Oracle® Database

High Availability Best Practices

12*c* Release 1 (12.1)

E40019-02

July 2015

ORACLE®

Oracle Database High Availability Best Practices 12*c* Release 1 (12.1)

E40019-02

Primary Authors: Lawrence To, Viv Schupmann, Thomas Van Raalte, Virginia Beecher

Contributing Author: Janet Stern

Contributors: Andrew Babb, Janet Blowney, Larry Carpenter, Timothy Chien, Jay Davison, Senad Dizdar, Ray Dutcher, Mahesh Girkar, Stephan Haisley, Wei Ming Hu, Holger Kalinowski, Nitin Karkhanis, Frank Kobylanski, Rene Kundersma, Joydip Kundu, Barb Lundhild, Roderick Manalac, Pat McElroy, Robert McGuirk, Joe Meeks, Markus Michalewicz, Valarie Moore, Michael Nowak, David Parker, Darryl Presley, Hector Pujol, Michael T. Smith, Vinay Srihari, Douglas Utzig, Thomas Van Raalte, James Viscusi, Vern Wagman, Steve Wertheimer, Shari Yamaguchi

Contents

4 Configuring Oracle Database

5 Configuring Oracle Database with Oracle Clusterware

8 Configuring Oracle Data Guard

9 Configuring Oracle GoldenGate

10 Client Failover Best Practices for Highly Available Oracle Databases

11 Monitoring for High Availability

12 Recovering from Unscheduled Outages

List of Figures

List of Tables

Preface

This book provides high availability best practices for configuring and maintaining your Oracle Database system and network components.

Audience

This book is intended for chief information technology officers and architects, as well as administrators who perform the following database, system, network, and application tasks:

- Plan data centers
- Implement data center policies
- Maintain high availability systems
- Plan and build high availability solutions

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 Oracle database documentation set. These books may be of particular interest:

- *Oracle Database High Availability Overview*
- *Oracle Data Guard Concepts and Administration* and *Oracle Data Guard Broker*
- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database Backup and Recovery User's Guide*

- *Oracle Database Administrator's Guide*
- The Oracle High Availability Best Practice white papers that can be downloaded from the Oracle Technology Network (OTN) at

  `http://www.oracle.com/goto/maa`
- The Oracle Enterprise Manager documentation library at

  `http://www.oracle.com/technetwork/oem/grid-control/overview/index.html`

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 High Availability Best Practices

By implementing and using Oracle Maximum Availability Architecture (MAA) best practices, you can provide high availability for the Oracle database and related technology.

This chapter contains the following topics:

- Oracle Database High Availability Architectures
- Oracle Database High Availability Best Practices
- Oracle Maximum Availability Architecture

Oracle Database High Availability Architectures

Designing and implementing a high availability architecture can be a daunting task given the broad range of Oracle technologies and hardware, software, and deployment options. A successful effort begins with clearly defined and thoroughly understood business requirements. *Oracle Database High Availability Overview* Chapter 2, "High Availability and Data Protection – Getting From Requirements to Architecture," helps you map your specific business requirements to one of four MAA reference architectures. Later chapters in Oracle Database High Availability Overview describe how each MAA reference architecture provides various benefits and tradeoffs in addressing unplanned outages and planned maintenance activities. Finally, Chapter 7, "High Availability Architectures," compares and contrasts the different architectures. We highly recommend reviewing *Oracle Database High Availability Overview* to choose the most appropriate MAA reference architecture and to learn how Oracle Engineered Systems and other architectural decisions can optimize your return on investment.

Oracle Database High Availability Best Practices

Oracle High Availability (HA) best practices help you deploy a highly available architecture throughout your enterprise. Having a set of configuration and operational best practices helps you achieve high availability and reduces the cost associated with the implementation and ongoing maintenance of your enterprise. Also, employing best practices can optimize usage of system resources.

By implementing the HA best practices you can:

- Reduce the cost of creating an Oracle Database high availability system by following detailed guidelines on configuring your database, storage, application failover, backup and recovery. See Chapter 2, "Overview of Configuration Best Practices" for more information.

- Use operational best practices to maintain your system. See Chapter 6, "Operational Prerequisites to Maximizing Availability" in *Oracle Database High Availability Overview* for more information.
- Detect and quickly recover from unscheduled outages caused by computer failure, storage failure, human error, or data corruption. For more information, see Section , "Protect Against Data Corruption" and Chapter 12, "Recovering from Unscheduled Outages".
- Eliminate or reduce downtime due to scheduled maintenance such as database patches or application upgrades as described in Chapter 13, "Reducing Downtime for Planned Maintenance".

Oracle Maximum Availability Architecture

Oracle Maximum Availability Architecture (MAA) is Oracle's best practices blueprint based on Oracle High Availability (HA) technologies, extensive validation performed by the Oracle MAA development team, and the accumulated production experience of customers who have successfully deployed business critical applications on Oracle.

MAA covers Oracle products within the following technologies:

- Oracle Database as described in this book
- Oracle Exadata Database Machine and Oracle Exalogic Elastic Cloud
- Oracle Fusion Middleware and Oracle WebLogic Server
- Oracle Applications (Siebel, Peoplesoft, E-Business Suite)
- Oracle Collaboration Suite
- Oracle Enterprise Manager
- MAA reference architectures: Bronze, Silver, Gold, and Platinum as described in *Oracle Database High Availability Overview*.

This book, *Oracle Database High Availability Best Practices* primarily focuses on high availability best practices for the Oracle Database. There are also other components for which you might want to consider Oracle Maximum Availability Architecture (MAA) best practices. For more information go to:

`http://www.oracle.com/goto/MAA`

See:

- *Oracle Fusion Middleware Disaster Recovery Guide* for information about Oracle Fusion Middleware high availability
- *Oracle Fusion Middleware Administrator's Guide* for information about backup and recovery for Oracle Fusion Middleware

The goal of MAA is to achieve the optimal HA architecture at the lowest cost and complexity. MAA provides:

- Best practices that span the Exadata Database Machine, Oracle Database, Oracle Fusion Middleware, Oracle Applications, Oracle Enterprise Manager, and solutions provided by Oracle Partners.
- Accommodates a range of business requirements with MAA reference architectures to support all Oracle databases.
- Leverages lower-cost servers and storage.

- Uses hardware and operating system independent features and evolves with new Oracle versions and features. The only exceptions are Oracle Engineered Systems such as Oracle Exadata Database Machine which has specific and customized configuration and operating practices optimized for Oracle Database.
- Makes high availability best practices as widely applicable as possible considering the various business service level agreements (SLA).
- Uses the Oracle Grid Infrastructure with Database Server Grid and Database Storage Grid to provide highly resilient, scalable, and lower cost infrastructures.
- Provides the ability to control the length of time to recover from an outage and the amount of acceptable data loss from any outage.

For more information about MAA and documentation about best practices for all MAA components, visit the MAA website at

http://www.oracle.com/goto/maa

2

Overview of Configuration Best Practices

Using Oracle Enterprise Manager 11*g* is the MAA best practice recommendation for configuring your entire high availability environment. Oracle Enterprise Manager is Oracle's single, integrated solution for managing all aspects of the Oracle Grid and the applications running on it. Oracle Enterprise Manager Grid Control couples top-down monitoring for applications with automated configuration management, provisioning, and administration. This powerful combination provides unequaled management for any size Oracle data center.

