying to tape.
DB_KEY	NUMBER		The primary key of the protected database whose backups the Recovery Appliance considers eligible for copying to tape.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database whose backups the Recovery Appliance considers eligible for copying to tape.
BACKUP_TYPE	VARCHAR2(16)		The types of backups to be copied to tape by this job: ALL, FULL, INCR, ARCH, or TAPE_RESERVE.
FROM_TAG	VARCHAR2(32)		The backups with the specified tag to be copied to tape by this job.
PRIORITY	NUMBER		The priority for scheduling this job.
COPIES	NUMBER	NOT NULL	The number of copies to be created on tape.
LAST_SCHEDULE_TIME	TIMESTAMP(6) WITH TIME ZONE		The last time at which this SBT job was scheduled to run.
WINDOW	INTERVAL DAY(2) TO SECOND(6)		The time allotted for copy tasks to start for this job.
COMPRESSION_ALGORITHM	VARCHAR2(6)		The compression algorithm used by this job: NONE, BASIC, LOW, MEDIUM, HIGH, or OFF. OFF indicates that compression was explicitly turned off for this job (the compression setting in the SBT job template was ignored). Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

RA_SBT_LIBRARY

This view lists the defined SBT libraries.

Column	Data Type	NULL	Description
LIB_KEY	NUMBER	NOT NULL	The key of this SBT library in the Recovery Appliance metadata database.
LIB_NAME	VARCHAR2(128)	NOT NULL	The SBT library name.
DRIVES	NUMBER	NOT NULL	The number of drives available for use by this SBT library.
RESTORE_DRIVES	NUMBER	NOT NULL	The number of drives reserved for restore operations.

Column	Data Type	NULL	Description
PARMS	VARCHAR2(1024)		The SBT parameters passed while allocating an RMAN channel.
SEND	VARCHAR2(1024)		The SEND command string passed to the allocated channel.
STATUS	VARCHAR2(5)		The SBT library status: READY, PAUSE, ERROR, or null.
LAST_ERROR_TEXT	VARCHAR2(4000)		The most recent error text of the task that failed.

RA_SBT_RESTORE_RANGE

This view describes the restore range of each database from SBT backups on the Recovery Appliance.

Column	Data Type	NULL	Description
DB_KEY	NUMBER		The primary key of the protected database.
LOW_TIME	DATE		The earliest time to which the database can be restored.
HIGH_TIME	DATE		The latest time to which the database can be restored.
LOW_SCN	NUMBER		The lowest SCN to which the database can be restored.
HIGH_SCN	NUMBER		The highest SCN to which the database can be restored.
LOW_DBINC_KEY	NUMBER		The primary key for the incarnation of the target database to which the low SCN belongs.
HIGH_DBINC_KEY	NUMBER		The primary key for the incarnation of the target database to which the high SCN belongs.
LAST_UPDATED	DATE		The time that the restore range for this protected database was last updated.

RA_SBT_TASK

This view lists the queued background SBT tasks and their run statuses.

Column	Data Type	NULL	Description
TASK_ID	NUMBER		The ID for the task.
STATE	VARCHAR2(47)		The processing state for the task: EXECUTABLE, RUNNING, COMPLETED, TASK_WAIT, FAILED, and so on.
COMPLETION_TIME	TIMESTAMP(6) WITH TIME ZONE		The timestamp for task completion. The column is null if the task is not complete.

Column	Data Type	NULL	Description
ELAPSED_SECONDS	NUMBER		The elapsed run time (in seconds) for the task.
EXECUTE_INSTANCE_ID	NUMBER		The ID of the database instance ID on which the task must run. The column is null if the task can run on any instance.
ERROR_COUNT	NUMBER		The number of times that the task had errors.
ERROR_TEXT	VARCHAR2(4000)		The error text for the task that failed.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database for which the task is running.
DB_KEY	NUMBER		The primary key of the protected database for which the task is running.
RESTORE_TASK	VARCHAR2(3)		The type of task: YES if this is a restore task; NO if this is a backup task.
BS_KEY	NUMBER		The key of the backup set that is accessed by this task.
PIECE#	NUMBER		The number of the backup piece that is accessed by this task.
COPIES	NUMBER		The number of copies created by this task.
TEMPLATE_NAME	VARCHAR2(128)		The SBT job template to which this task belongs.
ATTRIBUTE_SET_NAME	VARCHAR2(128)		The name of the SBT attribute set to which this task belongs.
LIB_NAME	VARCHAR2(128)	NOT NULL	The name of the SBT library used by this task.
REPLICATION	VARCHAR2(3)		The type of task: YES if this is a replication task; NO if this is an SBT task.
FILENAME	VARCHAR2(513)		The name of the backup file being read or written.
START_TIME	TIMESTAMP(6) WITH TIME ZONE		The start time of this task.
BYTES	NUMBER		The number of bytes read or written so far.
TOTAL	NUMBER		The total number of bytes to be read or written.
COMPRESSION_ALGORITHM	VARCHAR2(6)		The compression algorithm used by this task: NONE, BASIC, LOW, MEDIUM, HIGH, or OFF. OFF indicates that compression was explicitly turned off for this task (the compression setting in the SBT job template was ignored). Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

RA_SBT_TEMPLATE_MDF

This view lists missing level 0 data file backups for each SBT template.

Column	Data Type	NULL	Description
TEMPLATE_KEY	NUMBER	NOT NULL	The key identifying the SBT template.
DB_KEY	NUMBER	NOT NULL	The key for the protected database that contains the missing file.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the database that contains the missing data file.
DF_FILE#	NUMBER		The number of the missing data file.
DF_TS#	NUMBER		The tablespace number of the missing data file.
DF_PLUGIN_CHANGE#	NUMBER		The plugin SCN for the missing data file.
DF_FOREIGN_DBID	NUMBER		The foreign DBID for the database that contains the missing data file.
DF_TABLESPACE	VARCHAR2(30)		The tablespace that contains the missing data file.
DF_CREATION_CHANGE#	NUMBER		The creation SCN for the missing data file.

RA_SERVER

This view describes the current settings for the Recovery Appliance.

Column	Data Type	NULL	Description
STATE	VARCHAR2(10)		The state of the Recovery Appliance: ON if the Recovery Appliance is running; OFF if it is not active.
NETWORK_CHUNKSIZE	NUMBER		The size (in MB) of network messages used by the Recovery Appliance client module to communicate with the Recovery Appliance.
SCHEDULERS	NUMBER		The number of normal schedulers currently running on the Recovery Appliance. This number excludes special purpose schedulers used for tape, replication, purge_immediate, or restore operations. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
RESOURCE_WAIT_TASK_LIMIT	VARCHAR2(48)		The limitations on task concurrency caused by resource waits.

RA_STORAGE_HISTOGRAM

This view describes the storage allocation history for recent time periods.

Column	Data Type	NULL	Description
NAME	VARCHAR2(128)	NOT NULL	The name of the Recovery Appliance storage location.
SL_KEY	NUMBER	NOT NULL	The primary key for this Recovery Appliance storage location in the recovery catalog.

Column	Data Type	NULL	Description
SLOT	NUMBER		The slot (ordered by sampling time period) in the histogram.
USAGE	NUMBER		The amount of space (in GB) that was allocated during the histogram slot.

RA_STORAGE_LOCATION

This view lists defined Recovery Appliance storage locations and their allocations.

Column	Data Type	NULL	Description
NAME	VARCHAR2(128)	NOT NULL	The Recovery Appliance storage location name.
SL_KEY	NUMBER	NOT NULL	The primary key for this Recovery Appliance storage location in the recovery catalog.
DISK_GROUPS	VARCHAR2(4000)		The list of names of Oracle ASM disk groups used for storage.
MIN_ALLOC	NUMBER		The minimum amount of storage (in GB) that may be allocated.
TOTAL_SPACE	NUMBER		The maximum amount of storage (in GB) that the Recovery Appliance storage location can use for backup data.
USED_SPACE	NUMBER		The amount of space (in GB) currently used in the Recovery Appliance storage location.
FREESPACE	NUMBER		The amount of space (in GB) available for immediate use.
FREESPACE_GOAL	NUMBER		The expected free space requirement (in GB) based on usage history. Purges may occur to meet this goal.
LAST_CHECK_FILES	TIMESTAMP(6) WITH TIME ZONE		The most recent time that files were checked for consistency.
SYSTEM_PURGING_SPACE	NUMBER		The amount of space (in GB) reserved for purging operations.

RA_TASK

This view lists queued background tasks and their run statuses.

Column	Data Type	NULL	Description
TASK_ID	NUMBER		The ID for the task.
TASK_TYPE	VARCHAR2(30)		The type of processing performed by the task.
PRIORITY	NUMBER		The run priority for the task.
STATE	VARCHAR2(47)		The processing state for the task: EXECUTABLE, RUNNING, COMPLETED, TASK_WAIT, FAILED, and so on.

Column	Data Type	NULL	Description
WAITING_ON	NUMBER		The ID of the task that is blocking this task when its state is TASK_WAIT.
CREATION_TIME	TIMESTAMP(6) WITH TIME ZONE		The time of task creation.
COMPLETION_TIME	TIMESTAMP(6) WITH TIME ZONE		The timestamp for task completion. The column is null if the task is not complete.
ELAPSED_SECONDS	NUMBER		The elapsed run time (in seconds) for the task.
ERROR_COUNT	NUMBER		Number of times that the task had errors
INTERRUPT_COUNT	NUMBER		The number of times that the task was interrupted.
LAST_INTERRUPT_TIME	TIMESTAMP(6) WITH TIME ZONE		The most recent time that the task was interrupted.
EXECUTE_INSTANCE_ID	NUMBER		The ID of the database instance on which the task must run. The column is null if the task can run on any instance.
LAST_EXECUTE_TIME	TIMESTAMP(6) WITH TIME ZONE		The most recent time that the task was restarted.
DB_UNIQUE_NAME	VARCHAR2(30)		The unique name of the protected database for which the task is running.
DB_KEY	NUMBER		The primary key of the protected database for which the task is running.
SL_NAME	VARCHAR2(128)		The name of the Recovery Appliance storage location used by the task.
SL_KEY	NUMBER		The primary key of the Recovery Appliance storage location used by the task.
OSPID	VARCHAR2(128)		The platform-specific ID of the process in which the task is current running.
INSTANCE_ID	NUMBER		The ID of the database instance on which the task is currently running.
LAST_INCIDENT_ID	NUMBER		The ID of the last incident reported by the task that is currently running. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
BP_KEY	NUMBER		The key of the backup piece that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
BS_KEY	NUMBER		The key of the backup set that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.

Column	Data Type	NULL	Description
DF_KEY	NUMBER		The key of the data file that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
VB_KEY	NUMBER		The key of the virtual backup that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
HANDLE	VARCHAR2(1000)		The media manager handle that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
FILENAME	VARCHAR2(4000)		The name of the backup file that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
LIB_KEY	NUMBER		The unique identifier of the SBT library that is accessed by this task. Note that this column is available only with Zero Data Loss Recovery Appliance software update 12.1.1.1.8 and later.
ARCHIVED	CHAR(1)		The archive status of the task: Y if it has moved to the archive; otherwise, N.

RA_TIMER_TASK

Timer process tasks and their planned executions.

Column	Data Type	NULL	Description
TIMER_TYPE	VARCHAR2(54)		Purpose for running this timer task.
NEXT_EXECUTE	TIMESTAMP(6) WITH TIME ZONE		Next planned execution for this timer task.
TIMER_INTERVAL	INTERVAL DAY(9) TO SECOND(6)		Frequency for repeating this timer task.
KEY	NUMBER		The poll_key, db_key, or lib_key operated referenced by this timer task.

RA_TIME_USAGE

This view describes the Recovery Appliance elapsed and idle time for the last 30 days.

Column	Data Type	NULL	Description
TOTAL_TIME	NUMBER		The sum of the elapsed times (in seconds) across all sessions.
IDLE_TIME	NUMBER		The sum of the idle times (in seconds) across all sessions.

14

rastat Utility Reference

This chapter provides details on the rastat utility. You use rastat to generate statistics to help you evaluate the performance of Recovery Appliance.

You can find the utility, `rastat.pl`, in the `/opt/oracle.RecoveryAppliance/client/` directory of a Recovery Appliance compute server.

rastat Command Syntax

```
perl rastat.pl --test=<options> --rasys=<string> --catalog=<string>
--filesize=<size>M --chunksize=<size>M --diskgroup=<string> --parms=<string>
--oracle_home=<string> --oracle_sid=<string>
```

Options

Table 14-1 rastat Options

Option	Description
`-h, --help`	Displays help information.

Table 14-1 (Cont.) rastat Options

Option	Description
`--test`	Specifies which of the following atomic tests to run: `[NETBACKUP|NETRESTORE|ASMREAD|ASMWRITE|CONTAINERREAD|CONTAINERWRITE|CONTAINERALLOC|ALL]` `NETBACKUP`: Measures the network performance of a protected database sending backup byte streams to the Recovery Appliance. Requires `--catalog`; `--filesize` is optional. `--parms` is also optional if it is already configured for the RMAN client. `NETRESTORE`: Measures the network performance of a protected databases receiving backup byte streams from the Recovery Appliance. Requires `--catalog`; `--filesize` is optional. `--parms` is also optional if it is already configured for the RMAN client. `ASMREAD`: Measures the disk I/O performance of the Recovery Appliance reading from an ASM disk group. Requires `--diskgroup` and `--rasys`. `--filesize` and `--chunksize` are optional. `ASMWRITE`: Measures the disk I/O performance of the Recovery Appliance writing to an ASM disk group. Requires `--diskgroup` and `--rasys`. `--filesize` and `--chunksize` are optional. `CONTAINERREAD`: Measures the disk I/O performance of the Recovery Appliance reading from container files. Requires `--diskgroup` and `--rasys`. `--filesize` and `--chunksize` are optional. `CONTAINERWRITE`: Measures the disk I/O performance of the Recovery Appliance writing to container files. Requires `--diskgroup` and `--rasys`. `--filesize` and `--chunksize` are optional. `CONTAINERALLOC`: Measures the Recovery Appliance container file allocation rate. Requires `--diskgroup` and `--rasys`. `--filesize` and `--chunksize` are optional. `ALL`: Performs all of the tests. All of the required options must be set.
`--rasys`	The connection string for the Recovery Appliance SYS account. Required for all I/O tests.
`--catalog`	The connection string for the Recovery Appliance virtual private catalog (VPC) account. Required for `NETBACKUP` and `NETRESTORE` tests.
`--filesize`	Optional. The file size in megabytes for the utility to use for the test. Setting the appropriate file size for your test requirements is highly recommended. The default file size is 1024M.
`--chunksize`	Optional. The chunk size in megabytes for the utility to use for the `CONTAINERREAD` or `CONTAINERWRITE` test. The default is the system configured chunk size.

Table 14-1 (Cont.) rastat Options

Option	Description
`--diskgroup`	The name of the disk group the I/O test should read from or write to. For example, `--diskgroup=+DISK1` specifies an ASM disk group and `--diskgroup=/:DELTA` specifies a container group. Required for all I/O tests.
`--parms`	Optional if already configured in RMAN. The PARMS setting in RMAN to use for a `NETBACKUP` or `NETRESTORE` test. This parameter must specify the location of libra.so and the wallet information. For example, `--parms='SBT_LIBRARY=/u01/oracle/lib/libra.so, ENV=(RA_WALLET=location=file:/u01/oracle/dbs/racredential_alias=ra-scan:1521/zdlra5:dedicated)'`.
`--oracle_home`	Optional. The `$ORACLE_HOME` environment variable. Use this option to set the variable or to override the current setting.
`--oracle_sid`	Optional. The `$ORACLE_SID` environment variable. Use this option to set the variable or to override the current setting.

15

Recovery Appliance Error Message Reference

This chapter provides details on the Zero Data Loss Recovery Appliance (Recovery Appliance) error messages, which occur between ranges `ORA-45100` and `ORA-45299`, and `ORA-64700` and up.