Using Oracle Enterprise Manager you can perform most configuration tasks. For example, you can:

- Migrate to Oracle Automatic Storage Management (Oracle ASM)
- Migrate a single-instance Oracle Database to Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC)
- Create Oracle Data Guard standby databases
- Configure backup and recovery
- Implement Oracle Active Data Guard
- Use the MAA Advisor to implement Oracle's best practices and achieve a high availability architecture

For information about the configuration Best Practices for Oracle Database, see the following chapters:

- Chapter 3, "Configuring Storage"
- Chapter 4, "Configuring Oracle Database"
- Chapter 5, "Configuring Oracle Database with Oracle Clusterware"
- Chapter 6, "Configuring Oracle Database with Oracle RAC"
- Chapter 8, " Configuring Oracle Data Guard"
- Chapter 7, "Configuring Backup and Recovery"
- Chapter 9, "Configuring Oracle GoldenGate"
- Chapter 10, "Client Failover Best Practices for Highly Available Oracle Databases"

> **See Also:** Oracle Enterprise Manager online help system, and the documentation set available at
>
> `http://www.oracle.com/technetwork/oem/grid-control/index.html`

3

Configuring Storage

This chapter describes best practices for configuring a fault-tolerant storage subsystem that protects data while providing manageability and performance. These practices apply to all Oracle Database high availability architectures described in *Oracle Database High Availability Overview*.

This chapter contains the following topics:

Evaluate Database Performance and Storage Capacity Requirements

Characterize your database performance requirements using different application workloads. Extract statistics during your target workloads by gathering the beginning and ending statistical snapshots. Some examples of target workloads include:

- Average load
- Peak load
- Application workloads such as batch processing, Online Transaction Processing (OLTP), decision support systems (DSS) and reporting, Extraction, Transformation, and Loading (ETL)

Evaluating Database Performance Requirements

You can gather the necessary statistics by using Automatic Workload Repository (AWR) reports or by querying the `GV$SYSSTAT` view. Along with understanding the database performance requirements, you must evaluate the performance capabilities of a storage array.

Choosing Storage

When you understand the performance and capacity requirements, choose a storage platform to meet those requirements.

> **See Also:** *Oracle Database Performance Tuning Guide* for Overview of the Automatic Workload Repository (AWR) and on Generating Automatic Workload Repository Reports

Use Automatic Storage Management (Oracle ASM) to Manage Database Files

Oracle ASM is a vertical integration of both the file system and the volume manager built specifically for Oracle database files. Oracle ASM extends the concept of stripe and mirror everything (SAME) to optimize performance, while removing the need for manual I/O tuning (distributing the data file layout to avoid hot spots). Oracle ASM helps manage a dynamic database environment by letting you grow the database size without shutting down the database to adjust the storage allocation. Oracle ASM also enables low-cost modular storage to deliver higher performance and greater availability by supporting mirroring and striping.

Oracle ASM provides data protection against drive and SAN failures, the best possible performance, and extremely flexible configuration and reconfiguration options. Oracle ASM automatically distributes the data across all available drivers, transparently and dynamically redistributes data when storage is added or removed from the database.

Oracle ASM manages all of your database files. You can phase Oracle ASM into your environment by initially supporting only the fast recovery area.

Note: Oracle recommends host-based mirroring using Oracle ASM.

See:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM
- *Oracle Database Backup and Recovery User's Guide* for information about duplicating a database
- The MAA white papers "Migration to Automatic Storage Management (ASM)" and "Best Practices for Creating a Low-Cost Storage Grid for Oracle Databases" from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa

Use Clustered Oracle ASM to Enable the Storage Grid

The Grid Infrastructure is the software that provides the infrastructure for an enterprise grid architecture. In a cluster, this software includes Oracle Clusterware and Oracle ASM.

You can use clustered Oracle ASM with both Oracle single-instance databases and Oracle Real Application Clusters (Oracle RAC). In an Oracle RAC environment, there is one Oracle ASM instance for each node, and the Oracle ASM instances communicate with each other on a peer-to-peer basis. Only one Oracle ASM instance is required and supported for each node regardless of the number of database instances on the node. Clustering Oracle ASM instances provides fault tolerance, flexibility, and scalability to your storage pool.

See Also: *Oracle Automatic Storage Management Administrator's Guide* for more information about clustered Oracle ASM

Use Oracle Restart for Oracle ASM Instances (non-clustered Oracle Database)

Oracle Restart improves the availability of your Oracle database. When you install the Oracle Grid Infrastructure for a standalone server, it includes both Oracle ASM and

Oracle Restart. Oracle Restart runs out of the Oracle Grid Infrastructure home, which you install separately from Oracle Database homes.

Oracle Restart provides managed startup and restart of a single-instance (non-clustered) Oracle Database, Oracle ASM instance, service, listener, and any other process running on the server. If an interruption of a service occurs after a hardware or software failure, Oracle Restart automatically takes the necessary steps to restart the component.

With Server Control Utility (SRVCTL) you can add a component, such as an Oracle ASM instance to Oracle Restart. You then enable Oracle Restart protection for the Oracle ASM instance. With SRVCTL, you also remove or disable Oracle Restart protection.

See Also:

- *Oracle Database Administrator's Guide* for more information about Oracle Restart
- *Oracle Automatic Storage Management Administrator's Guide* for more information about using Oracle Restart

Oracle ASM Strategic Best Practices

Use the following Oracle ASM strategic best practices:

Use a Simple Disk and Disk Group Configuration

Oracle recommends using Oracle high redundancy disk groups (3 way mirroring) or an external redundancy disk group with equivalent mirroring resiliency for mission critical applications. This higher level of mirroring provides greater protection and better tolerance of different storage failures. This is especially true during planned maintenance windows when a subset of the storage is offline for patching or upgrading. For more information about redundancy, see Chapter , "Use Redundancy to Protect from Disk Failure."

When you use Oracle ASM for database storage, create at least two disk groups: one disk group for the data area and another disk group for the fast recovery area. Also, it is a good practice to have the OCR and voting files in their own disk group:

- *data area*: contains the active database files and other files depending on the level of Oracle ASM redundancy. If Oracle ASM with high redundancy is used, then the data area can also contain OCR, Voting, spfiles, control files, online redo log files, standby redo log files, broker metadata files, and change tracking files used for RMAN incremental backup.

  For example (high redundancy):

  ```
  CREATE DISKGROUP data HIGH REDUNDANCY
  ```

```
FAILGROUP controller1 DISK
     '/devices/c1data01' NAME c1data01,\
     '/devices/c1data02' NAME c1data02
FAILGROUP controller2 DISK
     '/devices/c2data01' NAME c2data01,
     '/devices/c2data02' NAME c2data02
FAILGROUP controller3 DISK
     '/devices/c3data01' NAME c3data01,
     '/devices/c3data02' NAME c3data02
ATTRIBUTE 'au_size'='4M',
     'compatible.asm'  = '11.2',
     'compatible.rdbms'= '11.2',
     'compatible.advm' = '11.2';
```

- *fast recovery area*: contains recovery-related files, such as a copy of the current control file, a member of each online redo log file group, archived redo log files, RMAN backups, and flashback log files.

  For example (normal redundancy):

```
CREATE DISKGROUP reco NORMAL REDUNDANCY
FAILGROUP controller1 DISK
     '/devices/c1reco01' NAME c1reco01,
     '/devices/c1reco02' NAME c1reco02
FAILGROUP controller2 DISK
     '/devices/c2reco01' NAME c2reco01,
     '/devices/c2reco02' NAME c2reco02
ATTRIBUTE 'au_size'='4M',
     'compatible.asm'  = '11.2',
     'compatible.rdbms'= '11.2',
     'compatible.advm' = '11.2';
```

Note 1: If you are using ASMLib in a Linux environment, then create the disks using the `ORACLEASM CREATEDISK` command. ASMLib is a support library for Oracle ASM and is not supported on all platforms. For more information about ASMLib, see Section , "Use ASMLib On Supported Platforms".

For example:

```
/etc/init.d/oracleasm createdisk lun1 /devices/lun01
```

Then, create the disk groups. For example:

```
CREATE DISKGROUP DATA DISK
'ORCL:lun01','ORCL:lun02','ORCL:lun03','ORCL:lun04';
```

Note 2: Oracle recommends using four (4) or more disks in each disk group. Having multiple disks in each disk group spreads kernel contention accessing and queuing for the same disk.

To simplify file management, use Oracle Managed Files to control file naming. Enable Oracle Managed Files by setting the following initialization parameters: `DB_CREATE_FILE_DEST` and `DB_CREATE_ONLINE_LOG_DEST_`*n*.

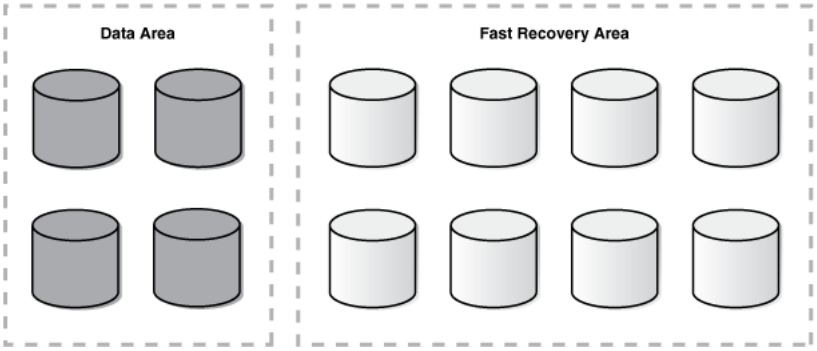
For example:

```
DB_CREATE_FILE_DEST=+DATA
DB_CREATE_ONLINE_LOG_DEST_1=+RECO
```

You have two options when partitioning disks for Oracle ASM:

- Allocate entire disks to the data area and fast recovery area disk groups. Figure 3–1 illustrates allocating entire disks.
- Partition each disk into two partitions, one for the data area and another for the fast recovery area. Figure 3–2 illustrates partitioning each disk into two partitions.

Figure 3–1 Allocating Entire Disks



The advantages of the option shown in Figure 3–1 are:

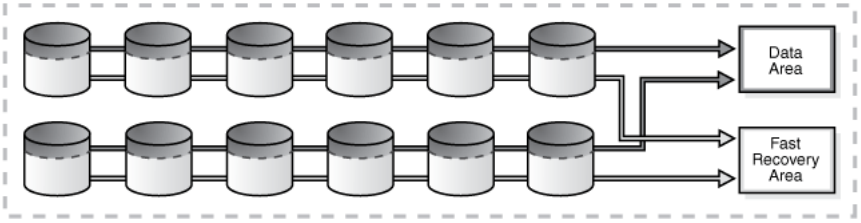
- Easier management of the disk partitions at the operating system level because each disk is partitioned as just one large partition.
- Quicker completion of Oracle ASM rebalance operations following a disk failure because there is only one disk group to rebalance.
- Fault isolation, where storage failures only cause the affected disk group to go offline.
- Patching isolation, where you can patch disks or firmware for individual disks without impacting every disk.

The disadvantage of the option shown in Figure 3–1 is:

- Less I/O bandwidth, because each disk group is spread over only a subset of the available disks.

Figure 3–2 illustrates the partitioning option where each disk has two partitions. This option requires partitioning each disk into two partitions: a smaller partition on the faster outer portion of each drive for the data area, and a larger partition on the slower inner portion of each drive for the fast recovery area. The ratio for the size of the inner and outer partitions depends on the estimated size of the data area and the fast recovery area.

Figure 3–2 Partitioning Each Disk



The advantages of the option shown in Figure 3–2 for partitioning are:

- More flexibility and easier to manage from a performance and scalability perspective.
- Higher I/O bandwidth is available, because both disk groups are spread over all available spindles. This advantage is considerable for the data area disk group for I/O intensive applications.
- There is no need to create a separate disk group with special, isolated storage for online redo logs or standby redo logs if you have sufficient I/O capacity.
- You can use the slower regions of the disk for the fast recovery area and the faster regions of the disk for data.

The disadvantages of the option shown in Figure 3–2 for partitioning are:

- A double partner disk failure will result in loss of both disk groups, requiring the use of a standby database or tape backups for recovery. This problem is eliminated when using high redundancy ASM disk groups.
- An Oracle ASM rebalance operation following a disk failure is longer, because both disk groups are affected.

See Also:

- *Oracle Database 2 Day DBA* for an Overview of Disks, Disk Groups, and Failure Groups and a description of normal redundancy, high redundancy and external redundancy
- *Oracle Database Backup and Recovery User's Guide* for details about setting up and sizing the fast recovery area
- *Oracle Automatic Storage Management Administrator's Guide* for details about Oracle ASM

Use Redundancy to Protect from Disk Failure

When setting up redundancy to protect from hardware failures, there are two options to consider:

- Storage Array Based RAID
- Oracle ASM Redundancy

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for an overview of Oracle Automatic Storage Management
- *Oracle Automatic Storage Management Administrator's Guide* for information about creating disk groups

Storage Array Based RAID

If you are using a high-end storage array that offers robust *built-in* RAID solutions, then Oracle recommends that you configure redundancy in the storage array by enabling RAID protection, such as RAID1 (mirroring) or RAID5 (striping plus parity). For example, to create an Oracle ASM disk group where redundancy is provided by the storage array, first create the RAID-protected **logical unit numbers (LUNs)** in the storage array, and then create the Oracle ASM disk group using the `EXTERNAL REDUNDANCY` clause:

```
CREATE DISKGROUP DATA EXTERNAL REDUNDANCY DISK
```

```
'/devices/lun1','/devices/lun2','/devices/lun3','/devices/lun4';
```

You must ensure that the storage array provides true redundancy among physical disks. If virtual disks are used, you must ensure that redundancy is established with mirror copies on that are on separate physical disks.

External redundancy is not supported for Exadata systems; you must choose either normal or high redundancy disk groups.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM Mirroring and Disk Group Redundancy
- *Oracle Database 2 Day DBA* for information about Creating a Disk Group

Oracle ASM Redundancy

Oracle ASM provides redundancy with the use of failure groups, which are defined during disk group creation. The disk group type determines how Oracle ASM mirrors files. When you create a disk group, you indicate whether the disk group is a normal redundancy disk group (2-way mirroring for most files by default), a high redundancy disk group (3-way mirroring), or an external redundancy disk group (no mirroring by Oracle ASM). You use an external redundancy disk group if your storage system does mirroring at the hardware level, or if you have no need for redundant data. The default disk group type is normal redundancy. After a disk group is created the redundancy level cannot be changed.

The main advantages of using ASM level redundancy are:

- Removal of hot spots due to striping across all disks
- Rebalancing and redistribution of data and I/O is simpler when adding or dropping disks
- Auto repair of data corruption when ASM leverages a good extent on another mirror copy

When a corrupted data block is detected and ASM level redundancy is used then Oracle will automatically check its mirror. If the mirror is uncorrupted then the application does not get an error and Oracle ASM automatically fixes the corrupted block in the initial mirror. Oracle Exadata and SuperCluster use ASM level redundancy for the benefits mentioned in the list above.