ORA-45100: Database incarnation went from *string* to *string*. Recovery Appliance repair is required.
Cause: A '`startup resetlogs`' command was executed on the Recovery Appliance. This caused old metadata to be used to refer to the storage locations. Before the Recovery Appliance can be started, a repair operation must be run to synchronize its metadata with its storage.

Action: Execute `DBMS_RA.STARTUP_RECOVERY_APPLIANCE`. If any incidents are logged during the subsequent repair, they will need to be corrected. Once they have been corrected, repeat the execution of `DBMS_RA.STARTUP_RECOVERY_APPLIANCE`.

ORA-45102: unable to allocate *string* bytes of storage
Cause: The Recovery Appliance was unable to allocate additional disk space in the storage location of the database for the current allocation. This condition may be due to one of the following reasons: * Guaranteed copy has been specified for a database and there are too many backups waiting to be copied to tape.
* The metadata of the Recovery Appliance is being repaired.
* Nothing can be purged within the storage location of the database.

Action: Add additional storage to the storage location of the database.

ORA-45109: metadata for database *string*; file *string* is corrupt
Cause: Internal self checks found corruption in the metadata used to manage the Recovery Appliance block pool.

Action: Contact Oracle Support Services and provide trace and alert files.

ORA-45111: Task *string* is being terminated after *string* restarts.
Cause: A Recovery Appliance task generated too many errors. Following an error, a task is normally restarted. If it fails to restart after 10 tries, the Recovery Appliance marks the task as broken and no longer tries to restart it.

Action: Correct the error that terminated the task and request the task to be rerun.

ORA-45113: Recovery Appliance internal error *string*
Cause: An internal error was encountered.

Action: Contact Oracle Support Services and provide trace and alert files.

ORA-45114: file "*string*" not referenced by metadata for storage location *string*
Cause: A consistency check performed by the check files task of the Recovery Appliance identified that the file was not being referenced by the metadata of the

Recovery Appliance. Without these references, the Recovery Appliance cannot manage the file.

Action: If the file was inadvertently put in the storage location, it should be moved elsewhere. If the file has been separated from its metadata, contact Oracle Support Services and provide trace and alert files.

ORA-45115: database with DB_KEY *string* is too big to move.
Cause: An attempt was made to move the specified database to a new storage location, but the database could not be shrunk to within its storage reservation and still preserve its retention window.

Action: Increase the storage reservation for the database or shrink its retention window.

ORA-45116: anomaly detected while reading metadata for backup piece with BP_KEY *string*
Cause: A transient anomaly was found in the backup data.

Action: If the anomaly persists, find a copy of the backup piece, if available, and reinsert it into the storage location. If no copy is available, generate a new level 0 backup for all data files in the backup piece.

ORA-45118: servlet timeout error
Cause: A restore task was waiting on a servlet process to pass data to a client. The time allotted for responding was exceeded and the restore task was aborted.

Action: This can be a common occurrence if the client cancels the restore request. Reissue the request.

ORA-45119: received a nonexistent operation for privilege change
Cause: An illegal option was specified.

Action: `BACKUP`, `RECONCILE`, `READ`, `WRITE`, and `NULL` are only supported values.

ORA-45120: operation failed due to insufficient space
Cause: The storage location was too small to support the new database.

Action: increase the size of your storage location or reduce `DISK_RESERVED_SPACE` in the protection policy

ORA-45121: received an incorrect value for a privilege change
Cause: An internal error was detected while granting or revoking privileges.

Action: Contact Oracle Support Services and provide trace and alert files.

ORA-45122: invalid size or number specified
Cause: An invalid size or number was specified.

Action: Use a non-NULL or number greater than 0.

ORA-45123: The name *string* (*string*) already exists.
Cause: The object name was not unique.

Action: Specify a unique name for this object.

ORA-45124: Object *string* (*string*) is referenced and cannot be deleted.
Cause: The object was in use by a storage location or database.

Action: Delete all objects that reference this item.

ORA-45125: Object *string* (*string*) did not exist.
Cause: The object name did not exist.

Action: Specify an existing object.

ORA-45126: failed to delete database *string*
Cause: The database could not be deleted. An unexpected error has occurred.

Action: Examine the associated messages to determine the cause of the exception.

ORA-45127: Required parameter *string* must be specified.
Cause: The parameter was not supplied to API routine.

Action: Rerun the command specifying the missing parameter.

ORA-45128: backup piece *string* in database *string* is not referenced by the catalog
Cause: A consistency check performed by the check files task of the Recovery Appliance identified that the specified backup piece was unreferenced by the RMAN catalog. Without this reference, the Recovery Appliance cannot reclaim space used by this piece.

Action: Contact Oracle Support Services.

ORA-45129: expected *string* bytes used by database *string* in storage location *string*, but found *string* bytes used
Cause: A consistency check performed by the check files task of the Recovery Appliance identified that the storage usage of the database in a storage location did not match the sum of the size of storage pieces assigned to the Recovery Appliance.

Action: Contact Oracle Support Services.

ORA-45130: Storage parameter overlaps with storage in *string*.
Cause: A parameter for a storage location was specified that overlapped storage previously assigned to another storage location.

Action: Reissue the command specifying a different location for the storage.

ORA-45131: illegal or unknown restore compression option specified
Cause: The specified compression option was not supported on either the Recovery Appliance database or the database providing the backup.

Action: Query `V$RMAN_COMPRESSION_ALGORITHM` view to ensure the algorithm name matches one of the algorithm names in that table and that the option has `IS_VALID = 'YES'` and that the `INITIAL_RELEASE` column is less than both the Recovery Appliance and the database providing the backup. Reissue the command specifying a valid compression algorithm name.

ORA-45132: corrupt block detected in backup piece
Cause: A corrupt block was detected in a backup piece when populating the Recovery Appliance block pool.

Action: Perform block media recovery on the corrupt blocks of the database and do a cumulative level 1 backup.

ORA-45133: expected *string* byte allocation by database *string*, but found *string* bytes allocated
Cause: A consistency check performed by the check files task of the Recovery Appliance found that the storage allocations of the database did not match the sum of the size of allocations for that database in all storage locations.

Action: Contact Oracle Support Services.

ORA-45135: request terminated by the Recovery Appliance
Cause: A request was holding resources needed by the Recovery Appliance and was terminated to free those resources. This can be the result of a lack of disk space or some other resource.

Action: Check available disk space, as well as for errors on the Recovery Appliance database.

ORA-45136: invalid value for parameter *string*
Cause: The value supplied for the specified parameter was invalid.

Action: Check the Recovery Appliance documentation and rerun the command with a correct value.

ORA-45137: unknown platform
Cause: The Recovery Appliance has not received any backups from System Backup to Tape (SBT) or through polling. This is necessary for the Recovery Appliance to learn about the protected platform of the database and for the current operation to succeed.

Action: Backup a small archived log or other backup using SBT or by sending it to the polling location. Then retry this operation.

ORA-45138: Backup not found.
Cause: The specified backup could not be found in the catalog.

Action: Please check and specify the correct backup piece key or backup set key.

ORA-45139: A useful backup could not be found to correct this corruption.
Cause: A virtual backup piece key was provided, but there was no known backup on tape or disk to correct this backup.

Action: If the broken backup is the oldest virtual backup for the data file, sometimes an even older backup will have the data needed to correct the catalog. Find and specify that older backup directly.

ORA-45140: cannot insert backup into catalog
Cause: The specified backup was either not an incremental or not in the proper SCN range to correct problems in the catalog.

Action: Make sure you have provided the correct key value and find a proper incremental backup piece.

ORA-45141: File "*string*" was missing from storage location *string*.
Cause: During recovery of the Recovery Appliance, the specified file was referenced by the metadata of the Recovery Appliance, but was not found in its storage location.

Action: The file should be recovered from a replicated Recovery Appliance if it exists. If the file has been separated from its metadata, then contact Oracle Support Services and provide trace and alert files.

ORA-45142: The Recovery Appliance prerequisite is already set up.
Cause: The `DBMS_RA_INSTALL` procedure was executed to set up prerequisite objects for creation of a catalog schema for the Recovery Appliance. This error is reported because there can be only one user schema that manages the Recovery Appliance for the database.

Action: To re-create the Recovery Appliance schema in another user schema, uninstall the earlier Recovery Appliance schema setup.

ORA-45143: The Recovery Appliance prerequisite setup administors user name is mismatched.
Cause: The `DBMS_RA_INSTALL` procedure was executed to uninstall the Recovery Appliance prerequisite object for the wrong user name.

Action: Correct the user name parameter for `DBMS_RA_INSTALL` and reexecute the procedure.

ORA-45144: Undefined initial replication type for protection policy.
Cause: The initial_replication_type was undefined for the protection policy.

Action: Update the protection policy initial_replication_type with one of `LAST` `FULL`, `ALL`, or `NONE`.

ORA-45145: Recovery Appliance user *string* does not exist.
Cause: The Recovery Appliance user did not exist.

Action: Specify an existing Recovery Appliance user.

ORA-45146: Storage location *string* needs *string* additional bytes of storage.
Cause: The metadata of the Recovery Appliance was being repaired following a database open with the '`resetlogs`' command and the storage allocated in the specified storage location was insufficient. This may be caused by either an '`update_storage_location`' call being lost due to the 'resetlogs' command or the storage location becoming very low on free storage when the resetlogs command was executed.

Action: Update the storage location with the specified values and try the repair again by executing `DBMS_RA.STARTUP_RECOVERY_APPLIANCE`.

ORA-45147: Database *string* and database *string* are both moving.
Cause: The metadata of the Recovery Appliance was being repaired following a database open with 'resetlogs' and two databases were found to be moving between storage locations. The Recovery Appliance will only function correctly when one database is being moved. This may be caused by an 'update_protection_policy' or 'update_db' call being lost due to the 'resetlogs' command.

Action: Determine the storage locations used by each database and repeat any database movements that may have been lost.

ORA-45148: must fix *string* errors before restarting the Recovery Appliance
Cause: During a repair of the metadata of the Recovery Appliance, errors were found that precluded the restart of the Recovery Appliance.

Action: Fix the identified errors and execute `DBMS_RA.STARTUP_RECOVERY_APPLIANCE` to retry the repair.

ORA-45149: unknown task type: *string*
Cause: The Recovery Appliance tried to execute a task with an unknown task type.

Action: Contact Oracle Support Services.

ORA-45150: File *string* references unknown DBID *string*.
Cause: During a repair of the metadata of the Recovery Appliance, the specified file was found that referenced the specified database which was unknown to the Recovery Appliance. This may be caused by an '`add_db`' call being lost due to a '`resetlogs`' command.

Action: Repeat the lost '`add_db`' call.

ORA-45151: bad locking protocol for lock *string*
Cause: An internal error caused locking to be used incorrectly.

Action: Contact Oracle Support Services.

ORA-45152: bad backup piece format for *string*
Cause: During a repair of the metadata of the Recovery Appliance, the specified file was found whose type could not be determined.

Action: Remove the corrupted file and reexecute `DBMS_RA.STARTUP_RECOVERY_APPLIANCE`.

ORA-45153: unknown data file *string* for DBID *string*
Cause: During repair of the metadata of the Recovery Appliance, data for the specified database was found whose data file could not be found.

Action: Using RMAN on the specified database, use the '`resync`' command to refresh the metadata on the Recovery Appliance and retry the restart of the Recovery Appliance.

ORA-45154: bad Recovery Appliance format found in file *string*
Cause: During a repair of the metadata of the Recovery Appliance, a file was found in a storage location that was neither a chunk file nor a backup piece.

Action: Remove the offending file from the storage location of the Recovery Appliance and retry the restart of the Recovery Appliance.

ORA-45155: The Recovery Appliance has not been installed.
Cause: The Recovery Appliance was never installed on this database. The requested procedure is only supported on the Recovery Appliance.

Action: Do not attempt the procedure except on the Recovery Appliance.

ORA-45156: SBT job *string* not found
Cause: The specified SBT job was not found.

Action: Check if the SBT job has been deleted by user. If so, then drop the scheduler job.

ORA-45157: Parameter value *string* (*string*) is invalid.
Cause: The specified value for the parameter was invalid.

Action: Specify a valid value.

ORA-45158: SBT library *string* is not ready.
Cause: The specified SBT library was found to not be ready.

Action: Check if the library has been paused by user. If so, then resume the SBT library.

ORA-45159: RECOVERY_WINDOW_GOAL is lost for database *string*.
Cause: A low space condition forced the deletion of backups needed to support the recovery window goal for the named database.

Action: This is a warning and no action is needed. However, you may use `DBMS_RA.UPDATE_PROTECTION_POLICY` to increase the `DISK_RESERVED_SPACE` value of the database to ensure additional backups are saved. Select `SPACE_USAGE` from `RA_DATABASE` to see how much space is currently in use. You should also check for `KEEP` backups consuming space on disk and decide if they should be moved to tape or other disk storage. See `DBMS_RA.MOVE_BACKUP_PIECE` for more details.

ORA-45160: Incremental forever strategy is lost for database *string*.
Cause: A low space condition has forced the deletion of backup data needed to generate the last remaining virtual `LEVEL 0` of one or more data files. The next client backup will be a full `LEVEL 0` backup, even if `LEVEL 1` was specified.

Action: This is a warning and no action is needed. However, you may use `DBMS_RA.UPDATE_DB` to increase the `RESERVED_SPACE` value of the database to ensure additional backups are saved. Select `SPACE_USAGE` from `RA_DATABASE` to see how much space is currently in use. You should also check for `KEEP` backups consuming space on disk and decide if they should be moved to tape or other disk storage. See `DBMS_RA.MOVE_BACKUP_PIECE` for more details.

ORA-45161: The backup piece size cannot exceed database DISK_RESERVED_SPACE.
Cause: An individual backup piece exceeded the database protection policy `DISK_RESERVE_SPACE` value. A safe `DISK_RESERVED_SPACE` value would exceed the size of the database.

Action: Use `DBMS_RA.UPDATE_DB` to increase the `DISK_RESERVED_SPACE` value of the database.

ORA-45162: System global area memory is configured incorrectly.
Cause: Check initialization parameters `LARGE_POOL_SIZE` and `SHARED_POOL_SIZE`. The Recovery Appliance will use all of `LARGE_POOL_SIZE` or 20% of `SHARED_POOL_SIZE` to restore virtual or tape backups. The actual space needed is `NETWORK_CHUNKSIZE * 2 * (number of concurrent restore channels)` where `NETWORK_CHUNKSIZE` is set using `DBMS_RA.CONFIG`. Use `DBMS_RA.CONFIG` to lower `NETWORK_CHUNKSIZE` or preferably, increase either `LARGE_POOL_SIZE` or `SHARED_POOL_SIZE`.

Action: Check initialization parameters `LARGE_POOL_SIZE` and `SHARED_POOL_SIZE` and set it correctly.

ORA-45163: operation is only supported for user *string*
Cause: An attempt was made to start the Recovery Appliance by a user other than the Recovery Appliance administrator.

Action: Only start the Recovery Appliance as the user specified at installation time.

ORA-45164: The Recovery Appliance is not running.
Cause: An attempt was made to use the Recovery Appliance, but the Recovery Appliance has been deactivated by the administrator.

Action: Have the Recovery Appliance administrator execute DBMS_RA.STARTUP_RECOVERY_APPLIANCE and retry the operation.

ORA-45165: Recovery Appliance backup piece with BP_KEY *string* is corrupt
Cause: Corruption was found in the backup data.

Action: Ensure that you have a functioning backup of the affected data file. Then delete the corrupt backup piece to clear the condition.

ORA-45166: unable to access file *string*
Cause: An attempt was made to access the specified file which resulted in an error. An explanation of the error appears in the following messages.

Action: Verify the correctness of the file. If it is corrupt, either delete it or replace it.

ORA-45167: unable to validate backup piece with BP_KEY *string*
Cause: An attempt was made to validate the specified backup piece which resulted in an error. An explanation of the error appears in the following messages.

Action: Verify the correctness of the backup piece. If it is corrupt, either delete it or replace it.