Failure group definition is specific to each storage setup, but you should follow these guidelines:

- If every disk is available through every I/O path, as would be the case if using disk multipathing software, then keep each disk in its own failure group. This is the default Oracle ASM behavior if creating a disk group without explicitly defining failure groups.

```
CREATE DISKGROUP DATA NORMAL REDUNDANCY DISK
    '/devices/diska1','/devices/diska2','/devices/diska3','/devices/diska4',
    '/devices/diskb1','/devices/diskb2','/devices/diskb3','/devices/diskb4';
```

- For an array with two controllers where every disk is seen through both controllers, create a disk group with each disk in its own failure group:

```
CREATE DISKGROUP DATA NORMAL REDUNDANCY
  DISK
```

```
'/devices/diska1','/devices/diska2','/devices/diska3','/devices/diska4',
'/devices/diskb1','/devices/diskb2','/devices/diskb3','/devices/diskb4';
```

- If every disk is not available through every I/O path, then define failure groups to protect against the piece of hardware that you are concerned about failing. Here are some examples:
  - For an array with two controllers where each controller sees only half the drives, create a disk group with two failure groups, one for each controller, to protect against controller failure:

```
CREATE DISKGROUP DATA NORMAL REDUNDANCY
  FAILGROUP controller1 DISK
   '/devices/diska1','/devices/diska2','/devices/diska3','/devices/diska4'
  FAILGROUP controller2 DISK
   '/devices/diskb1','/devices/diskb2','/devices/diskb3','/devices/diskb4';
```

  - For a storage network with multiple storage arrays, you want to mirror across storage arrays, then create a disk group with two failure groups, one for each array, to protect against array failure:

```
CREATE DISKGROUP DATA NORMAL REDUNDANCY
  FAILGROUP array1 DISK
   '/devices/diska1','/devices/diska2','/devices/diska3','/devices/diska4'
  FAILGROUP array2 DISK
   '/devices/diskb1','/devices/diskb2','/devices/diskb3','/devices/diskb4';
```

When determining the proper size of a disk group that is protected with Oracle ASM redundancy, enough free space must exist in the disk group so that when a disk fails Oracle ASM can automatically reconstruct the contents of the failed drive to other drives in the disk group while the database remains online. The amount of space required to ensure Oracle ASM can restore redundancy following disk failure is in the column `REQUIRED_MIRROR_FREE_MB` in the `V$ASM_DISKGROUP` view. The amount of free space that you can use safely in a disk group, taking mirroring into account, and still be able to restore redundancy after a disk failure is in the `USABLE_FILE_MB` column in the `V$ASM_DISKGROUP` view. The value of the `USABLE_FILE_MB` column should always be greater than zero. If `USABLE_FILE_MB` falls below zero, then add more disks to the disk group.

> **See:** *Oracle Database 2 Day DBA* for information about Creating a Disk Group

Oracle ASM in the Grid Infrastructure Home

Oracle Automatic Storage Management (Oracle ASM) and Oracle Clusterware are installed into a single home directory, which is called the Grid home. Oracle Grid Infrastructure for a cluster software refers to the installation of the combined products. The Grid home is separate from the home directories of other Oracle software products installed on the same server.

> **See Also:** *Oracle Database 2 Day + Real Application Clusters Guide* for information about Oracle ASM and the Grid home.

Ensure Disks in the Same Disk Group Have the Same Characteristics

Although same sized LUNs and the same capacity of Failure Groups is enforced in Oracle Database 12*c*, and ensuring that all disks in the same disk group have the same size and performance characteristics is not required, doing so provides more predictable overall performance and space utilization. When possible, present physical

disks (spindles) to Oracle ASM as opposed to Logical Unit Numbers (LUNs) that create a layer of abstraction between the disks and Oracle ASM.

If the disks are the same size, then Oracle ASM spreads the files evenly across all of the disks in the disk group. This allocation pattern maintains every disk at the same capacity level and ensures that all of the disks in a disk group have the same I/O load. Because Oracle ASM load balances workload among all of the disks in a disk group, different Oracle ASM disks should not share the same physical drive.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for complete information about administering Oracle ASM disk groups

Use Failure Groups When Using Oracle ASM Redundancy

Using failure groups to define a common failure component ensures continuous access to data when that component fails. For maximum protection, use at least three failure groups for normal redundancy and at least five failure groups for high redundancy. Doing so enables Oracle ASM to tolerate multiple failure group failures and avoids the confusing state of having Oracle ASM running without full redundancy.

> **Note:** If you have purchased a high-end storage array that has redundancy features built in, then you can optionally use those features from the vendor to perform the mirroring protection functions and set the Oracle ASM disk group to external redundancy. Along the same lines, use Oracle ASM normal or high redundancy with low-cost storage and Exadata storage.

Use Intelligent Data Placement

Intelligent Data Placement enables you to specify disk regions on Oracle ASM disks for best performance. Using the disk region settings you can ensure that frequently accessed data is placed on the outermost (hot) tracks which have greater speed and higher bandwidth. In addition, files with similar access patterns are located physically close, reducing latency. Intelligent Data Placement also enables the placement of primary and mirror extents into different hot or cold regions.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for more information about Intelligent Data Placement

Use Oracle ACFS to Manage Files Outside the Database

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support customer files maintained outside of Oracle Database. Oracle ACFS includes a volume management service and comes with fine grained security policies, encryption, snapshotting and replication.

Oracle ACFS supports many database and application files, including executables, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and other general-purpose application file data.

Note: Oracle database binaries can be put on Oracle ACFS but not binaries in the Grid Infrastructure home.

See Also: *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle ACFS

Oracle ASM Configuration Best Practices

Use the following Oracle ASM configuration best practices:

- Use Disk Multipathing Software to Protect from Path Failure
- Set the PROCESSES Initialization Parameter
- Use Disk Labels
- Set the FAILGROUP_REPAIR_TIME or DISK_REPAIR_TIME Disk Group Attribute Appropriately
- Use ASMLib On Supported Platforms

Use Disk Multipathing Software to Protect from Path Failure

Disk multipathing software aggregates multiple independent I/O paths into a single logical path. The path abstraction provides I/O load balancing across host bus adapters (HBA) and nondisruptive failovers when there is a failure in the I/O path. You should use disk multipathing software with Oracle ASM.

When specifying disk names during disk group creation in Oracle ASM, use the logical device representing the single logical path. For example, when using Device Mapper on Linux 2.6, a logical device path of `/dev/dm-0` may be the aggregation of physical disks `/dev/sdc` and `/dev/sdh`. Within Oracle ASM, the `ASM_DISKSTRING` parameter should contain `/dev/dm-*` to discover the logical device `/dev/dm-0`, and that logical device is necessary during disk group creation:

```
asm_diskstring='/dev/dm-*'

CREATE DISKGROUP DATA DISK
'/dev/dm-0','/dev/dm-1','/dev/dm-2','/dev/dm-3';
```

Note: For more information about using the combination of ASMLib and Multipath Disks, see "Configuring Oracle ASMLib on Multipath Disks" in My Oracle Support Note 309815.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=309815.1

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM and Multipathing
- For more information, see "Oracle ASM and Multi-Pathing Technologies" in My Oracle Support Note 294869.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=294869.1

Set the PROCESSES Initialization Parameter

The PROCESSES initialization parameter affects Oracle ASM, but the default value is usually suitable. However, if multiple database instances are connected to an Oracle ASM instance, you can use the following formulas:

For < 10 instances per node...	For > 10 instances per node...
`PROCESSES = 50 * (n + 1)`	`PROCESSES = 50 * MIN (n + 1, 11) + 10 * MAX (n - 10, 0)`

where n is the number database instances connecting to the Oracle ASM instance.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM Parameter Setting Recommendations
- *Oracle Database Administrator's Guide* for more information about setting the PROCESSES initialization parameter
- *Oracle Database Reference* for more information about the PROCESSES parameter

Use Disk Labels

Disk labels ensure consistent access to disks across restarts. ASMLib is the preferred tool for disk labeling. For more information about ASMLib, see Section , "Use ASMLib On Supported Platforms".

Set the FAILGROUP_REPAIR_TIME or DISK_REPAIR_TIME Disk Group Attribute Appropriately

In Oracle Database 12*c* the disk group attribute FAILGROUP_REPAIR_TIME governs the amount of time an entire failure group is offline before dropping the disks in the failure group. This defaults to 24 hours to give enough time to repair more complex components. Change this time if the time to repair such components will take longer than 24 hours but still less time than it would take to completely rebalance the impacted diskgroups.

The DISK_REPAIR_TIME disk group attribute specifies how long a disk remains offline before Oracle ASM drops the disk. If a disk is made available before the DISK_REPAIR_TIME parameter has expired, the storage administrator can issue the ONLINE DISK command and Oracle ASM resynchronizes the stale data from the mirror side. The online disk operation does not restart if there is a failure of the instance on which the disk is running. You must reissue the command manually to bring the disk online.

You can set a disk repair time attribute on your disk group to specify how long disks remain offline before being dropped. The appropriate setting for your environment depends on how long you expect a typical transient type of failure to persist.

Set the DISK_REPAIR_TIME disk group attribute to the maximum amount of time before a disk is definitely considered to be out of service.

See Also: *Oracle Automatic Storage Management Administrator's Guide* for information about restoring the redundancy of an Oracle ASM disk group after a transient disk path failure

Use ASMLib On Supported Platforms

To improve manageability use ASMLib on platforms where it is available. ASMLib is a support library for Oracle ASM.

Although ASMLib is not required to run Oracle ASM, using ASMLib is beneficial because ASMLib:

- Eliminates the need for every Oracle process to open a file descriptor for each Oracle ASM disk, thus improving system resource usage.
- Simplifies the management of disk device names, makes the discovery process simpler, and removes the challenge of having disks added to one node and not be known to other nodes in the cluster.
- Eliminates the impact when the mappings of disk device names change upon system restart.

Note: ASMLib is not supported on all platforms.

See Also:

- *Oracle Database 2 Day + Real Application Clusters Guide* for more information about installing ASMLib
- Oracle ASMLib website at

  http://www.oracle.com/technetwork/topics/linux/asmlib/index-101839.html

Configure LUNs Properly

The number of LUNs (Oracle ASM disks) for each disk group should be at least equal to four times the number of active I/O paths. For example, if a disk group has two active I/O paths, then minimum of eight LUNs should be used. The LUNs should be of equal size and performance for each disk group.

An I/O path is a distinct channel or connection between storage presenting LUNs and the server. An active I/O path is an I/O path in which the I/O load on a LUN is multiplexed through multipathing software.

Use Disks with Similar Characteristics and Capacity

Ensure that all Oracle ASM disks in a disk group have similar storage performance and availability characteristics. In storage configurations with mixed speed drives, such as flash memory and hard disk drives (HDD) I/O performance is constrained by the slowest speed drive.

Oracle ASM data distribution policy is capacity-based. Ensure that Oracle ASM disks in a disk group have the same capacity to maintain balance.

Minimize I/O contention between Oracle ASM disks and other applications by dedicating disks in Oracle ASM disk groups.

Choose a hardware RAID stripe size that is a power of 2 and less than or equal to the size of the Oracle ASM allocation unit.

Configure Disk Groups Properly

Configure a minimum of three failure groups for normal redundancy disk groups and five failure groups for high redundancy disk groups to maintain the necessary number of copies of the Partner Status Table (PST) to ensure robustness with respect to storage hardware failures.

Create external redundancy disk groups when using high-end storage arrays. High-end storage arrays generally provide hardware RAID protection. Ensure the storage array is properly configured to provide true disk redundancy.

Use Oracle ASM mirroring redundancy when not using hardware RAID, or when you need host-based volume management functionality, such as mirroring across storage systems. You can use Oracle ASM mirroring in configurations when mirroring between geographically-separated sites (extended clusters).

Configure two disk groups, one for data and the other for the fast recovery area.

> **See Also:** Section , "Use Failure Groups When Using Oracle ASM Redundancy"

Oracle ASM Operational Best Practices

Use the following Oracle ASM operational best practices:

- Use SYSASM for Oracle ASM Authentication
- Set Rebalance Power Limit to the Maximum Limit that Does Not Affect Service Levels
- Use a Single Command to Mount Multiple Disk Groups
- Use a Single Command to Add or Remove Storage
- Check Disk Groups for Imbalance
- Proactively Mine Vendor Logs for Disk Errors
- Use ASMCMD Utility to Ease Manageability of Oracle ASM
- Use Oracle ASM Configuration Assistant (ASMCA)

Use SYSASM for Oracle ASM Authentication

The Oracle ASM instance is managed by a privileged role called `SYSASM`, which grants full access to Oracle ASM disk groups. Using `SYSASM` enables the separation of authentication for the storage administrator and the database administrator. By configuring a separate operating system group for Oracle ASM authentication, you can have users that have `SYSASM` access to the Oracle ASM instances and do not have `SYSDBA` access to the database instances.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for information about authentication to access Oracle ASM instances

Set Rebalance Power Limit to the Maximum Limit that Does Not Affect Service Levels

Higher Oracle ASM rebalance power limits make a rebalance operation run faster but can also affect application service levels. Rebalancing takes longer with lower power values, but consumes fewer processing and I/O resources that are shared by other applications, such as the database.

After performing planned maintenance, for example adding or removing storage, it is necessary to subsequently perform a rebalance to spread data across all of the disks. There is a power limit associated with the rebalance. You can set a power limit to specify how many processes perform the rebalance. If you do not want the rebalance to impact applications, then set the power limit lower. However, if you want the rebalance to finish quickly, then set the power limit higher. To determine the default power limit for rebalances, check the value of the `ASM_POWER_LIMIT` initialization parameter in the Oracle ASM instance.

If the `POWER` clause is not specified in an `ALTER DISKGROUP` statement, or when rebalance is run implicitly when you add or drop a disk, then the rebalance power defaults to the value of the `ASM_POWER_LIMIT` initialization parameter. You can adjust the value of this parameter dynamically.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for more information about rebalancing Oracle ASM disk groups

Use a Single Command to Mount Multiple Disk Groups

Mounting multiple disk groups in the same command ensures that disk discovery runs only one time, thereby increasing performance. Disk groups that are specified in the `ASM_DISKGROUPS` initialization parameter are mounted automatically at Oracle ASM instance startup.