ORA-45168: unexpected scheduler exit while executing task ID *string* of type *string*
Cause: A task failed with an unexpected error code in the Recovery Appliance.

Action: Contact Oracle Support Services.

ORA-45169: unexpected timer process exit
Cause: The timer process failed with an unexpected error code in the Recovery Appliance.

Action: Contact Oracle Support Services.

ORA-45170: Storage location *string* is too full.
Cause: Purging the specified storage location would result in the loss of the recovery window goal for one of its databases.

Action: Add more storage to the storage location or remove some databases from the storage location or reduce the recovery window goal for some of the databases in the storage location.

ORA-45171: The chunk optimization task has not run recently for one or more databases.
Cause: The background chunk optimization task had not been performed recently for at least one database. This may happen if the Recovery Appliance is too overloaded with foreground activities to have time to do background tasks.

Action: Remove some load from the Recovery Appliance by reducing the frequency of backups by protected databases or by offloading some of the databases from the Recovery Appliance.

ORA-45172: The validation task has not run recently for one or more databases.
Cause: The background validation task had not been performed recently for at least one database. This may happen if the Recovery Appliance is too overloaded with foreground activities to have time to do background tasks.

Action: Remove some load from the Recovery Appliance by reducing the frequency of backups by protected databases or by offloading some of the databases from the Recovery Appliance.

ORA-45173: The checkfiles task has not run recently for one or more storage locations.
Cause: The background checkfiles task had not been performed recently for one or more storage locations. This may happen if the Recovery Appliance is too overloaded with foreground activities to have time to do background tasks.

Action: Remove some load from the Recovery Appliance by reducing the frequency of backups by protected databases or by offloading some of the databases from the Recovery Appliance.

ORA-45174: unable to use replication server *string*
Cause: While using the Recovery Appliance, either a backup failed to be transmitted to the target replicated Recovery Appliance or a restore request failed to complete on the replicated Recovery Appliance.

Action: Check the following error messages to diagnose the actual error.

ORA-45175: unable to use SBT library *string*
Cause: While using the Recovery Appliance, a request failed to complete while using the specified System Backup to Tape library.

Action: Check the following error messages to diagnose the actual error.

ORA-45176: Replication server *string* is not in the paused state.
Cause: An attempt was made to update information for a replication server that was not in a paused state.

Action: Pause the replication server on this Recovery Appliance.

ORA-45177: unable to find file *string* previously found while polling
Cause: A backup piece file previously found in a polling location was later not accessible to the Recovery Appliance.

Action: If the file was unavailable due to network errors, the file will be found again once the network is available. If the backup piece was deleted, create a new backup.

ORA-45178: The allocation unit size cannot be changed.
Cause: An attempt was made to move one or more databases into a storage location with a different allocation unit size. This value comes from the ASM disk group allocation unit size specified when creating the disk groups referenced by the storage location.

Action: Use a storage location with the same minimum allocation size as the source. If necessary create new disk groups with a matching size before creating a new storage location.

ORA-45179: The reconcile task has not run recently for database *string*.
Cause: The background reconcile task had not been performed recently for the specified database. This may happen if the Recovery Appliance is too overloaded with foreground activities to have time to do background tasks.

Action: Remove some load from the Recovery Appliance by reducing the frequency of backups by protected databases or by offloading some of the databases from the Recovery Appliance.

ORA-45180: The crosscheck task has not run recently for database one or more databases.
Cause: The background reconcile task had not been performed recently for at least one database. This may happen if the Recovery Appliance is too overloaded with foreground activities to have time to do background tasks.

Action: Remove some load from the Recovery Appliance by reducing the frequency of backups by protected databases or by offloading some of the databases from the Recovery Appliance.

ORA-45182: database access cannot be granted or revoked using catalog owner or sys
Cause: The catalog owner or `SYS` was specified as the user in the `DBMS_RA.GRANT_DB_ACCESS` and `DBMS_RA.REVOKE_DB_ACCESS` procedures. This is not allowed.

Action: A different user should be created and granted the necessary access.

ORA-45183: request is blocked by session *string* on instance *string*
Cause: An API request was made to the Recovery Appliance while another API was in progress. Only one API may be performed at a time.

Action: Wait for the other API to complete or kill the identified session before repeating the failed API request.

ORA-45184: ORA-*string* occurred during wallet operation; WRL *string*
Cause: An operation on the wallet failed due to the indicated error.

Action: Refer to the indicated Oracle message for more information.

ORA-45185: alias *string* not found in *string* wallet
Cause: The specified WALLET alias did not appear in the wallet.

Action: Check the WALLET alias or create an alias in the wallet for the specified attribute and retry the command.

ORA-45187: storage location *string* is unusable; container repair key is *string*
Cause: During a repair of the Recovery Appliance, fatal errors were detected while rebuilding the specified storage location.

Action: Inspect the alert log for the instance upon which the repair command was issued for the errors detected. If you detect that the errors are caused by missing disk groups, add those disk groups and execute the `STARTUP_RECOVERY_APPLIANCE` API. If you decide that the missing data cannot be restored, execute the `REPAIR_STORAGE_LOCATION` API with the `REPLACE` option prior to executing the `STARTUP_RECOVERY_APPLIANCE` API.

ORA-45188: storage location *string* requires repair; container repair key is *string*
Cause: During a repair of the Recovery Appliance, consistency errors were detected while rebuilding the specified storage location.

Action: Inspect the alert log for the instance upon which the repair command was issued for the errors detected. If you detect that the errors are caused by missing disks or disk groups, add those disks or disk groups and execute the `STARTUP_RECOVERY_APPLIANCE` API. If you decide that the missing data cannot be restored, execute the `REPAIR_STORAGE_LOCATION` API with the `REPLACE` option prior to executing the `STARTUP_RECOVERY_APPLIANCE` API.

ORA-45189: repair failed because storage location was renamed from *string* to *string*
Cause: During a repair of the Recovery Appliance, a storage location was found whose name was different from the name originally used to define the storage location.

Action: Delete the bad storage location and re-create it with the proper name.

ORA-45190: anomaly detected while reading metadata for database with DB_KEY *string*
Cause: A transient anomaly was found in the backup data.

Action: If the anomaly persists, generate a new level 0 backup for all data files in the database.

ORA-45191: no suitable SBT library was found for the Recovery Appliance backups
Cause: Recovery Appliance could not find a suitable System Backup to Tape (SBT) library for performing the Recovery Appliance metadata backups.

Action: Create an SBT library that can be used for the Recovery Appliance backup.

ORA-45192: reservation already exists for the Recovery Appliance backup
Cause: An attempt was made to create a new reservation to perform a Recovery Appliance metadata backup when an unexpired reservation exists.

Action: Remove the existing reservation and then create a new reservation.

ORA-45193: multiple SBT libraries are present
Cause: More than one System Backup to Tape (SBT) library was found that can be used to back up the Recovery Appliance metadata.

Action: Specify a name while reserving the SBT library.

ORA-45194: Recovery Appliance metadata backup to SBT library failed
Cause: An error occurred while backing up the Recovery Appliance metadata to the System Backup to Tape (SBT) library. The error could be caused by the SBT library configuration or an internal Recovery Appliance error.

Action: Check the SBT library configuration or `RA_INCIDENT_LOG` view.

ORA-45195: reservation wait time exceeded
Cause: A timeout occurred while waiting for the system backup to tape (SBT) library reservation.

Action: Increase the wait time for the reservation by modifying the '`_drive_wait_minutes`' configuration parameter and retry the operation.

ORA-45196: failed to unreserve existing reservation
Cause: The existing System Backup to Tape (SBT) library reservation could not be unreserved.

Action: Check the '`ERROR_LOG`' table and trace files for information about the cause of this error.

ORA-45197: SBT library *string* could not be found for reservation
Cause: The Recovery Appliance could not find a System Backup to Tape (SBT) library for the given name.

Action: Check the SBT library name and retry the operation.

ORA-45198: machine is not a physical Recovery Appliance
Cause: Recovery Appliance services were attempted to start on a machine that was not a physical Recovery Appliance.

Action: The Recovery Appliance services cannot be started on this system.

ORA-45199: Error *string* encountered when executing *string*.
Cause: An error was encountered when executing PL/SQL code. This message should be accompanied by other error message(s) indicating the cause of the error.

Action: Check the accompanying errors.

ORA-45200: HTTP status code: *string*
Cause: The indicated HTTP status code was received while processing a servlet request,

Action: None

ORA-45201: additional Information: *string*
Cause: The indicated additional error was received while processing a servlet request.

Action: None

ORA-45202: operation failed, retry possible
Cause: A backup, restore operation failed while processing a servlet request. The operation may be retried.

Action: This message is used by the SBT client to decide whether to retry the operation.

ORA-45203: failed to *string* backup piece file "*string*"
Cause: An OS operation on the specified backup piece returned an error.

Action: Check additional messages.

ORA-45210: resource busy, retry possible
Cause: A backup or restore operation failed while processing a servlet request. The operation may be retried.

Action: This message is used by the SBT client to decide whether to retry the operation.

ORA-45211: error encountered while sending data; error code *string*
Cause: An error was encountered while sending data to the client.

Action: Check additional messages.

ORA-45212: error encountered while receiving data; error code *string*
Cause: An error was encountered while receiving data from client.

Action: Check additional messages.

ORA-45213: user or role '*string*' does not exist
Cause: There was no user or role with the name specified.

Action: Provide a valid user name or role.

ORA-45214: cannot convert '*string*' to number
Cause: An arithmetic, numeric, string, conversion, or constraint error occurred. For example, a NULL value was assigned to a variable that was declared as NOT NULL or an integer larger than 99 was assigned to a variable declared as `NUMBER(2)`.

Action: Change the data, how it is manipulated, or how it is declared so that values do not violate defined constraints.

ORA-45215: cannot delete a replication server that is in use
Cause: An attempt was made to delete a replication server that was actively restoring a backup.

Action: Cancel the restore or wait for the restore to complete before deleting the replication server.

ORA-45216: backup metadata of *string* (*string*) for database *string* was not found
Cause: The reported backup metadata was not found.

Action: This is an informational message. Ensure that you retain the backups until SBT tasks are executed.

ORA-45217: SBT task *string* is not found
Cause: The specified SBT task was not found.

Action: Provide a valid SBT task identifier and retry the command.

ORA-45264: error encountered during Recovery Appliance test recovery *string*
Cause: As part of Recovery Appliance protection, test recovery was performed by the Recovery Appliance metadata protection script. The test recovery uses the image copies, without additional disk space requirement to restore datafiles, to test the database.

Action: Additional information regarding this failure is recorded in the `RA_INCIDENT_LOG` and is also displayed in the Oracle Enterprise Manager console.

ORA-45265: error encountered during Recovery Appliance backup health check *string*
Cause: As part of Recovery Appliance protection, backup health check was performed by the Recovery Appliance metadata protection script. The backup health check uses the database 'validate' and 'preview' command to test the backups.

Action: Additional information regarding this failure is recorded in the `RA_INCIDENT_LOG` and is also displayed in the Oracle Enterprise Manager console.

ORA-45266: error encountered during Recovery Appliance database health check *string*
Cause: As part of Recovery Appliance protection, database health check was performed by the Recovery Appliance metadata protection script. The database health check uses the 'backup validate' command to perform database health check.

Action: Additional information regarding this failure is recorded in the `RA_INCIDENT_LOG` and is also displayed in the Oracle Enterprise Manager console.

ORA-45275: container: '*string*'
Cause: This message reports the name of the Recovery Appliance container involved in other messages.

Action: See associated error messages for a description of the problem.

ORA-45276: could not create container
Cause: A container creation failed. There will be other messages printed in the error stack that show more details about the problem.

Action: Investigate the entire error stack. If the problem is correctable, do so and retry creating this container.

ORA-45277: New AU size *string* differs from existing AU size *string* in group *string*.
Cause: An attempt was made to create a new container that has a different AU size than the other containers that already exist in this container group.

Action: Specify a container with the same AU size as the other containers in the container group.

ORA-45278: Container group *string* is not empty.
Cause: An attempt was made to drop the specified container group but it is not empty.

Action: Either remove the remaining objects from the group or use the `FORCE` option. Note that the `FORCE` option will irretrievably delete all files in the container group.

ORA-45279: Container group *string* does not exist.
Cause: The specified container group does not exist.

Action: Specify a container group that exists.

ORA-45280: Container group *string* already exists.
Cause: The specified container group already exists.

Action: Specify a container group that does not exist.

ORA-45281: Size of *string* bytes exceeds maximum container size of *string* bytes.
Cause: You tried to create a container greater than the maximum size.

Action: Specify a smaller size.

ORA-45282: error identifying container
Cause: An error occurred while identifying a container.

Action: There will be other messages on the error stack that show details of the problem.

ORA-45283: error writing to container
Cause: An I/O error occurred while writing to a container.

Action: There will be other messages on the error stack that show details of the problem.

ORA-45284: error reading from container
Cause: An I/O error occurred while reading from a container.

Action: There will be other messages on the error stack that show details of the problem.

ORA-45285: Cannot create more than *string* container groups.
Cause: An attempt to add a new container group would cause the number of container groups to exceed the system maximum.

Action: Increase the container group limit.

ORA-45286: Cannot create more than *string* containers.
Cause: An attempt to add a new container would cause the number of containers to exceed the system maximum.

Action: Increase the container limit.

ORA-45287: File name *string* is not valid for creation.
Cause: An attempt was made to create a contained file name in an invalid format.

Action: Correct the name and retry the operation.

ORA-45289: Cannot reserve *string* bytes in container group *string*.
Cause: The specified container group was out of space.

Action: Add another container to the container group.

ORA-45290: Cannot shrink file *string* because file is busy.
Cause: The specified file cannot be reduced in size because some other process was holding the file open.

Action: Wait until the other process releases the file before attempting to reduce its size.

ORA-45291: Container *string* is not globally identified.
Cause: An attempt to create or identify a file failed because a required container is not globally identified in this instance. The logs from the GEN0 process will usually indicate the reason why the file could not be identified.

Action: Examine the logs from the GEN0 process and correct the problem that is making some containers inaccessible.

ORA-45292: error during container group rebuild
Cause: An unrecoverable error occurred during container group rebuild.

Action: Review other messages on the error stack for additional details.

ORA-45293: Cannot shrink file
Cause: An attempt to shrink file was requested but this has been prevented because the file has been marked as not shrinkable.

Action: Nothing. File shrink is aborted.

ORA-64700: Recovery Appliance is shutting down
Cause: The Recovery Appliance was in the process of shutting down. This message is recorded in the incident log for the Recovery Appliance. When the shutdown completes, the incident is marked as `FIXED`.

Action: Wait for the Recovery Appliance to complete its shutdown.

ORA-64701: storage location *string* can no longer honor its reservations.
Cause: The specified storage location did not contain enough space to fulfill the reservations of all of the databases assigned to it. This error will be seen if a storage location lost part of its disk space and was in the process of being repaired.

Action: Either shrink the reservations for the databases contained within the storage location or add additional disk space to the storage location.

ORA-64702: error repairing container files for storage location *string*: *string*
Cause: An error was returned while trying to rebuild or repair the container files used to store data from protected databases.

Action: The subsequent error will identify the error that needs to be addressed before the Recovery Appliance can be repaired.

ORA-64703: resource error detected
Cause: A task needed to be interrupted because it detected a resource limitation such as insufficient temporary table space or a snapshot being too old. It will be retried once the contention for the resource decreases. The secondary messages will identify the resource that has been exhausted.

Action: If this error occurs rarely, no user action is required. If the condition becomes persistent, the Recovery Appliance administrator should increase the resource that is exhausted.

ORA-64705: no destination in "*string*" at column *string*
Cause: The storage destination contained a syntax error.

Action: Correct the syntax error and retry the operation.

ORA-64708: more than one polling_location in "*string*" at column *string*
Cause: More than one `polling_location` was specified. Only one `polling_location` is allowed.

Action: Specify only one polling destination directory and retry the operation.

ORA-64709: ASM polling_location is not supported in "*string*" at column *string*
Cause: The `polling_location` specified an ASM-based location. Only non-ASM-based `polling_locations` are supported.