To mount disk groups manually, use the `ALTER DISKGROUP...MOUNT` statement and specify the `ALL` keyword:

```
ALTER DISKGROUP ALL MOUNT;
```

Note: The `ALTER DISKGROUP...MOUNT` command only works on one node. For cluster installations use the following command:

```
srvctl start diskgroup -g
```

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for information about mounting and dismounting disk groups

Use a Single Command to Add or Remove Storage

Oracle ASM permits you to add or remove disks from your disk storage system while the database is operating. When you add a disk to a disk group, Oracle ASM automatically redistributes the data so that it is evenly spread across all disks in the disk group, including the new disk. The process of redistributing data so that it is also spread across the newly added disks is known as *rebalancing*. By executing storage maintenance commands in the same command, you ensure that only one rebalance is required to incur minimal impact to database performance.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for information about Altering Disk Groups

Check Disk Groups for Imbalance

You should periodically check disk groups for imbalance. Occasionally, disk groups can become unbalanced if certain operations fail, such as a failed rebalance operation. Periodically checking the balance of disk groups and running a manual rebalance, if needed, ensures optimal Oracle ASM space utilization and performance.

Use the following methods to check for disk group imbalance:

- To check for an imbalance on all mounted disk groups, see "Script to Report the Percentage of Imbalance in all Mounted Diskgroups" in My Oracle Support Note 367445.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=367445.1

- To check for an imbalance from an I/O perspective, query the statistics in the V$ASM_DISK_IOSTAT view before and after running a large SQL*Plus statement. For example, if you run a large query that performs only read I/O, the READS and BYTES_READ columns should be approximately the same for all disks in the disk group.

Proactively Mine Vendor Logs for Disk Errors

You should proactively mine vendor logs for disk errors and have Oracle ASM move data off the bad disk spots. Disk vendors usually provide disk-scrubbing utilities that notify you if any part of the disk is experiencing problems, such as a media sense error. When a problem is found, use the ASMCMD utility REMAP command to move Oracle ASM extents from the bad spot to a good spot.

Note that this is only applicable for data that is not accessed by the database or Oracle ASM instances, because in that case Oracle ASM automatically moves the extent experiencing the media sense error to a different location on the same disk. In other words, use the ASMCMD utility REMAP command to proactively move data from a bad disk spot to a good disk spot before that data is accessed by the application.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for information about the ASMCMD utility

Use ASMCMD Utility to Ease Manageability of Oracle ASM

Use the ASMCMD utility to ease the manageability of day-to-day storage administration. Use the ASMCMD utility to view and manipulate files and directories in Oracle ASM disk groups and to list the contents of disk groups, perform searches, create and remove directories and aliases, display space usage. Also, use the ASMCMD utility to backup and restore the metadata of the disk groups (using the md_backup and md_restore commands).

> **Note:** As a best practice to create and drop Oracle ASM disk groups, use SQL*Plus, ASMCA, or Oracle Enterprise Manager.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for more information about ASMCMD Disk Group Management Commands

Use Oracle ASM Configuration Assistant (ASMCA)

Oracle ASM Configuration Assistant (ASMCA) supports installing and configuring Oracle ASM instances, disk groups, volumes, and Oracle Automatic Storage Management Cluster File System (Oracle ACFS). In addition, you can use the ASMCA command-line interface as a silent mode utility.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide*

Use Oracle Storage Grid

The Oracle Storage Grid consists of either:

- Oracle ASM and third-party storage using external redundancy.
- Oracle ASM and Oracle Exadata or third-party storage using Oracle ASM redundancy. The Oracle Storage Grid with Exadata seamlessly supports MAA-related technology, improves performance, provides unlimited I/O scalability, is easy to use and manage, and delivers mission-critical availability and reliability to your enterprise.

See Also: *Oracle Automatic Storage Management Administrator's Guide*

Oracle Storage Grid Best Practices for Unplanned Outages

To protect storage against unplanned outages:

- Set the `DB_BLOCK_CHECKSUM` initialization parameter to `TYPICAL` (default) or `FULL` in each database. For more information, see Section , "Set `DB_BLOCK_CHECKSUM=FULL` and `DB_BLOCK_CHECKING=MEDIUM` or `FULL`".

  Note: Oracle Exadata Database Machine also prevents corruptions from being written to disk by incorporating the hardware assisted resilient data (HARD) technology in its software. HARD uses block checking, in which the storage subsystem validates the Oracle block contents, preventing corrupted data from being written to disk. HARD checks in Oracle Exadata operate completely transparently and no parameters must be set for this purpose at the database or storage tier. For more information see the White Paper "Optimizing Storage and Protecting Data with Oracle Database 11*g*" at

  `http://www.oracle.com/us/products/database/database-11g-managing-storage-wp-354099.pdf`

- Choose Oracle ASM redundancy type (`NORMAL` or `HIGH`) based on your desired protection level and capacity requirements

  The `NORMAL` setting stores two copies of Oracle ASM extents, while the `HIGH` setting stores three copies of Oracle ASM extents. Normal redundancy provides more usable capacity and high redundancy provides more protection.

- If a storage component is to be offlined when one or more databases are running, then verify that taking the storage component offline does not impact Oracle ASM disk group and database availability. Before dropping a failure group or offlining a storage component perform the appropriate checks.

- Ensure I/O performance can be sustained after an outage

  Ensure that you have enough I/O bandwidth to support your service-level agreement if a failure occurs. For example, a typical case for a Storage Grid with *n* storage components would be to ensure that *n*-1 storage components could support the application service levels (for example, to handle a storage component failure).

- Ensure that the disk group attribute content.type is set properly on Exadata or SuperCluster.

  This setting helps constrain the impact of losing two disks to a single disk group. The content.type attribute should be set to "data" for diskgroups containing

datafiles. It should be set to "recovery" for diskgroups containing the Fast Recovery Area. On the DBFS_DG disk group it should be set to "system".

Oracle Storage Grid Best Practices for Planned Maintenance

Use the following list of best practices for planned maintenance:

- Size I/O for performance first, and then set it for capacity:

  When building your Oracle Storage Grid, make sure you have enough drives to support I/O's per second and MBs per second to meet your service-level requirements. Then, make sure you also have enough capacity. The order is important because you do not want to buy enough drives to support capacity but then find the system cannot meet your performance requirements.

  When you are sizing, you must consider what happens to performance when you offline a subset of storage for planned maintenance. For Example, when a subset of the overall storage is offlined you still must make sure you get the required number of IOPS if that is important to meet your SLAs. Also, if offlining storage means the system cannot add more databases, then one has to consider that upfront.

- Set Oracle ASM power limit for faster rebalancing For more information, see Section , "Set Rebalance Power Limit to the Maximum Limit that Does Not Affect Service Levels".

  See Also:

  - *Oracle Automatic Storage Management Administrator's Guide* for information about the `ASM_POWER_LIMIT` initialization parameter
  - *Oracle Automatic Storage Management Administrator's Guide* for information about tuning rebalance operations
  - *Oracle Database Reference* for more information about the `ASM_POWER_LIMIT` initialization parameter

4

Configuring Oracle Database

This chapter describes best practices for configuring all Oracle databases, including single-instance, Oracle RAC databases, Oracle RAC One Node databases, and the primary and standby databases in Oracle Data Guard configurations (for more information about High Availability architectures, see *Oracle Database High Availability Overview*). Adopt these best practices to reduce or avoid outages, reduce the risk of corruption, and to improve recovery performance.

This chapter contains the following topics:

> **See Also:** *Oracle Database High Availability Overview* for more information about high availability architectures

Database Configuration High Availability and Fast Recoverability Best Practices

To reduce recovery time and increase database availability and redundancy:

Set the Database ARCHIVELOG Mode and FORCE LOGGING Mode

Running the database in `ARCHIVELOG` mode and using database `FORCE LOGGING` mode are prerequisites for database recovery operations. The `ARCHIVELOG` mode enables online database backup and is necessary to recover the database to a point in time later than what has been restored. Features such as Oracle Data Guard and Flashback Database require that the production database run in `ARCHIVELOG` mode.