Action: Specify one non-ASM polling destination and retry the operation.

ORA-64711: storage destination do not reference an ASM diskgroup
Cause: The storage destination referenced a non-ASM storage location. Only ASM-based storage destinations are allowed. The operation has been rolled back.

Action: Correct the syntax error and retry the operation.

ORA-64713: requested size *string* for *string* was too small; already using *string*
Cause: The size requested for the storage destination was smaller than its current size.

Action: Increase the requested size and retry the operation.

ORA-64714: requested size *string* for *string* was larger than total available space *string*
Cause: The size requested for the storage destination was larger than its current used space plus its current free space.

Action: Decrease the requested size and retry the operation.

ORA-64715: instance *string* is not available to Recovery Appliance
Cause: The Recovery Appliance is not utilizing the specified instance due to its absence from the Oracle RAC.

Action: Restart the specified instance or repair any connectivity issues with the specified instance.

ORA-64716: storage location *string* allocation size *string* does not equal diskgroup *string* allocation size *string*
Cause: The minimum allocation size of the specified storage location was not the same as the specified diskgroup allocation unit size.

Action: Specify a different diskgroup or a different storage location and retry the operation.

ORA-64717: network chunk size *string* is not a multiple of diskgroup *string* allocation size *string*
Cause: The configured network chunk size was not a multiple of the specified diskgroup allocation unit size.

Action: Specify a different diskgroup or reconfigure the network chunk size and retry the operation.

ORA-64718: diskgroup *string* allocation size *string* is not a power of two
Cause: The diskgroup allocation unit size was not a power of two.

Action: Specify a different diskgroup and retry the operation.

ORA-64719: diskgroup *string* allocation size *string* is less than two megabytes *string*
Cause: The diskgroup allocation unit size was less than two megabytes.

Action: Specify a different diskgroup and retry the operation.

ORA-64720: no containers were created for storage location *string*
Cause: No storage was allocated and initialized for the specified storage location.

Action: Specify a different diskgroup for the storage location, increase the size of the diskgroup, or reduce the size of the storage location and retry the operation.

ORA-64721: reserved space *string* is less than the minimum reservation *string*
Cause: No storage was allocated and initialized for the specified storage location.

Action: Specify a different diskgroup for the storage location, increase the size of the diskgroup, or reduce the size of the storage location and retry the operation.

ORA-64722: number of drives must be greater than zero
Cause: The specified number of tape drives was NULL or was less than or equal to zero.

Action: Specify a number of tape drives greater than 0 and retry the operation.

ORA-64723: number of drives reserved for restore operations must be greater than or equal to zero
Cause: The number of tape drives reserved for restore operations was NULL or was less than zero.

Action: Specify a number of tape drives reserved for restore operations greater than or equal to zero and retry the operation.

ORA-64724: number of restore drives *string* too large; must be less than *string*
Cause: The number of tape drives reserved for restore operations was at least as large as the total number of drives available. The number of tape drives reserved for restore operations must be at least one less than the total number of drives available.

Action: Specify a number of tape drives reserved for restore operations less than the total number of drives available and retry the operation.

ORA-64725: number of streams must be greater than zero
Cause: The number of streams was less than or equal to zero.

Action: Specify a number of streams greater than zero and retry the operation.

ORA-64726: number of streams *string* too large; must be no larger than *string*
Cause: The number of streams was larger than the total number of drives available. The number of streams must be no larger than the total number of drives available.

Action: Reduce the number of available streams and retry the operation.

ORA-64727: number of copies *string* not in the range 1 through 4
Cause: The number of copies was either NULL or not in the range 1 through 4 inclusive.

Action: Specify a number of copies in the range 1 through 4 and retry the operation.

ORA-64728: replication server name length *string* is too long
Cause: The replication server name was longer than 128 characters.

Action: Specify a replication server name shorter than 128 characters and retry the operation.

ORA-64729: replication server proxy port *string* must be greater than zero
Cause: The replication server proxy port number was less than or equal to zero.

Action: Specify a replication server proxy port number greater than zero and retry the operation.

ORA-64730: replication server proxy URL provided but proxy port is NULL
Cause: A replication server proxy URL was provided but a proxy port number was not. If either a proxy URL or a proxy port are specified, both must be specified.

Action: Specify both a replication server URL and a replication server proxy port number and retry the operation.

ORA-64731: replication server proxy port provided but proxy URL is NULL
Cause: A replication server proxy port number was provided but a proxy URL was not. If either a proxy URL or a proxy port are specified, both must be specified.

Action: Specify both a replication server URL and a replication server proxy port number and retry the operation.

ORA-64732: HTTP server not configured at replication host
Cause: The HTTP server at the replication host site has not been configured.

Action: Configure the HTTP server at the replication host site and retry the operation.

ORA-64733: unable to move individual backup piece with BP_KEY *string*; not a KEEP backup
Cause: An attempt was made to move an individual backup piece, but the backup set of which this backup piece was a member was not a KEEP backup.

Action: Specify a backup piece key that is a member of a KEEP backup set and retry the operation.

ORA-64735: unknown incarnation detected at Recovery Appliance, need catalog resync
Cause: A new archived log or backup set belonging to the new incarnation was received at Recovery Appliance.

Action: Using RMAN, connect to the Recovery Appliance as a recovery catalog, primary database as target database and perform the resynchronization operation using the `RESYNC CATALOG` RMAN command. If this error occurred at downstream of the Recovery Appliance (in a replicated Recovery Appliance setup), the reconcile operation fixes this error automatically when the same error is fixed at the upstream Recovery Appliance.

ORA-64736: Task ID *string* of type *string* has been interrupted *string* times.
Cause: The specified task was restarted an unexpected number of times. Tasks get interrupted when there is competition for resources. This is only a warning. It does not necessarily indicate a problem with the Recovery Appliance.

Action: If these problems persist for long periods, contact Oracle Support Services.

ORA-64737: unable to copy a full backup for database *string* because of missing data files
Cause: While creating a full database backup to tape or to a replicated Recovery Appliance, level 0 backups of one or more data files were missing.

Action: Query the `RA_SBT_TEMPLATE_MDF` view to determine the data files for which backups are missing. If using the "incremental forever" backup strategy, perform a level-0 incremental backup for the given database and retry the operation.

ORA-64738: guaranteed copy suspended for database *string*
Cause: One of the following operations was performed resulting in the database using more than its allotted disk space:
- `UPDATE_DB` lowering the `DISK_SPACE_RESERVE` value.
- `UPDATE_PROTECTION_POLICY` setting the `guaranteed_copy` parameter to `YES`
- `DELETE_SBT_LIBRARY` where backup data for the given database existed. New backup requests may be stalled until the system can recompute the safety of allowing additional backup data or backup data may be lost.

Action: This event can be avoided by ensuring backups are written to tape in a timely manner. Conversely, one should avoid the activities listed in the Cause statement when backups are not being written to tape in a timely manner.

ORA-64740: Backups from database *string* have not been seen for more than UNPROTECTED_WINDOW period
Cause: An `UNPROTECTED_WINDOW` parameter in protection policy has been specified and the Recovery Appliance has not received sufficient archive log backups or data file backups from the given target database for at least that period.

Action: Ensure that backups are being performed in a timely manner and that, if set up, redo logs or backups are being sent to the Recovery Appliance.

ORA-64741: Scheduler *string* running task *string* of type *string* did not stop after *string* requests.
Cause: The specified Recovery Appliance scheduler process could not be stopped.

Action: If these problems persist for long periods, contact Oracle Support Services.

ORA-64742: database *string* is being deleted
Cause: The current command failed because the database was in the process of being deleted.

Action: No action is necessary.

ORA-64744: Argument *string* is null, invalid, or out of range.
Cause: The argument was expecting a non-null, valid value but the argument value passed in was null, invalid, or out of range.

Action: Check your program and correct the caller of the routine to not pass a null, invalid or out-of-range argument value.

ORA-64745: Name length is *string* characters; maximum length is *string* characters.
Cause: The length of the name exceeded the limit.

Action: Specify a shorter name and retry the operation.

ORA-64746: Name contains invalid characters.
Cause: The name incorrectly started with "_", "-", ":" or digits or contained non-alphanumeric characters. Verify that all other double quotation marks, if any, in the string are adjacent pairs of double quotation marks. Double quotation marks must not be used in the middle of the name.

Action: Change the name and exclude the invalid characters.

ORA-64747: Name contains invalid character "*string*" at the position [*string*].
Cause: The name incorrectly started with "_", "-", ":" or digits or contained non-alphanumeric characters. Verify that all other double quotation marks, if any, in the string are adjacent pairs of double quotation marks. Double quotation marks must not be used in the middle of the name.

Action: Change the name and exclude the invalid characters.

ORA-64748: trace file writing initiated using *string*
Cause: The configuration of the Recovery Appliance was modified to enable the producing of trace files. Trace files have the capacity to exhaust disk space on the Recovery Appliance.

Action: Turn off the tracing when it is no longer required.

ORA-64750: Instance *string* is unable to access *string*.
Cause: The Recovery Appliance was unable to find a file that is required for its operation.

Action: Ensure that the file system of the specified file is available on the specified instance.

ORA-64751: Replication setup error during *string*. replication server:*string* database:*string*.
Cause: The Recovery Appliance was unable to complete the configuration and setup of replication for the database specified.

Action: Validate that the downstream replication server is properly configured and all network communication paths are valid.

ORA-64752: storage unavailable for new redo or backups for database *string*
Cause: There was a failure while backing up redo or copying backups from a polling location. This condition may be due to one of the following reasons:
* An individual backup piece exceeded the database protection policy `DISK_RESERVE_SPACE` value. * Guaranteed_copy is enabled but not enough data has been spooled to tape. * Misconfiguration of the storage location size.

Action: Check for the value of `DISK_RESERVED_SPACE` and storage location uses.

ORA-64753: Incorrect object type specified; specified *string*, expected *string*
Cause: An incorrect object type was given to an API command.

Action: Use the object-specific API. For example, use 'resume_replication_server' instead of 'resume_sbt_library'.

ORA-64754: unable to perform operation with associated tape or replication objects
Cause: An attempt to execute 'update_db' or 'update_protection_policy' and change storage locations with a replication server or tape job associated with the protection policy failed.

Action: Create a temporary protection policy that has the same storage location as the current protection policy with the tape and replication attributes of the target protection policy, Update to the temporary protection policy then finally update to the target protection policy.

ORA-64755: failed to delete database *string*; the Recovery Appliance is not running
Cause: An attempt was made to delete a database, but the Recovery Appliance has been deactivated by the administrator.

Action: Have the Recovery Appliance administrator execute DBMS_RA.STARTUP_RECOVERY_APPLIANCE and retry the delete_db() operation.

ORA-64757: unable to restore backup piece with BP_KEY *string*
Cause: An attempt was made to restore the specified backup piece which resulted in an error. An explanation of the error appears in the following messages.

Action: Verify the correctness of the backup piece. If it is corrupt, either delete it or replace it.

ORA-64758: unable to grow delta store metadata in tablespace *string*
Cause: Additional extents could not be allocated for the tables used to implement the delta store.

Action: Add additional storage to the indicated tablespace.

ORA-64759: Recovery Appliance is leaving restricted resources state
Cause: The Recovery Appliance ended its restrictions on task execution. The restricted resources state was entered when tasks could not run due to insufficient temporary table space or insufficient undo space. At that time, resource intensive tasks were put into RESOURCE_WAIT state.

Action: None. This is only an informational message entered in the alert log.

ORA-64760: Database *string* has had tasks in ordering wait state for over *string* days.
Cause: The specified database had an INDEX_BACKUP task that could not be run because the task did not tile against the delta store. An incremental backup piece will not tile into the delta store when the full backup that it was built upon is not found in the delta store.

Action: Provide the missing incremental backups for the database. If no other incremental backups can be found, provide a new full backup for the database.

ORA-64761: disk group *string* is not usable by the Recovery Appliance
Cause: A disk group was supplied to either the `create_storage_location` or `update_storage_location` APIs that was not previously prepared by the installation software for the Recovery Appliance.

Action: Run the `ra_update` procedure to process the disk group and retry the API.

ORA-64762: Task *string* of type *string* has been running for *string*.
Cause: The specified task did not complete its execution after a reasonable period. This is only a warning message.

Action: If these errors persist, contact Oracle Support Services.

ORA-64763: Task *string* of type *string* was terminated after running for *string*.
Cause: The specified task did not complete and was presumed hung. Its process was stopped and restarted.

Action: If these errors persist, contact Oracle Support Services.

ORA-64766: backup deletion using RMAN prevented by protection policy
Cause: Recovery Manager was prevented from deleting a backup piece because the 'allow_backup_deletion' parameter of the applicable Recovery Appliance protection policy was 'NO'.

Action: Modify the 'allow_backup_deletion' parameter of the applicable protection policy to 'YES' to allow for deletion of backups.

ORA-64767: restore timed out
Cause: The Recovery Appliance terminated the restore operation due to unresponsiveness of the client database.

Action: Verify the network between the client database and the Recovery Appliance. Also verify the client database I/O performance. If the error persists, contact Oracle Support Services.

ORA-64768: KEEP file size *string* cannot exceed available DISK_RESERVED_SPACE *string*
Cause: There was no more space for a `KEEP` backup piece. The total `DISK_RESERVED_SPACE` minus the space currently consumed by `KEEP` backups was less than the size of the current `KEEP` backup piece. The piece that caused the error was deleted.

Action: Use DBMS_RA.UPDATE_DB to increase the DISK_RESERVED_SPACE value of the database or move KEEP (archival) backups to other storage using the DBMS_RA.MOVE_BACKUP function.

ORA-65000: missing or invalid pluggable database name
Cause: A valid pluggable database name was not present where required.

Action: Reissue the statement with a valid pluggable database name.

ORA-65001: missing or invalid administrative user name
Cause: A valid administrative user name was not present where required by the syntax of `CREATE PLUGGABLE DATABASE` statement.

Action: Reissue the CREATE PLUGGABLE DATABASE statement with a valid administrative user name.

ORA-65002: missing or invalid administrative user password
Cause: A valid administrative user password was not present where required by the syntax of CREATE PLUGGABLE DATABASE statement.

Action: Reissue the CREATE PLUGGABLE DATABASE statement with a valid administrative user password.

ORA-65003: missing or invalid XML file name
Cause: A valid XML file name was not present where required by the syntax of CREATE PLUGGABLE DATABASE statement.

Action: Reissue the CREATE PLUGGABLE DATABASE statement with a valid XML file name.

ORA-65004: missing or invalid database link name
Cause: A valid database link name was not present where required by the syntax of CREATE PLUGGABLE DATABASE statement.

Action: Reissue the CREATE PLUGGABLE DATABASE statement with a valid database link name.

ORA-65005: missing or invalid file name pattern for file - *string*
Cause: Either source or replacement file name pattern was missing or invalid in a SOURCE_FILE_NAME_CONVERT or FILE_NAME_CONVERT clause.

Action: Correct the SOURCE_FILE_NAME_CONVERT or FILE_NAME_CONVERT clause and reissue the statement.

ORA-65006: missing or invalid ENABLE PLUGGABLE DATABASE clause
Cause: An ENABLE PLUGGABLE DATABASE clause was not present where required by the syntax of CREATE DATABASE statement.

Action: Reissue the CREATE DATABASE statement with a valid ENABLE PLUGGABLE DATABASE clause.

ORA-65007: duplicate ENABLE PLUGGABLE DATABASE clause
Cause: A duplicate ENABLE PLUGGABLE DATABASE clause was specified in a CREATE DATABASE statement.

Action: Reissue the CREATE DATABASE statement with a valid ENABLE PLUGGABLE DATABASE clause.

ORA-65008: missing or invalid SEED clause
Cause: A SEED clause was not present where required by the syntax of CREATE DATABASE statement.

Action: Reissue the CREATE DATABASE statement with a valid SEED clause.

ORA-65010: maximum number of pluggable databases created
Cause: User attempted to create more than supported number of pluggable databases.

Action: Avoid creating too many pluggable databases.

ORA-65011: Pluggable database *string* does not exist.
Cause: User attempted to specify a pluggable database that does not exist.

Action: Check `DBA_PDBS` to see if it exists.

ORA-65012: Pluggable database *string* already exists.
Cause: User attempted to create a pluggable database with a name that already exists.

Action: Check `DBA_PDBS` to see if the name exists.

ORA-65013: invalid CONTAINER clause
Cause: An invalid `CONTAINER` clause was encountered.

Action: Reissue the DDL statement with a valid `CONTAINER` clause.

ORA-65014: invalid SHARING clause
Cause: An invalid `SHARING` clause was encountered.

Action: Reissue the DDL statement with a valid `SHARING` clause.

ORA-65015: missing or invalid container name
Cause: A valid container name was not present where required by the syntax of `ALTER SESSION SET CONTAINER` or `ALTER USER ... SET|ADD|REMOVE CONTAINER_DATA` statement.

Action: Reissue the statement with a valid container name.

ORA-65016: FILE_NAME_CONVERT must be specified
Cause: Data files, and possibly other files, needed to be copied as a part of creating a pluggable database. However, Oracle Managed Files (OMF) was not enabled, `PDB_FILE_NAME_CONVERT` was not defined, and there was a failure to specify the `FILE_NAME_CONVERT` clause.

Action: Enable OMF or define `PDB_FILE_NAME_CONVERT` system parameter before issuing `CREATE PLUGGABLE DATABASE` statement, or specify `FILE_NAME_CONVERT` clause as a part of the statement.

ORA-65017: seed pluggable database may not be dropped or altered
Cause: User attempted to drop or alter the Seed pluggable database which is not allowed.

Action: Specify a legal pluggable database name.

ORA-65018: FILE_NAME_CONVERT or NOCOPY must be specified
Cause: Oracle Managed Files (OMF) was not enabled and `PDB_FILE_NAME_CONVERT` was not defined. The `FILE_NAME_CONVERT` or the `NOCOPY` clause was not specified as a part of creating a pluggable database using data files.

Action: Enable OMF or define `PDB_FILE_NAME_CONVERT` system parameter before issuing `CREATE PLUGGABLE DATABASE` statement, or specify `FILE_NAME_CONVERT` clause or `NOCOPY` as a part of the statement.

ORA-65019: pluggable database *string* already open
Cause: An attempt was made to open a pluggable database that was already opened.

Action: Check the `OPEN_MODE` column in `V$PDBS` view.

ORA-65020: pluggable database *string* already closed
Cause: An attempt was made to close a pluggable database that was already closed.

Action: Check the `OPEN_MODE` column in `V$PDBS` view.

ORA-65021: illegal use of SHARING clause
Cause: A `SHARING` clause was encountered in unexpected context.

Action: Do not use `SHARING` clause outside of Oracle-supplied scripts.

ORA-65022: CONTAINER clause already specified
Cause: A statement contained multiple `CONTAINER` clauses.

Action: Eliminate redundant `CONTAINER` clauses.

ORA-65023: active transaction exists in container *string*
Cause: A statement attempted to create a new transaction in the current container while there was an active transaction in another container.

Action: Switch to the container with the active transaction and commit, rollback or detach the active transaction before attempting to issue any statement that will attempt to create a new transaction in another container.

ORA-65024: Pluggable database *string* is not open.
Cause: An operation was attempted on a pluggable database that was not open.

Action: Open the pluggable database using appropriate open mode.

ORA-65025: Pluggable database *string* is not closed on all instances.
Cause: An operation was attempted on a pluggable database that was not closed on all Oracle RAC instances.

Action: Close the pluggable database on all instances and retry the operation.

ORA-65026: XML metadata file error : *string*
Cause: An error occurred while trying to parse or write to the XML metadata file.

Action: Check that the XML metadata file exists and is readable.

ORA-65027: XML metadata file error while getting node or value for (*string* - *string*)
Cause: An error occurred while trying to parse the XML metadata file.

Action: Check and correct the XML metadata file.

ORA-65028: Unable to open plugin data file at path *string*
Cause: Error occurred while trying to open the datafile.

Action: Check that the datafile exists at the path.

ORA-65029: a Local User may not grant or revoke a Common Privilege or Role
Cause: A Local User issued a `GRANT` or `REVOKE` statement specifying `CONTAINER=ALL`, which is illegal.

Action: Remove `CONTAINER=ALL` from the statement.

ORA-65030: one may not grant a Common Privilege to a Local User or Role
Cause: A Common User issued a `GRANT` statement specifying `CONTAINER=ALL` and naming a Local User or Role as a grantee, which is illegal.

Action: If trying to grant a Local Privilege, remove `CONTAINER=ALL` from the statement. If trying to grant a Common Privilege, remove Local Users and Roles from the list of grantees.

ORA-65031: one may not revoke a Common Privilege from a Local User or Role
Cause: A Common User issued a `REVOKE` statement specifying `CONTAINER=ALL` and naming a Local User or Role as a grantee, which is illegal.

Action: If trying to revoke a Local Privilege, remove `CONTAINER=ALL` from the statement. If trying to revoke a Common Privilege, remove Local Users and Roles from the list of grantees.

ORA-65032: a Local Role may only be granted or revoked within the current Container
Cause: A user issued a `GRANT` or `REVOKE` statement specifying `CONTAINER=ALL` and listing a Local Role among roles to be granted or revoked, which is illegal.

Action: If trying to revoke a Local Role, remove `CONTAINER=ALL` from the statement. If trying to revoke Common Privileges and/or Roles, remove Local Roles from the list of roles being granted or revoked.

ORA-65033: a common privilege may not be granted or revoked on a local object
Cause: A `GRANT` or `REVOKE` statement was issued specifying `CONTAINER=ALL` and naming a local object on which privileges are to be granted or revoked, which is illegal.

Action: Remove `CONTAINER=ALL` from the statement.

ORA-65034: PDB describe output file not specified
Cause: User attempted to describe a pluggable database without specifying an output XML file location.

Action: Specify an output file location for describe

ORA-65035: unable to create pluggable database from *string*
Cause: The pluggable database had ongoing or active transactions that need to be recovered.

Action: Open the pluggable database in read/write mode before cloning again so that transaction recovery can be performed.

ORA-65036: pluggable database *string* not open in required mode
Cause: Attempted to perform an operation on a pluggable database in incorrect open mode.

Action: Open the pluggable database in the mode required for this operation

ORA-65037: a common privilege may not be granted or revoked on a local user
Cause: A `GRANT` or `REVOKE` statement was issued specifying `CONTAINER=ALL` and naming a local user on which privileges are to be granted or revoked, which is illegal.

Action: Remove `CONTAINER=ALL` from the statement.

ORA-65039: container identifier column missing or is of unexpected type in a definition of a CONTAINER_DATA object
Cause: Table or view whose definition contained a `CONTAINER_DATA` clause lacked a column used to identify a container to which data belongs or the column was of unexpected type.

Action: Correct the statement and reenter.

ORA-65040: operation not allowed from within a pluggable database
Cause: An operation was attempted that can only be performed in the root container.

Action: Switch to the root container to perform the operation.

ORA-65041: CONTAINER_DATA attribute for this user cannot be modified
Cause: An attempt was made to modify `CONTAINER_DATA` attribute for user `SYS` or `SYSBACKUP` which is disallowed.

Action: Do not attempt to modify `CONTAINER_DATA` attribute for users `SYS` or `SYSBACKUP`.

ORA-65042: name is already used by an existing container
Cause: The name was already used by another container.

Action: Specify a valid name.

ORA-65043: TABLESPACE keyword expected
Cause: `TABLESPACE` keyword was missing.

Action: Specify a `TABLESPACE` keyword.

ORA-65044: missing or invalid option following STORAGE keyword
Cause: An option other than `MAXSIZE` or `MAX_SHARED_TEMP_SIZE` was specified.

Action: Specify only legal options.

ORA-65045: pluggable database not in a restricted mode
Cause: An operation was attempted on a pluggable database that was not in restricted mode.

Action: Open the pluggable database in a restricted mode.

ORA-65046: operation not allowed from outside a pluggable database
Cause: An operation was attempted that can only be performed from within a pluggable database.

Action: Switch to a pluggable database to perform the operation.

ORA-65047: object *string*.*string* is invalid or compiled with errors in CDB$ROOT
Cause: An attempt was made to issue a metadata link DDL for an object that was invalid or compiled with errors in `CDB$ROOT`.

Action: Check the validity of the object in `CDB$ROOT`.

ORA-65048: error encountered when processing the current DDL statement in pluggable database *string*
Cause: An error was encountered when executing a statement in one of the pluggable databases.

Action: Examine the cause of failure in the pluggable database.

ORA-65049: creation of local user or role is not allowed in CDB$ROOT
Cause: An attempt was made to create a local user or role in `CDB$ROOT`.

Action: If trying to create a common user or role, specify `CONTAINER=ALL`.

ORA-65050: Common DDLs only allowed in CDB$ROOT
Cause: An attempt was made to issue a Common DDL in a pluggable database.

Action: Switch to `CDB$ROOT` and issue the Common DDL there.

ORA-65051: missing valid container identifier
Cause: A valid container identifier was not specified.

Action: Specify a valid container identifier.

ORA-65052: statement involves operations with different container scope
Cause: An attempt was made to combine one operation that applies to all containers with another that only applies to the local container into one statement.

Action: Execute the operations in separate statements.

ORA-65053: A global user cannot change the container in the session.
Cause: An attempt was made by a global user to change the container using the `ALTER SESSION SET CONTAINER` statement.

Action: This operation is not allowed for global users.

ORA-65054: Cannot open a pluggable database in the desired mode.
Cause: An attempt was made to open a pluggable database in a mode incompatible with that of the CDB.

Action: Open the CDB in a compatible mode first and retry the operation.

ORA-65056: CONTAINER_DATA attribute is not used in a pluggable database.
Cause: User connected to a pluggable database attempted to modify a `CONTAINER_DATA` attribute which is illegal.

Action: User connected to a pluggable database may only see rows of `CONTAINER_DATA` objects on which he has been granted appropriate privilege and which pertain to that pluggable database or to the CDB as a whole. Ability to see this data is not controlled by `CONTAINER_DATA` attribute, which is only used to control ability to see rows pertaining to certain Containers while connected to `CDB$ROOT` in a CDB.

ORA-65057: CONTAINER_DATA attribute must always include the current container
Cause: User issuing `ALTER USER ... SET|REMOVE CONTAINER_DATA = ...` statement attempted to exclude the current container from a `CONTAINER_DATA` attribute.

Action: Ensure that a `CONTAINER_DATA` attribute always includes the current container.

ORA-65058: object-specific CONTAINER_DATA attribute may only be specified for a CONTAINER_DATA object
Cause: Object referenced in an `ALTER USER` statement modifying an object-specific `CONTAINER_DATA` attribute is not a `CONTAINER_DATA` table or view.

Action: Ensure that an object referenced in the statement is a `CONTAINER_DATA` table or view.

ORA-65059: duplicate container name in CONTAINER_DATA clause
Cause: Duplicate references to a name of some container was encountered in a `CONTAINER_DATA` clause.

Action: Ensure that no container name appears more than once.

ORA-65060: CONTAINER_DATA attribute is not set
Cause: User attempted to add container(s) to an object-specific `CONTAINER_DATA` attribute which has not been explicitly set or has been set to `DEFAULT` or remove containers from a `CONTAINER_DATA` attribute which has not been explicitly set or has been set to `DEFAULT`.

Action: Avoid adding containers to object-specific `CONTAINER_DATA` attribute which has not been explicitly set or has been set to `DEFAULT` or removing containers from a `CONTAINER_DATA` attribute which has not been explicitly set or has been set to `DEFAULT`.

ORA-65061: some of specified containers do not belong to the CONTAINER_DATA attribute
Cause: User attempted to remove from a `CONTAINER_DATA` attribute container(s) which do not belong to it.

Action: Do not attempt to remove from a `CONTAINER_DATA` attribute container(s) which do not belong to it.

ORA-65062: CONTAINER_DATA attribute is set to ALL
Cause: User attempted to add/remove container(s) to/from a `CONTAINER_DATA` attribute whose current value is ALL, which is not supported.

Action: Do not attempt to add/remove container(s) to/from a `CONTAINER_DATA` attribute whose current value is ALL.

ORA-65063: CONTAINER_DATA clause has already been specified
Cause: The `CONTAINER_DATA` clause was specified twice.

Action: Specify only one `CONTAINER_DATA` clause.

ORA-65064: incorrect contents of XML metadata file
Cause: The contents of the XML metadata file were different from the actual file properties.

Action: Check and correct the XML metadata file.

ORA-65065: A local user or role can only be altered within the current container
Cause: The `ALTER USER` or `ALTER ROLE` statement was issued specifying `CONTAINER=ALL` and listing a local user or role among the users or roles to be altered, which is illegal.

Action: If trying to alter a local user or role, specify `CONTAINER=CURRENT`. If trying to alter common users or roles, remove local users or roles from the list of roles being altered.

ORA-65066: The specified changes must apply to all containers
Cause: An attempt was made to apply the specified changes to the current container.

Action: Specify `CONTAINER=ALL`.

ORA-65067: DEFAULT ROLE clause referencing a local role can only apply to the current container
Cause: An attempt was made to reference a local role across all containers using the `DEFAULT ROLE` clause.

Action: If you are trying to set a local role as the default role, specify `CONTAINER=CURRENT`. If trying to set common roles as the default roles, remove local roles from the list of roles referenced in the `DEFAULT ROLE` clause.

ORA-65068: cannot define a trigger that fires after a pluggable database is unplugged
Cause: An attempt was made to create a trigger that fires after a pluggable database has been unplugged. This type of trigger is not supported.

Action: Do not attempt to create a trigger that fires after a pluggable database has been unplugged.

ORA-65069: AFTER DB_ROLE_CHANGE triggers cannot be defined on a pluggable database
Cause: An attempt was made to create a pluggable database trigger that fires after a role change occurs from a standby database to primary or vice versa. This type of trigger is not supported.

Action: Do not specify `AFTER DB_ROLE_CHANGE` when creating a trigger on a pluggable database.

ORA-65070: AFTER CLONE trigger can only be created on a pluggable database
Cause: An attempt was made to create an `AFTER CLONE` trigger on a schema or a database. This type of trigger is not supported.

Action: Do not specify `AFTER CLONE` when creating a trigger on a schema or a database.

ORA-65071: BEFORE UNPLUG trigger can only be created on a pluggable database
Cause: An attempt was made to create a `BEFORE UNPLUG` trigger on a schema or a database. This type of triggers is not supported.

Action: Do not specify `BEFORE UNPLUG` when creating a trigger on a schema or a database.

ORA-65072: user must be connected to a pluggable database on which a trigger is being created
Cause: An attempt was made to create a database event trigger on a pluggable database while not connected to a pluggable database. This is not supported.

Action: Connect to the pluggable database on which a database event trigger needs to be created before attempting to create such a trigger.

ORA-65073: cannot define a trigger that fires before a pluggable database is cloned
Cause: An attempt was made to create a trigger that fires before a pluggable database has been cloned. This type of trigger is not supported.

Action: Do not attempt to create a trigger that fires before a pluggable database has been cloned.

ORA-65074: editions not supported for common users
Cause: An attempt was made to support editions for common users.

Action: Do not attempt to enable editions for common users.

ORA-65080: cannot determine pluggable database name
Cause: An attempt was made to map the database ID to the pluggable database name.

Action: Contact Oracle Support Services.

ORA-65081: database or pluggable database is not open in read only mode
Cause: An operation was attempted on a database or pluggable database that is not open in read only mode

Action: Open the database or pluggable database in read only mode and then retry the operation.

ORA-65082: cannot add any more pluggable databases: limit of *string* exceeded
Cause: There was no more room in the control file for adding pluggable databases.

Action: Recreate the control file.

ORA-65083: pluggable database (PDB) shutdown in progress
Cause: The pluggable database is in the middle of shutdown abort

Action: Retry the operation later.