If you can isolate data that never needs to be recovered within specific tablespaces, then you can use tablespace level `FORCE LOGGING` attributes instead of the database `FORCE LOGGING` mode.

See Also:

- *Oracle Database Administrator's Guide* for more information about controlling archiving mode
- *Oracle Database Administrator's Guide* for information about Specifying `FORCE LOGGING` Mode
- See "Reduce Overhead and Redo Volume During ETL Operations in the technical white paper, "Oracle Data Guard: Disaster Recovery for Oracle Exadata Database Machine" from the MAA Best Practices area for Exadata Database Machine at

  http://www.oracle.com/goto/maa

Configure the Size of Redo Log Files and Groups Appropriately

Use Oracle log multiplexing to create multiple redo log members in each redo group, one in the data area and one in the Fast Recovery Area (unless the redo logs are in an Oracle ASM high redundancy disk group). This protects against a failure involving the redo log, such as a disk or I/O failure for one member, or a user error that accidentally removes a member through an operating system command. If at least one redo log member is available, then the instance can continue to function.

To size redo log files and groups:

- Use a minimum of three redo log groups: this helps prevent the log writer process (LGWR) from waiting for a group to be available following a log switch.
- All online redo logs and standby redo logs are equal size.
- Use redo log size = (peak redo rate MB/sec) * (1200 secs or 20 mins).

  Start with default 4 GB redo size for larger workloads and 100MB for smaller workloads.
- Locate redo logs on high performance disks.
- Log switch every 15 minutes during peak.
- Place log files in a high redundancy disk group, or multiplex log files across different normal redundancy disk groups, if using ASM redundancy.

Note: Do not multiplex the standby redo logs.

See Also:

- Chapter 8, " Configuring Oracle Data Guard"
- *Oracle Database Administrator's Guide* for more information about managing redo logs
- *Oracle Database Administrator's Guide* for information about Multiplexing Redo Log Files
- *Oracle Data Guard Concepts and Administration* for more information about online, archived, and standby redo log files

Use a Fast Recovery Area

The Fast Recovery Area is Oracle managed disk space that provides a centralized disk location for backup and recovery files.

The Fast Recovery Area is defined by setting the following database initialization parameters:

- `DB_RECOVERY_FILE_DEST`: specifies the default location for the fast recovery area.
- `DB_RECOVERY_FILE_DEST_SIZE`: specifies (in bytes) the hard limit on the total space to be used by database recovery files created in the recovery area location.

The Oracle Suggested Backup Strategy described in the *Oracle Database 2 Day DBA* recommends using the fast recovery area as the primary location for recovery. When the fast recovery area is properly sized, files needed for repair are readily available. The minimum recommended disk limit is the combined size of the database, incremental backups, all archived redo logs that have not been copied to tape, and flashback logs.

See Also:

- *Oracle Database Administrator's Guide* for information about Specifying a Fast Recovery Area
- *Oracle Database Backup and Recovery User's Guide* for detailed information about sizing the fast recovery area and setting the retention period
- *Oracle Database 2 Day DBA*

Enable Flashback Database

Flashback Database provides an efficient alternative to point-in-time recovery for reversing unwanted database changes. Flashback Database enables you to rewind an entire database backward in time, reversing the effects of database changes within a time window. The effects are similar to database point-in-time recovery (DBPITR). You can flash back a database by issuing a single RMAN command or a SQL*Plus statement instead of using a complex procedure.

To enable Flashback Database, you configure a fast recovery area and set a flashback retention target. This retention target specifies how far back you can rewind a database with Flashback Database. For more information about specifying a fast recovery area, see Section , "Use a Fast Recovery Area".

When configuring and enabling Flashback Database:

- Know your application performance baseline before you enable flashback to help determine the overhead and to assess the application workload implications of turning on flashback database.
- Ensure the fast recovery area space is sufficient to hold the flashback database flashback logs. For more information about sizing the fast recovery area, see the *Oracle Database Backup and Recovery User's Guide*. A general rule of thumb is to note that the volume of flashback log generation is approximately the same order of magnitude as redo log generation. For example, if you intend to set `DB_FLASHBACK_RETENTION_TARGET` to 24 hours, and if the database generates 20 GB of redo in a day, then a rule of thumb is to allow 20 GB to 30 GB disk space for the flashback logs. The same rule applies for guaranteed restore points. For example, if the database generates 20 GB redo every day, and if the guaranteed restore point will be kept for a day, then plan to allocate 20 to 30 GB.

  - An additional method to determine fast recovery area sizing is to enable flashback database and allow the database to run for a short period (2-3 hours). The estimated amount of space required for the fast recovery area can be retrieved by querying `V$FLASHBACK_DATABASE_STAT.ESTIMATED_FLASHBACK_SIZE`.
  - Note that the `DB_FLASHBACK_RETENTION_TARGET` is a target and there is no guarantee that you can flashback the database that far. In some cases if there is space pressure in the fast recovery area where the flashback logs are stored then the oldest flashback logs may be deleted. For a detailed explanation of the fast recovery area deletion rules see the *Oracle Database Backup and Recovery User's Guide*. To guarantee a flashback point-in-time you must use guaranteed restore points.

- Set the Oracle Enterprise Manager monitoring metric, "Recovery Area Free Space (%)" for proactive alerts of space issues with the fast recovery area.
- Ensure there is sufficient I/O bandwidth to the fast recovery area. Insufficient I/O bandwidth with flashback database on is usually indicated by a high occurrence of the `"FLASHBACK BUF FREE BY RVWR"` wait event in an Automatic Workload Repository (AWR) report.
- Set the `LOG_BUFFER` initialization parameter to at least 8 MB to give flashback database more buffer space in memory. For large databases with more than a 4GB SGA, you may consider setting `LOG_BUFFER` to values in the range of 32-64 MB (for more information about `LOG_BUFFER` and valid values on 32-bit and 64-bit operating systems, see *Oracle Database Reference*).
- If you have a Data Guard standby database, always set `DB_FLASHBACK_RETENTION_TARGET` to the same value on the standby database(s) as the primary. Set `DB_FLASHBACK_RETENTION_TARGET` initialization parameter to the largest value prescribed by any of the following conditions that apply:
  - To leverage flashback database to reinstate your failed primary database after Data Guard failover, for most cases set `DB_FLASHBACK_RETENTION_TARGET` to a minimum of 60 (mins) to enable reinstatement of a failed primary.
  - Consider cases where there are multiple outages, for example, first a network outage, followed later by a primary database outage, that may result in a transport lag between primary and standby database at failover time. For such cases set `DB_FLASHBACK_RETENTION_TARGET` to a value equal to the sum of 60 (mins) plus the maximum transport lag to accommodate. This ensures that the failed primary database can be flashed back to an SCN that precedes the SCN at which the standby became primary - a requirement for primary reinstatement.
  - If using Flashback Database for fast point in time recovery from user error or logical corruptions, set `DB_FLASHBACK_RETENTION_TARGET` to a value equal to the farthest time in the past to which the database should be recovered.
- Review *Oracle Database Backup and Recovery User's Guide* for information about Configuring the Fast Recovery Area.
- To monitor the progress of a flashback database operation you can query the `V$SESSION_LONGOPS` view. An example query to monitor progress is:

```
select sofar, totalwork, units FROM v$session_longops WHERE opname = 'Flashback
Database';
```