ORA-65084: object *string.string* does not exist in root
Cause: An attempt was made to create a common object that does not exist in root.

Action: Invoke the script using catcon.pl to create the object in all containers.

ORA-65085: cannot open pluggable database in read-only mode
Cause: The pluggable database has been created and not opened.

Action: The pluggable database needs to be opened in read/write or restricted mode first.

ORA-65086: cannot open/close the pluggable database
Cause: The pluggable database has been unplugged.

Action: The pluggable database can only be dropped.

ORA-65087: Oracle-supplied operation not allowed from within a pluggable database
Cause: An operation in an Oracle-supplied script was attempted that can only be performed in the root container.

Action: Confirm that the operation is needed in the root container, and switch to the root container to perform the operation.

ORA-65088: database open should be retried
Cause: An inconsistency between the control file and the data dictionary was found and fixed during the database open. The database open needs to be executed again.

Action: Retry the database open.

ORA-65089: pluggable database is not clean
Cause: The pluggable database was not open anywhere but was not marked as clean yet.

Action: Wait until the cleanup is done and retry.

ORA-65090: operation only allowed in a container database
Cause: User attempted an operation that is only allowed in a CDB

Action: Connect to a CDB to perform this operation

ORA-65091: operation on *string* not allowed in a pluggable database
Cause: An undo tablespace or rollback segment operation was attempted that can only be performed in the root container.

Action: Switch to the root container to perform the undo tablespace or rollback segment operation.

ORA-65092: system privilege granted with a different scope to '*string*'
Cause: An attempt to revoke a system privilege that was granted with a different scope has been made. The user tried to either revoke a common privilege in the current container or to revoke a local privilege in a container database (CDB).

Action: Specify the correct value for the `CONTAINER` clause.

ORA-65093: multitenant container database not set up properly
Cause: An attempt was made to open a multitenant container database without the correct parameter set for a multitenant container database in the initialization parameter file.

Action: Set the '`enable_pluggable_database=true`' parameter for the multitenant container database in the initialization parameter file and restart the database.

ORA-65094: invalid local user or role name
Cause: An attempt was made to create a local user or role with a name that was not valid for local users or roles. In addition to the usual rules for user and role names, local user and role names cannot start with `C##` or `c##`.

Action: Specify a valid local user or role name.

ORA-65095: invalid common object name
Cause: An attempt was made to create a common object with a name that was not valid for common objects. In addition to the usual rules for object names, common object names must consist only of ASCII characters.

Action: Specify a valid common object name.

ORA-65096: invalid common user or role name
Cause: An attempt was made to create a common user or role with a name that was not valid for common users or roles. In addition to the usual rules for user and role names, common user and role names must start with C## or c## and consist only of ASCII characters.

Action: Specify a valid common user or role name.

ORA-65097: Invalid argument supplied to CDB$VIEW function
Cause: `CDB$VIEW` may be applied to a valid table name of the form <owner>.<tablename>

Action: Correct the statement

ORA-65098: Datatype not supported with CDB$VIEW function
Cause: Column with unsupported datatype was selected from a CDB View.

Action: Remove the column from the select list

ORA-65099: Operation cannot be performed when the CDB is not open
Cause: An operation was attempted inside a pluggable database that requires the container database (CDB) to be open. *Action: Open the CDB and then reissue the operation.

Action: None

ORA-65100: missing or invalid path prefix - *string*
Cause: Path prefix was missing or invalid.

Action: Correct the `PATH_PREFIX` clause and reissue the statement.

ORA-65101: container database set up incorrectly
Cause: An attempt was made to use a non container database control file to startup a container database (CDB).

Action: Create a new control file for the CDB.

ORA-65102: missing or invalid instance name
Cause: A valid instance name was not present where required by the syntax of a `ALTER PLUGGABLE DATABASE` statement.

Action: Reissue the statement with a valid instance name.

ORA-65103: UPGRADE cannot be specified for PDBs being open in READ ONLY mode
Cause: `ALTER PLUGGABLE DATABASE ... OPEN UPGRADE` was specified, but the root is open `READ ONLY`, so the specified PDBs will also be opened `READ ONLY`, and `UPGRADE` cannot be specified for PDBs being opened `READ ONLY`.

Action: Reissue the statement without specifying `UPGRADE` or reopen the root in `READ WRITE` mode and then reissue the statement.

ORA-65104: operation not allowed on an inactive pluggable database
Cause: The pluggable database status was `INACTIVE`. It was still being created or there was an error during the create operation.

Action: Wait until the status is changed to `CREATED` or, in case of errors, drop the pluggable database and re-create it.

ORA-65105: SYSTEM data file for pluggable database #*string* not found
Cause: Data files were not specified in a `CREATE CONTROLFILE` statement in the SYSTEM tablespace of a pluggable database.

Action: Locate the data files and resubmit the `CREATE CONTROLFILE` statement.

ORA-65106: Pluggable database #*string* (*string*) is in an invalid state.
Cause: Data files were not specified in a `CREATE CONTROLFILE` statement of a pluggable database.

Action: Drop the pluggable database or locate the data files and resubmit the `CREATE CONTROLFILE` statement.

ORA-65107: Error encountered when processing the current task on instance:*string*
Cause: An error was encountered when executing a pluggable database task on one of the Oracle RAC instances.

Action: Examine the cause of failure on the instance.

ORA-65108: invalid use of a cursor belonging to another container
Cause: An attempt was made to use a cursor that was parsed or executed in a different container.

Action: Check if there are any incorrect uses of the `SET CONTAINER` statement. Parse, execute, fetch, and close a cursor only all in the same container.

ORA-65109: operation not allowed in CDB$ROOT
Cause: An operation was attempted that is not supported in the Container Database root.

Action: Switch to a pluggable database to perform the operation.

ORA-65110: Invalid instance name specified
Cause: An attempt was made to use an invalid or inactive instance name in the instance clause to alter the state of a pluggable database.

Action: Specify a valid and active instance name.

ORA-65111: Cannot relocate to the same instance
Cause: An attempt was made to relocate a pluggable database to the current instance.

Action: Specify an instance which is different then the current instance.

ORA-65112: pluggable database *string* not closed on all instances of the standby database
Cause: Media recovery stopped because the pluggable database was either unplugged, dropped or renamed on the primary database.

Action: Close the pluggable database on all instances and restart the recovery.

ORA-65113: value of MAX_PDB_STORAGE property for the PDB is too low
Cause: The value of `MAX_PDB_STORAGE` specified in the `ALTER` or `CREATE PLUGGABLE DATABASE` statement is less than the current space usage of data files and temporary files of the container.

Action: Specify a higher value for `MAX_PDB_STORAGE`.

ORA-65114: space usage in container is too high
Cause: Space usage in the current container exceeded the value of `MAX_PDB_STORAGE` for the container.

Action: Specify a higher value for `MAX_PDB_STORAGE` using the `ALTER PLUGGABLE DATABASE` statement.

ORA-65115: CDB resource plan *string* has more than *string* PDB directives.
Cause: An attempt was made to create or update the specified multitenant container database (CDB) resource plan to use more than the supported number of directives.

Action: Remove the directives for pluggable databases (PDBs) that are not active on this database.

ORA-65116: incompatible database character set
Cause: The database character set of the container database was not a superset of the database character set of the pluggable database being plugged in.

Action: Plug the pluggable database into a container database having a compatible database character set.

ORA-65117: CONTAINER clause may only be specified when connected to a container database
Cause: The `CONTAINER` clause was specified when not connected to a container database (CDB).

Action: Reenter the statement without the `CONTAINER` clause.

ORA-65118: operation affecting a pluggable database cannot be performed from another pluggable database
Cause: An attempt was made to perform an operation affecting a pluggable database while connected to a different pluggable database.

Action: Connect to the desired pluggable database and perform the operation.

ORA-65119: incompatible national character set
Cause: The national character set of the container database was not the same as the national character set of the pluggable database being plugged in.

Action: Plug the pluggable database into a container database having the same national character set.

ORA-65120: illegal character set ID in XML metadata file
Cause: The XML metadata file for the pluggable database being plugged in contained a database or national character set ID that was not valid for the container database. This can happen if the XML file is manually modified, if it is corrupted on disk, or if the set of supported character sets differs between the container database and the source database of the pluggable database because a user-defined character set has not been installed in the Oracle Home directory of the container database.

Action: If you have manually modified the XML metadata file of the pluggable database, restore the original file or re-create the pluggable database from its source database. Manual modifications of the XML file are not supported. If the pluggable database uses a user-defined character set, make sure the character set is installed in the Oracle Home directory of the container database. Otherwise, contact Oracle Support Services.

ORA-65121: ALTER SESSION SET CONTAINER not allowed from this client
Cause: Clients earlier than Oracle Database 12c Release 1 do not support this feature.

Action: Upgrade the client to Oracle Database 12c Release 1 or later.

ORA-65122: Pluggable database GUID conflicts with the GUID of an existing container.
Cause: While creating a pluggable database, the GUID conflicted with the GUID of an existing container in the container database.

Action: Retry creating the pluggable database or consider using the AS CLONE clause for `CREATE PLUGGABLE DATABASE` if plugging in a copy of another pluggable database.

ORA-65123: cannot perform a SET CONTAINER operation in this context
Cause: An attempt was made to set the container inside a context where such an operation is prohibited.

Action: Do not set the container from inside a system trigger or a DML context.

ORA-65125: valid XML file name is required
Cause: An attempt was made to provide a file name that did not have an XML extension.

Action: Specify an XML file.

ORA-65126: pluggable database *string* cannot be unplugged.
Cause: The pluggable database was not closed cleanly and there are active transactions that need to be recovered.

Action: Open the pluggable database in read/write mode before unplugging again so that transaction recovery can be performed.

ORA-65127: PDB recover output file not specified
Cause: An attempt was made to recover a pluggable database (PDB) without specifying an output XML file location.

Action: Specify an output file location for recovery.

ORA-65128: PDB recover data file name not specified
Cause: An attempt was made to recover a pluggable database (PDB) without specifying a data file location.

Action: Specify a data file location for recovery.

ORA-65129: Pluggable database *string* cannot be relocated.
Cause: The pluggable database was already opened on all instances.

Action: Close the pluggable database without `RELOCATE` clause or close the pluggable database on another instance.

ORA-65130: cannot relocate more than one pluggable database.
Cause: An attempt was made to relocate more than one pluggable database.

Action: `RELOCATE` pluggable database one at a time.

ORA-65131: The feature *string* is not supported in a pluggable database.
Cause: An attempt was made to use a feature that is not supported in a pluggable database.

Action: Do not use this feature in a pluggable database.

ORA-65134: endian mismatch
Cause: The endian of the container database was not the same as the endian of the pluggable database being plugged in.

Action: Plug the pluggable database into a container database having the same endian.

ORA-65135: cannot perform ALTER SESSION SET CONTAINER operation in this context
Cause: An `ALTER SESSION SET CONTAINER` operation was attempted in a context where such an operation is prohibited.

Action: Do not perform the `ALTER SESSION SET CONTAINER` operation from a session that can be migrated or a session from an OCI connection pool.

ORA-65136: SPFILE name cannot be specified for a pluggable database
Cause: An attempt was made to specify an SPFILE name for a pluggable database.

Action: Remove the SPFILE name and retry the operation in the pluggable database.

ORA-65137: Pluggable database *string* is in the middle of Pluggable Database RESETLOGS operation.
Cause: The pluggable database is in the middle of a Pluggable Database (PDB) `RESETLOGS` operation.

Action: If possible, recover the Container Database further so that this Pluggable database is not in the middle of a PDB `RESETLOGS` operation. Another alternative is to perform PDB point-in-time recovery. A message is logged in the alert log indicating SCN and time until which the pluggable database can be point-in-time recovered.

ORA-65138: Data file *string* of pluggable database *string* belongs to an orphan PDB incarnation.
Cause: Either the specified data file was restored from a backup that was taken during a period of time that was discarded by a `RESETLOGS` operation, or Oracle could not identify which pluggable database incarnation the file belongs to. The alert log contains more information.

Action: Restore a backup of this file that belonged to either the current or a prior incarnation of the pluggable database. If you are using RMAN to restore, RMAN will automatically select a correct backup.

ORA-65139: Mismatch between XML metadata file and data file *string* for value of *string* (*string*)
Cause: Either the XML metadata file or the data file was corrupt.

Action: Verify that the XML metadata file and the data file are consistent as of the point when the unplug was done and retry the operation.

ORA-65140: invalid common profile name
Cause: An attempt was made to create a common profile with a name that is not valid for common profiles. Common profile names must start with C## or c## and consist only of ASCII characters.

Action: Specify a valid common profile name.

ORA-65141: invalid local profile name
Cause: An attempt was made to create a local profile with a name that is not valid for local profiles. Local profile names cannot start with C## or c##.

Action: Specify a valid local profile name.

ORA-65142: A local profile can be altered only within the current container
Cause: An attempt was made to alter a local profile using an `ALTER PROFILE` statement with the `CONTAINER=ALL` clause.

Action: If attempting to alter a local profile, ensure that you are connected to the correct container and optionally specify `CONTAINER=CURRENT`.

ORA-65143: creation of local profiles is not allowed in CDB$ROOT
Cause: An attempt was made to create a local profile in `CDB$ROOT`.

Action: If attempting to create a common profile, do not specify `CONTAINER=CURRENT`.

ORA-65144: ALTER SYSTEM DISABLE RESTRICTED SESSION is not permitted
Cause: An attempt was made to disable a restricted session while an unresolved error existed in `PDB_PLUG_IN_VIOLATIONS`.

Action: Resolve all of the errors before trying to disable a restricted session.

ORA-65145: FORCE open of a pluggable database on more than one instance is not supported
Cause: An attempt was made to `FORCE` open a pluggable database on more than one instance.

Action: Retry the `FORCE` open of the pluggable database one instance at a time.

ORA-65146: account cannot be unlocked in a PDB while it is locked in the root
Cause: An attempt was made to unlock a common user account in a pluggable database (PDB) which was locked in the root of the container database.

Action: Ensure that a common user account is not locked in the root before attempting to unlock it in a PDB.

ORA-65147: DB_UNIQUE_NAME specified without SPFILE scope
Cause: An attempt was made to specify a `DB_UNIQUE_NAME` in an `ALTER SYSTEM` statement without `SCOPE=SPFILE`.

Action: Use `DB_UNIQUE_NAME` only with `SCOPE=SPFILE`.

ORA-65148: cannot FORCE open a pluggable database to or from upgrade mode
Cause: An attempt was made to `FORCE` open a pluggable database to or from upgrade mode.

Action: Close the pluggable database first and retry the operation.

ORA-65149: PDB name conflicts with existing service name in the CDB or the PDB
Cause: An attempt was made to create a pluggable database (PDB) whose name conflicts with the existing service name in the container database (CDB) or the PDB.

Action: Choose a different name for the PDB.

ORA-65150: unable to start the instance
Cause: An attempt was made to start an instance as a container or a non-container database when at least one of the other instances was started in a different mode.

Action: Start the instance in the same mode as all the other instances.

ORA-65151: invalid tablespace name specified
Cause: An attempt was made to use an invalid tablespace name or a mandatory tablespace name (`SYSTEM` or `SYSAUX`) in the `USER_TABLESPACES` clause when creating a pluggable database.

Action: Specify a valid tablespace name.

ORA-65152: cannot bring datafile online
Cause: An attempt was made to bring online one or more data files belonging to a pluggable database that is disabled for recovery.

Action: Issue the `ALTER PLUGGABLE DATABASE ENABLE RECOVERY` statement first and retry the operation.

ORA-65153: cannot bring tablespace online
Cause: An attempt was made to bring online a tablespace that was either missing in the plug XML file or excluded when the pluggable database was created.

Action: Drop and recreate the tablespace to bring it online.

ORA-65154: specified logging attribute for the pluggable database is same as the current attribute value
Cause: An attempt was made to change the default logging attribute to be the same as the current attribute value for the pluggable database.

Action: Change the logging attribute to a different value.

ORA-65155: missing or invalid source file directory *string*
Cause: Source file directory was missing or invalid.

Action: Correct the `SOURCE_FILE_DIRECTORY` clause and reissue the statement.

ORA-65156: pluggable database version *string* not allowed
Cause: An attempt was made to plug in a pluggable database with an incompatible version of the multitenant container database.

Action: Plug in a pluggable database with a compatible version of the multitenant container database.

ORA-65157: SOURCE_FILE_NAME_CONVERT and SOURCE_FILE_DIRECTORY cannot be specified together
Cause: `SOURCE_FILE_NAME_CONVERT` and `SOURCE_FILE_DIRECTORY` clauses were both specified.

Action: Use either the `SOURCE_FILE_NAME_CONVERT` or the `SOURCE_FILE_DIRECTORY` clause and reissue the statement.

ORA-65158: could not find a matching file for - *string*
Cause: A matching file was not found in the directory specified with the `SOURCE_FILE_DIRECTORY` clause.

Action: Correct the `SOURCE_FILE_DIRECTORY` clause or check that the file exists in the directory and reissue the statement.

ORA-65159: invalid service name specified
Cause: An attempt was made to use an invalid service name in the services clause.

Action: Specify a valid service name.

ORA-65160: invalid cleanup task ID
Cause: An attempt was made to pass an invalid cleanup task ID to the `DBMS_PDB.CLEANUP_TASK` function.

Action: Specify a valid cleanup task ID.

ORA-65161: Unable to create pluggable database with no data
Cause: An attempt was made to clone a pluggable database which contains some clustered tables or index organized tables or advanced queue tables.

Action: Retry the clone operation after dropping all such objects from the source pluggable database.

ORA-65162: The password has expired.
Cause: The user's account expired and the password needs to be changed by connecting to the root of the multitenant container database

Action: Change the password or contact the DBA.

ORA-65164: database is in NOARCHIVELOG mode
Cause: The database was in `NOARCHIVELOG` mode when pluggable database (PDB) shutdown abort was attempted.

Action: Set the database to `ARCHIVELOG` mode by issuing the `ALTER DATABASE ARCHIVELOG` command.

ORA-65165: missing or invalid path for file creation *string*
Cause: The path specified in `CREATE_FILE_DEST` clause was missing or invalid.

Action: Correct the `CREATE_FILE_DEST` clause and reissue the statement.

ORA-65166: cannot run noncdb_to_pdb.sql if PDB's version differs from CDB's
Cause: An attempt was made to run noncdb_to_pdb.sql when the version of the pluggable database (PDB) was different from the version of the multitenant container database (CDB).

Action: Upgrade the PDB before running noncdb_to_pdb.sql

ORA-65167: cannot run noncdb_to_pdb.sql if container database is in upgrade mode
Cause: An attempt was made to run noncdb_to_pdb.sql when the multitenant container database (CDB) was in upgrade mode.

Action: Open the CDB in non-upgrade mode.

ORA-65168: missing PFILE name
Cause: An operation involving PFILE was issued in a pluggable database without specifying the PFILE name

Action: Specify a valid PFILE name and retry the operation in the pluggable database.

ORA-65169: error encountered while attempting to copy file *string*
Cause: An error was encountered while attempting to copy the file while creating a pluggable database.

Action: Check additional error messages for the cause of the failure to copy the file, and resolve the issue accordingly.

ORA-65170: XML file *string* already exists
Cause: An attempt to create an XML file failed because a file with that name already exists.

Action: Use a different XML file name.

ORA-65171: invalid value for DB_BLOCK_CHECKING parameter
Cause: An attempt was made to specify a value of `FALSE` or `OFF` for the `DB_BLOCK_CHECKING` parameter in a pluggable database (PDB), when the value of `DB_BLOCK_CHECKING` parameter in the multitenant container database (CDB) is neither `FALSE` nor `OFF`. If the CDB has enabled the `DB_BLOCK_CHECKING` parameter, then the PDB cannot disable the `DB_BLOCK_CHECKING` parameter.

Action: Specify a value other than `FALSE` or `OFF` for the `DB_BLOCK_CHECKING` parameter in the PDB.

ORA-65172: cannot run noncdb_to_pdb.sql unless pluggable database is an unconverted non-container database
Cause: An attempt was made to run 'noncdb_to_pdb.sql' on a pluggable database (PDB) that was not an unconverted non-container database.

Action: 'noncdb_to_pdb.sql' is not necessary for this PDB.

ORA-65173: privilege granted with a different scope to '*string*'
Cause: An attempt to revoke a privilege that was granted with a different scope has failed. The user tried to either revoke a common privilege in the current container or to revoke a local privilege in a multitenant container database (CDB).

Action: Specify the correct value for the CONTAINER clause.

ORA-65174: invalid or conflicting name in service *string* found in the pluggable database
Cause: The service name or network name in the specified service is invalid or it conflicts with an existing service name or network name in the container database.

Action: Use an appropriate SERVICE_NAME_CONVERT clause and reissue the statement.

ORA-65175: cannot grant SYSDBA privilege locally in the root
Cause: An attempt was made to grant SYSDBA privilege locally in the root of a multitenant container database (CDB).

Action: While connected to the root, SYSDBA privilege can only be granted commonly.

ORA-65176: system tablespace block size (*string*) does not match configured block sizes
Cause: The block size of the system tablespace of the pluggable database to be plugged in did not match the block sizes configured in the container database.

Action: Configure the appropriate cache for the block size of the system tablespace using the DB_<n>K_CACHE_SIZE parameter (where <n> is 2, 4, 8, 16, or 32).

ORA-65177: cannot create common user with the same name as local user
Cause: An attempt was made to create a common user with the same name as an existing local user.

Action: Specify a different user name.

ORA-65178: invalid logging mode specified for the pluggable database
Cause: An attempt was made to either enable a conflicting logging mode or to disable a logging mode when it was not enabled.

Action: Check the current logging modes in DBA_PDBS for the pluggable database and specify a valid mode.

ORA-65179: cannot keep datafiles for a pluggable database that is not unplugged
Cause: An attempt was made to drop a pluggable database without specifying the INCLUDING DATAFILES clause, and the pluggable database has not been unplugged.

Action: Unplug the pluggable database before dropping the pluggable database or use the INCLUDING DATAFILES clause in the DROP PLUGGABLE DATABASE statement.

ORA-65180: duplicate file name encountered - *string*
Cause: An attempt was made to issue a `CREATE PLUGGABLE DATABASE` statement with a duplicate source or target file name for the source file name mentioned in the message. This could be caused by incorrect `SOURCE_FILE_NAME_CONVERT` or `FILE_NAME_CONVERT` clause.

Action: Use an appropriate `SOURCE_FILE_NAME_CONVERT` or `FILE_NAME_CONVERT` clause to ensure that there are no duplicate source or target file names and then reissue the statement.

ORA-65181: invalid argument supplied to CONTAINERS function
Cause: An attempt was made to create a multitenant container database (CDB) view with an invalid table or view name.

Action: Specify a valid table or view name of the form <owner>.<table_name | view_name>.

ORA-65182: unable to modify the state of pluggable database *string*
Cause: An attempt was made to modify the state of a pluggable database which is currently in the middle of a state transition.

Action: Retry the operation later.

ORA-65183: FORCE open of a PDB on only a few instances where its open on is not allowed
Cause: An attempt was made to `FORCE` open a pluggable database on only a few Oracle RAC instances where it is currently open.

Action: Retry the `FORCE` open and include all Oracle instances where the PDB is open

ORA-65184: encountered data files belonging to different PDBs
Cause: An attempt was made to invoke `DBMS_PDB.RECOVER()` by specifying data files or a directory name containing data files belonging to different PDBs.

Action: Check the paths provided to `DBMS_PDB.RECOVER()` and retry the operation.

ORA-65185: could not find data file belonging to SYSTEM tablespace
Cause: An attempt was made to invoke `DBMS_PDB.RECOVER()` by specifying data files or a directory name containing data files where no data file belonging to `SYSTEM` tablespace was found. `SYSTEM` tablespace is critical to recover a PDB.

Action: Check the paths specified to `dbms_pdb.recover()` and ensure that it at least contains data files belonging to `SYSTEM` tablespace.

ORA-65186: The user, role or profile *string* has sync errors.
Cause: There were pending SQL statements involving this user, role or profile which need to be resolved in order for the pluggable database (PDB) to be synced with ROOT.

Action: Resolve the pending SQL statement in the PDB and retry the operation.

ORA-65400: Table does not have CLUSTERING clause associated with it
Cause: User attempts to modify clustering clause on a table that does not have one.

Action: Create `CLUSTERING` clause on the table first.

ORA-65402: invalid option on CLUSTERING clause
Cause: Invalid option on `CLUSTERING` clause. Only `[YES | NO ] ON LOAD` or `[YES | NO ] ON DATA MOVEMENT` are valid.

Action: Correct the option on the `CLUSTERING` clause.

ORA-65403: invalid usage of CLUSTERING clause
Cause: Invalid usage of `CLUSTERING` clause. The clause can be specified only for `CREATE/ATLER TABLE/SNAPSHOT/MATERIALIZED VIEW`

Action: Don't use `CLUSTERING` clause in this statement

ORA-65404: CLUSTERING clause specified more than once on a table
Cause: A table can have only one clustering clause. The clause was specified multiple times.

Action: Correct the statement to use only one `CLUSTERING` clause.

ORA-65405: CLUSTERING clause cannot be used with existing table options
Cause: `CLUSTERING` clause conflicts with other clauses on the table. Clustering cannot be used on `IOT`, `CLUSTERED`, `GLOBAL TEMPORARY TABLE`, `EXTERNAL TABLE`.

Action: Correct the statement so clustering clause does not conflict with other clauses.

ORA-65406: invalid option on CLUSTERING clause
Cause: `CLUSTERING` was not followed by `BY [LINEAR | MULTIDIMENSIONAL] ORDER`.

Action: Correct the option on the `CLUSTERING` clause.

ORA-65407: CLUSTERING clause already exists on the table
Cause: Add `CLUSTERING` clause was specified on a table with an existing clustering clause.

Action: If table has clustering, you can only change `ON LOAD` or `ON DATA MOVEMENT` options. To change other options, you must first drop the clause.

ORA-65408: CLUSTERING clause has too many columns in BY ORDER
Cause: More than 10 columns were specified in the `BY ORDER` subclause. You can have at most 10 columns in the `BY ORDER` subclause.

Action: Lower the number of columns in the `BY ORDER` subclause to a value of `10` or less.

ORA-65409: CLUSTERING clause has too many MULTIDIMENSIONAL columns or groups
Cause: More than 40 columns were specified in the `MULTIDIMENSIONAL ORDER` subclause.You can have at most 40 columns in the `MULTIDIMENSIONAL ORDER` subclause.

Action: Lower the number of columns in the `BY MULTIDIMENSIONAL ORDER` subclause to a value of 40 columns or less or 4 groups or less.

ORA-65410: CLUSTERING clause can have only scalar columns
Cause: A non-scalar column was specified in the `BY LINEAR` or `MULTIDIMENSIONAL ORDER` subclause. Also, virtual or hidden columns are not allowed.

Action: Specify only scalar columns in the `BY [LINEAR | MULTIDIMENSIONAL] ORDER` subclause.

ORA-65411: CLUSTERING clause does not exist
Cause: An attempt was made to modify or drop a `CLUSTERING` clause in a table that had no `CLUSTERING` clause associated with it.

Action: Do not attempt to modify a nonexistent `CLUSTERING` clause.

ORA-65412: invalid option on ALTER TABLE .. CLUSTERING ..
Cause: You have specified an invalid option on `ALTER TABLE...CLUSTERING`. For example, you specified `DROP` and `ADD CLUSTERING` in one statement.

Action: Correct the statement.

ORA-65413: cannot resolve referenced object in the CLUSTERING clause
Cause: An object was referenced in the `CLUSTERING` clause that could not be resolved to a base table reference.

Action: The `CLUSTERING` clause can refer to base tables only.

ORA-65414: column resolves to multipe tables in the CLUSTERING clause
Cause: A column resolved to multiple tables in the `CLUSTERING` clause.

Action: Correct the statement. The column should resolve to only one table. Need to qualify by table name.

ORA-65415: wrong join condition in the CLUSTERING clause
Cause: The `CLUSTERING` clause had an `OR` condition and the individual join condition on the base table columns was not an equijoin.

Action: Correct the statement. Make sure that join conditions are equijoins and there are no OR conditions, or no expressions in the join.

ORA-65416: multiple mimension tables in the CLUSTERING clause
Cause: Joins in the `CLUSTERING` referenced a dimension table more than once.

Action: Correct the statement. Remove one of the references.

ORA-65417: illegal dimenstion table in the CLUSTERING clause
Cause: The dimension table in the `CLUSTERING` clause was illegal. Dimension table cannot be `GLOBAL TEMPORARY TABLE` or `EXTERNAL TABLE`.

Action: Rewrite the statement so it does not use an illegal option on the dimension table.

ORA-65418: primary or unique Key constraint missing in CLUSTERING join
Cause: The join between the fact and dimension tables in the `CLUSTERING` clause did not have a primary or unique key constraint on the dimension table.

Action: Add a primary or unique key constraint to the join columns on the dimension table.

ORA-65419: columns in a clustering group come from different tables
Cause: Columns in the clustering group did not come from the same table. For example: `... CLUSTERING BY ... ( (t1.c1, t2.c2), (t2.c1, t2.c2) )`. Clustering group `(t1.c1, t2.c2)` must contain references to only one table.

Action: Restructure the clustering group so it contains only references to a single table.

ORA-65420: columns in a join condition in the CLUSTERING clause are not compatible
Cause: The data type of columns in the join condition of `CLUSTERING` clause were not compatible. They must be of the same type.

Action: Select a compatible data type for the join conditions in the `CLUSTERING` clause.

ORA-65421: The CLUSTERING clause is defined on columns that are to be modified
Cause: An `ALTER TABLE MODIFY COLUMN` command was issued on a column on which a `CLUSTERING` clause exists.

Action: Drop the CLUSTERING clause before attempting to modify the column.

ORA-65422: The CLUSTERING clause already has a zonemap defined on it
Cause: An `ALTER TABLE MODIFY CLUSTERING` command was issued with the `WITH MATERIALIZED ZONEMAP` option but the table already had zonemap define on it.

Action: Drop the zonemap associated with the base table before attempting to add zonemap using the `CLUSTERING` clause.

ORA-65423: The CLUSTERING clause does not have a zonemap associated with it
Cause: An `ALTER TABLE MODIFY CLUSTERING` command was issued with the `WITHOUT MATERIALIZED ZONEMAP` option but the table did not have zonemap associated with the `CLUSTERING` clause.

Action: Do not use the `ALTER TABLE MODIFY CLUSTERING WITHOUT ZONEMAP`.

ORA-65424: CLUSTERING clause has too many joins
Cause: More than 4 joins were specified in the `CLUSTERING` clause. You can have at most 4 joins to the dimension tables.

Action: Reduce the number of joins to 4 or less.

ORA-65425: CLUSTERING clause not supported for table stored in tablespace of this storage type
Cause: An attempt was made to define clustering on a table stored in non-Oracle Exadata storage.

Action: Create this table in a tablespace residing on Oracle Exadata storage or remove the `CLUSTERING` clause.

ORA-65426: Non-scalar data type is used for a clustering column
Cause: Oracle cannot perform data clustering because an attempt was made to populate a clustering column with non-scalar data values, or a clustering column defaulted to non-scalar values was omitted.

Action: Use only scalar data type for the clustering columns, or disable data clustering with `NO_CLUSTERING` hint.

ORA-65451: Advanced index compression is not supported for tablespaces on this storage type.
Cause: An attempt was made to use advanced index compression on an unsupported storage type.

Action: Drop and re-create the index with a compression option that is supported on the current storage type.

ORA-65455: family *string* is invalid
Cause: An invalid family was used to specify a parameter value.

Action: Use a valid family.

ORA-65456: family *string* is not valid in *string* instance
Cause: The specified family could not be used in the current instance type.

Action: Remove the family from the environment variable `ORACLE_FAMILY`.

ORA-65457: family *string* is not valid in *string* instance
Cause: The specified family could not be used in the current instance type.

Action: Use a family that is valid for this instance to set the initialization parameter.

ORA-65458: maximum length of parameter qualifier exceeded
Cause: Too many characters were specified in the parameter qualifier in the initialization parameter file.

Action: Change the parameter qualifier to a valid SID or family.

ORA-65459: family used with a specific system identifier (SID)
Cause: A SID was used with `FAMILY`.

Action: Use `FAMILY` without a SID.

ORA-65460: family *string* is invalid
Cause: An invalid family was specified for this instance in `ORACLE_FAMILY`.

Action: Use a valid family in `ORACLE_FAMILY`.

ORA-65461: invalid parameter name specified with CONTAINER clause set to ALL
Cause: The `CONTAINER` clause was set to `ALL` in the `ALTER SYSTEM SET` statement for a non-PDB modifiable parameter.

Action: Retry the `ALTER SYSTEM SET` statement without the `CONTAINER` clause.

ORA-65466: family '*string*' contains an illegal character or is too long
Cause: The specified family contained an illegal character or the family specified was too long, which cannot occur in an SPFILE setting. Illegal characters include `*,#"'=()` and whitespace.

Action: Use a family that does not contain a special character or whitespace. Check platform-specific documentation for the maximum length of family.

ORA-65500: could not modify DB_UNIQUE_NAME, resource exists
Cause: `DB_UNIQUE_NAME` could not be modified, because a database resource identified by `DB_UNIQUE_NAME` existed in the cluster. To modify `DB_UNIQUE_NAME`, any resource it identifies must first be removed.

Action: Remove the database resource identified by `DB_UNIQUE_NAME` before modifying `DB_UNIQUE_NAME`.

ORA-65501: locator from *string* container cannot be used in container *string* for this operation
Cause: In a pluggable database environment, an attempt was made to modify a LOB using a locator which belongs to a different container.

Action: Switch back to the original container.

ORA-65502: cannot access temporary LOB data
Cause: The database did not open.

Action: Open the database before accessing LOB data.

ORA-65535: Oracle client cannot handle error code exceeding 65535. Actual error code and message follow:
Cause: Oracle server encountered an error whose error code exceeded 65535. Older Oracle clients (prior to version 12) cannot handle error codes larger than 65535.

Action: Actual Oracle error code and error message are a part of the error message for ORA-65535. Upgrade Oracle client to version 12 client libraries or higher.

Glossary

alert

In Cloud Control, an indicator that a particular metric condition has been encountered. For example, an alert might indicate that a metric threshold has been reached.

attribute set

A set of parameters set at the job level while copying Recovery Appliance backups to tape. Attribute sets are created as part of a media manager library for each drive associated with this library.

Auto Service Request (ASR)

A product feature that automatically opens service requests when specific Recovery Appliance hardware faults occur. ASR detects faults in the most common server components, such as disks, fans, and power supplies.

automated delta pool space management

The set of operations in which a Recovery Appliance determines which blocks are no longer needed, and then deletes them. Specifically, space management includes:

- Determining which backups (both in a Recovery Appliance storage location and on tape) are obsolete or expired based on the disk recovery window goal and SBT retention policy
- Deleting unneeded blocks from the Recovery Appliance storage to meet the disk recovery window goal and reserved space parameters configured for each protected database
- Optimizing the delta pools to improve performance of restore operations

backup copy policy

An attribute of a protection policy that determines whether the Recovery Appliance must ensure that new backups are replicated or copied to tape before deletion.

backup ingest

The automated stage in which a Recovery Appliance scans a backup that was sent by a protected database. The Recovery Appliance decomposes the backup into smaller sets of blocks, writes the blocks into the appropriate storage location, and indexes the backups. Indexing includes inserting rows into the Recovery Appliance metadata database to describe the physical location of every block.

backup mode

The database mode (also called *hot backup mode*) initiated when you issue the `ALTER TABLESPACE ... BEGIN BACKUP` or `ALTER DATABASE BEGIN BACKUP` statement before taking an online backup. You take a tablespace out of backup mode when you issue the `ALTER TABLESPACE ... END BACKUP` or `ALTER DATABASE END BACKUP` statement.

backup polling directory

A file system directory on shared storage, located outside the Recovery Appliance, that is a destination for backup pieces and archived redo log files from a protected database. The Recovery Appliance polls the directory at specified intervals, retrieves any found backup data, and then processes and stores the data.

backup polling policy

An optional Recovery Appliance object that defines a storage area where a client database will place backups without interacting directly with the Recovery Appliance. The polling policy defines the file system path to the storage and how often it is searched for new backups.

backup reception

The stage in which a protected database sends a backup over the network to a Recovery Appliance, but before the Recovery Appliance has indexed the backup.

backup window

The amount of time that it takes for a backup to complete.

block change tracking

A database option that causes Oracle Database to track data file blocks affected by each database update. The tracking information is stored in a block change tracking file. When block change tracking is enabled, RMAN uses the record of changed blocks from the change tracking file to improve incremental backup performance by only reading blocks known to have changed, instead of reading whole data files.

cascaded replication

A configuration in which a downstream Recovery Appliance also serves as an upstream Recovery Appliance for a Recovery Appliance further downstream.

Cloud Control

Oracle Enterprise Manager Cloud Control is Oracle's enterprise cloud management solution. It enables you to monitor and manage the complete Oracle IT infrastructure from a single console. The core components of the architecture include the Oracle Management Agent, Oracle Management Service, Oracle Management Repository, Enterprise Manager for Zero Data Loss Recovery Appliance plug-in, and Enterprise Manager Cloud Control Console.

copy-on-write snapshot

After a third-party storage snapshot is taken, and when the first change occurs on a storage block, the array copies the before-image block to a new location on disk. The

snapshot maintains the before-image block for the snapshot and the new block for the active version of the database.

delta pool

A set of data file blocks from which a virtual full backup is constructed. Each separate data file backed up to a Recovery Appliance has its own separate delta pool. The delta pools reside in the delta store.

delta pool optimization

The automatic tracking and reorganizing of the delta pools. As old blocks are deleted and new incremental backups arrive for updated data files, the blocks in a backup can become less contiguous. This state can degrade the performance of restore operations. Recovery Appliance automatically reorganizes the blocks to maintain contiguity during ordinary maintenance and validation.

delta push

The transfer of backups and changes from protected databases to the Recovery Appliance. This solution consists of two operations that run on each protected database: real-time redo transport, and the incremental-forever backup strategy.

delta store

The total Recovery Appliance storage that is used to store client backup data. The delta store contains all data file and archived redo log backups.

disk recovery window goal

The interval in which a point-in-time recovery must be possible using only disk backups. For example, if the recovery window goal is 15 days, and if it is noon on April 25, then the goal is the ability to perform point-in-time recovery to any time on or after noon on April 10. The goal, which is specified for each protection policy, is not a hard limit.

downstream Recovery Appliance

In a Recovery Appliance replication topology, the downstream Recovery Appliance receives replicated data from an upstream Recovery Appliance.

enrolling a database

The process of enabling a specific Recovery Appliance to receive backups from a protected database. Enrolling involves adding the protected database (`DBMS_RA.ADD_DB`), granting access to this database to a Recovery Appliance user account (`DBMS_RA.GRANT_DB_ACCESS`), and registering this database in the virtual private catalog (RMAN `REGISTER DATABASE` command).

fast recovery area

An optional disk location that you can use to store recovery-related files such as control file and online redo log copies, archived redo log files, flashback logs, and RMAN backups.

guaranteed copy

An optional setting of a protection policy that indicates that every backup must be copied to tape or replicated. Recovery Appliance cannot purge backups from the storage location until the operation succeeds. If tape or replication does not keep up, then the Recovery Appliance may reject new backups.

incremental-forever backup strategy

The strategy in which an initial level 0 backup is taken to the Recovery Appliance, with all subsequent incremental backups occurring at level 1. The Recovery Appliance creates a virtual full backup by combining the initial level 0 with subsequent level 1 backups.

media manager library

This is the media management library that manages tape backup job s. This library consists of attribute sets for each of its contained drives and defines storage parameters that apply to tape backup jobs.

media management software

The media management software is the middleware between the Recovery Appliance and the tape. It controls and manages the copying of backups from the Recovery Appliance to tape.

Recovery Appliance uses Oracle Secure Backup as its media management software and comes preconfigured with it.

one-way Recovery Appliance replication

The simplest form of the Recovery Appliance replication, in which one upstream Recovery Appliance sends backups to one downstream Recovery Appliance.

Oracle Configuration Manager

A tool that collects and uploads configuration information from Oracle homes in your environment. If you log a service request, then the configuration data enables Oracle Support Services to provide better service.

Oracle Enterprise Manager Cloud Control

See Cloud Control.

protected database

A client database that backs up data to a Recovery Appliance.

protection policy

A group of attributes that control how a Recovery Appliance stores and maintains backup data. Each protected database is assigned to exactly one protection policy, which controls all aspects of backup processing for that client.

real-time redo transport

The continuous transfer of redo changes from the SGA of a protected database to a Recovery Appliance. Real-time redo transport enables RMAN to provide a recovery

point objective (RPO) near 0. Typically, RMAN can recover to within a second of the time when the failure occurred. Protected databases write redo entries directly from memory to the Recovery Appliance as they are generated.

reconciling

In Recovery Appliance replication, the process by which a Recovery Appliance receives metadata from the Recovery Appliances that are immediately downstream.

Recovery Appliance

Shortened name for Zero Data Loss Recovery Appliance. Recovery Appliance is an Oracle Engineered System specifically designed to protect Oracle databases. Integrated with RMAN, it enables a centralized, incremental-forever backup strategy for hundreds to thousands of databases across the enterprise, using cloud-scale, fully fault-tolerant hardware and storage.

Recovery Appliance user account

A user account that is authorized to connect to, and request services from, Recovery Appliance. Every Recovery Appliance user account is an Oracle Database user account on the Recovery Appliance metadata database, and the owner of a virtual private catalog. When RMAN backs up a protected database, it connects to the recovery catalog with the Recovery Appliance user account credentials.

Recovery Appliance administrator

The administrator who manages a Recovery Appliance. Typical duties include creating and adding databases to protection policies, managing storage space, managing user accounts, configuring tape backups and the Recovery Appliance replication, and monitoring the Recovery Appliance.

Recovery Appliance Backup Module

An Oracle-supplied SBT library that RMAN uses to send backups of protected databases over the network to the Recovery Appliance. The library must be installed in each Oracle home used by a protected database.

The module functions as an SBT media management library that RMAN references when allocating or configuring a channel for backup to the Recovery Appliance. RMAN performs all backups to the Recovery Appliance, and all restores of complete backup sets, using this module.

Recovery Appliance metadata database

The Oracle database that runs inside of the Recovery Appliance. This database stores configuration data such as user definitions, protection policy definitions, and client database definitions. The metadata database also stores backup metadata, including the contents of the delta store.

Recovery Appliance replication

A configuration in which one Recovery Appliance receives backups, and then forwards them to another Recovery Appliance. The forwarder is the upstream Recovery Appliance, and the receiver is the downstream Recovery Appliance.

Recovery Appliance schema

The schema on the Recovery Appliance metadata database owned by the `RASYS` user. The schema is the super-set of the recovery catalog schema, and contains additional metadata used internally by Recovery Appliance to manage backups.

Recovery Appliance storage location

A set of Oracle ASM disk groups within Recovery Appliance that stores backups. A storage location can be shared among multiple protected databases. Every Recovery Appliance contains the default Recovery Appliance storage location named `DELTA`.

recovery point objective (RPO)

The data-loss tolerance of a business process or an organization. The RPO is often measured in terms of time, for example, five hours or two days worth of data loss.

recovery window

A setting that defines how long the Recovery Appliance maintains tape backups in its catalog for recovery purposes.

recovery window goal

The time interval within which a protected database must be recoverable to satisfy business requirements. For each protected database in a protection policy, the Recovery Appliance attempts to ensure that the oldest backup on disk is able to support a point-in-time recovery to any time within the specified interval (for example, the past 7 days), counting backward from the current time.

redo staging area

For Recovery Appliance installations that enable real-time redo transport recovery, the Recovery Appliance storage destination for redo streams transmitted by protected databases. The Recovery Appliance converts the redo streams into archived redo log files, which it then converts to backup pieces and writes to a storage location.

replication user account

Oracle requires that you create a replication user account exclusively for use with Recovery Appliance replication, and that you create a unique replication user account for each upstream appliance within the organization.

Oracle recommends that the replication user account takes the form of `REPUSER_FROM_[ZDLRA_DB_NAME` or `ZDLRA_DB_LOCATION]`.

For example, if two Recovery Appliances have the `DB_UNIQUE_NAME` of ZDLRA1 and ZDLRA2, then the replication user accounts could be `REPUSER_FROM_ZDLRA1` and `REPUSER_FROM_ZDLRA2`. Or if those same Recovery Appliances were in Florence and

Vienna, then the replication user accounts could be `REPUSER_FROM_FLORENCE` and `REPUSER_FROM_VIENNA`

The replication user account **should not** be used as a regular VPC user employed by protected databases to connect and send backups to the Recovery Appliance.

A database user account on the downstream Recovery Appliance that upstream Recovery Appliances will use to authenticate with this downstream Recovery Appliance.

reserved space

The minimum amount of disk space in the Recovery Appliance that is reserved for use by one protected database to meet its disk recovery window goal. The reserved space cannot be consumed by any other protected database. In general, the Recovery Appliance ignores reserved space settings until it is under space pressure, when it uses these settings and recovery window goals to determine which backups to purge.

retention policy

The length of time, expressed as a window of time extending backward from the present, that backups are kept on a SBT device. Backups may be kept longer than the specified window because they are kept long enough to guarantee that point-in-time recovery is possible to any point within the retention policy window.

RMAN recovery catalog

A set of metadata views residing in the Recovery Appliance metadata database.

SBT

System Backup to Tape. This term specifies a backup device type, typically either a tape device or Recovery Appliance. RMAN supports channels of type disk and SBT.

tape backup job

The operation that copies Recovery Appliance backups to tape based on the defined properties such as the associated media manager library, attribute set, backup type, and run-time window. This repeatable job can be scheduled to run immediately after being created or at a later specified time.

third-party storage snapshot

A set of pointers, managed by a third-party storage device, to storage blocks (*not* Oracle blocks) that existed when the snapshot was created. The device maintains a snapshot on the same storage array as the original data. The device only creates new versions of storage blocks when the snapshot perceives that they have changed.

unprotected window threshold

The user-specified maximum amount of data loss for protected databases that are subject to a protection policy. For example, a specified threshold of 5 minutes for protection policy `SILVER` means that every database protected by `SILVER` can lose no more than 5 minutes of data.

upstream Recovery Appliance

In a Recovery Appliance replication topology, the Recovery Appliance that is replicating backups to another Recovery Appliance.

virtual full backup

A complete database image as of one distinct point in time, maintained efficiently through the indexing of incremental backups from a protected database. The virtual full backups contain individual blocks from multiple incremental backups. For example, if you take a level 0 backup on Monday with SCN 10000, and if you take an incremental level 1 backup on Tuesday with SCN 11000, then the Recovery Appliance metadata database shows a virtual level 0 backup current to SCN 11000.

Essentially, virtual full backups are space-efficient, pointer-based representations of physical full backups as of the point-in-time of an incremental backup. When a restore operation is required, the delta store re-creates a physical full backup from the appropriate incremental backup SCN.

virtual private catalog

A subset of the metadata in a base RMAN recovery catalog to which a database user account is granted access. Each restricted user account has full read/write access to its own virtual private catalog.

Zero Data Loss Recovery Appliance

See Recovery Appliance.

Zero Data Loss Recovery Appliance Backup Module

See Recovery Appliance Backup Module.

Index

D

E

F

G

I

L

M

O

P

R

S

T

U

V

Z