- For repetitive tests where you must flashback to the same point, use Flashback database guaranteed restore points (GRP) instead of enabling flashback database to minimize space utilization.
- In general, the performance effect of enabling Flashback Database is minimal. In Oracle Database 11*g* Release 2 and later there are significant performance enhancements to nearly eliminate any overhead when you first enable flashback database, and during batch direct loads. For more information, see "Flashback Database Best Practices & Performance" in My Oracle Support Note 565535.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=565535.1

  See Also:

  - *Oracle Database Backup and Recovery User's Guide* for more information about guaranteed restore points and Flashback Database
  - *Oracle Database Backup and Recovery User's Guide* for information about configuring the environment for optimal Flashback Database performance
  - *Oracle Database Backup and Recovery User's Guide* for information about configuring Oracle Flashback Database and Restore Points
  - *Oracle Data Guard Concepts and Administration* for information about Using Flashback Database After a Role Transition
  - *Oracle Data Guard Concepts and Administration* for information about Converting a Failed Primary Into a Standby Database Using Flashback Database
  - *Oracle Database 2 Day + Performance Tuning Guide* for information about Gathering Database Statistics Using the Automatic Workload Repository (AWR)
  - The MAA white paper "Active Data Guard 11g Best Practices (includes best practices for Redo Apply)" for more information about media recovery best practices from the MAA Best Practices area for Oracle Database at

    http://www.oracle.com/goto/maa

Set FAST START MTTR TARGET Initialization Parameter

The Fast-Start Fault Recovery feature reduces the time required to recover from a crash and makes the recovery bounded and predictable by limiting the number of dirty buffers and the number of redo records generated between the most recent redo record and the last checkpoint.

Set the `FAST_START_MTTR_TARGET` initialization parameter to control instance recovery time. With the Fast-Start Fault Recovery feature, the `FAST_START_MTTR_TARGET` initialization parameter simplifies the configuration of recovery time from instance or system failure. This parameter specifies a target for the expected recovery time objective (RTO), which is the time, in seconds, that it should take to start the instance and perform cache recovery. When you set this parameter, the database manages incremental checkpoint writes in an attempt to meet the target. If you have chosen a practical value for this parameter, then you can expect your database to recover, on average, in approximately the number of seconds you have chosen.

Initially, set the FAST_START_MTTR_TARGET initialization parameter to 300 (seconds) or to the value required for your expected recovery time objective (RTO).

Outage testing for cases such as for node or instance failures during peak loads is recommended.

See Also:

- *Oracle Database Performance Tuning Guide* for information about Tuning Instance Recovery Performance: Fast-Start Fault Recovery
- The MAA white paper "Optimizing Availability During Unplanned Outages Using Oracle Clusterware and Oracle RAC" for more best practices from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa

Protect Against Data Corruption

A data block is corrupted when it is not in a recognized Oracle Database format, or its contents are not internally consistent. Data block corruption can damage internal Oracle control information or application and user data, leading to crippling loss of critical data and services. The Oracle Database corruption prevention, detection, and repair capabilities are built on internal knowledge of the data and transactions it protects, and on the intelligent integration of its comprehensive high availability solutions. For more information about recovery from data corruption, see Section , "Recovering from Data Corruption".

Once the corruption is detected, Oracle offers Data Guard, block media recovery, and data file media recovery to recover the data. Database-wide logical corruptions caused by human or application errors can be undone with Oracle Flashback Technologies. Tools are also available for proactive validation of logical data structures. For example, the SQL*Plus ANALYZE TABLE statement detects inter-block corruptions.

See Also:

- "Preventing, Detecting, and Repairing Block Corruption: Database 11g" MAA white paper from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa

- "Best Practices for Corruption Detection, Prevention, and Automatic Repair - in a Data Guard Configuration" in My Oracle Support Note 1302539.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1302539.1

Preventing Widespread Data Corruption

To achieve the most comprehensive data corruption prevention and detection:

- Use Oracle Data Guard with physical standby databases to prevent widespread block corruption. Oracle Data Guard is the best solution for protecting Oracle data against data loss and corruption, and lost writes. For more information, see Section , "General Data Guard Configuration Best Practices".
- Set the Oracle Database block-corruption initialization parameters on the Data Guard primary and standby databases:

On the Primary database set...	On the Standby databases set...
DB_BLOCK_CHECKSUM=FULL DB_LOST_WRITE_PROTECT=TYPICAL DB_BLOCK_CHECKING=FULL	DB_BLOCK_CHECKSUM=FULL DB_LOST_WRITE_PROTECT=TYPICAL DB_BLOCK_CHECKING=FULL

Performance overhead is incurred on every block change, therefore performance testing is of particular importance when setting the DB_BLOCK_CHECKING parameter. Oracle highly recommends the **minimum** setting of DB_BLOCK_CHECKING=MEDIUM (block checks on data blocks but not index blocks) on either the primary or standby database. If the performance overhead of enabling DB_BLOCK_CHECKING to MEDIUM or FULL is unacceptable on your primary database, then set DB_BLOCK_CHECKING to MEDIUM or FULL for your standby databases.

Caution: A thorough performance assessment is recommended when changing these settings.

See Also:

- Section , "Use Data Guard Redo Apply Best Practices"
- For the most current information, see the My Oracle Support Note 1302539.1: "Best Practices for Corruption Detection, Prevention, and Automatic Repair - in a Data Guard Configuration" at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1302539.1

- Use Oracle Automatic Storage Management (Oracle ASM) to provide disk mirroring to protect against disk failures. For more information, see Section , "Use Automatic Storage Management (Oracle ASM) to Manage Database Files".
- Use Oracle ASM HIGH REDUNDANCY for optimal corruption repair. Using Oracle ASM redundancy for disk groups provides mirrored extents that can be used by the database if an I/O error or corruption is encountered. For continued protection, Oracle ASM redundancy provides the ability to move an extent to a different area on a disk if an I/O error occurs. The Oracle ASM redundancy mechanism is useful if you have bad sectors returning media errors. For more information, see Section , "Use Redundancy to Protect from Disk Failure". Note that Automatic ASM corruption repair is not available when using ASM external redundancy.
- Use the Oracle Active Data Guard option for automatic block repair. For more information about Active Data Guard, see Section , "Use Oracle Active Data Guard Best Practices".
- Configure and use Configure Data Recovery Advisor to automatically diagnose data failures. For more information, see Section , "Use Data Recovery Adviser to Detect, Analyze and Repair Data Failures".
- Enable Flashback Technologies for fast point-in-time recovery from logical corruptions most often caused by human error and for fast reinstatement of a primary database following failover. For more information, see Section , "Enable Flashback Database".

- Implement a backup and recovery strategy with Recovery Manager (RMAN) and periodically use the RMAN `BACKUP VALIDATE CHECK LOGICAL...` scan to detect corruptions. For more information, see Chapter 7, "Configuring Backup and Recovery." Use RMAN and Oracle Secure Backup for additional block checks during backup and restore operations. Use Zero Data Loss Recovery Appliance for backup and recovery validation including corruption checks and repairs, central backup validation, reduced production database impact, and Enterprise Cloud backup and recovery solution.
- The Exadata Database Machine and Sparc Supercluster also implements comprehensive Oracle Hardware Assisted Resilient Data (HARD) specifications, providing a unique level of validation for Oracle block data structures. Oracle Exadata Storage Server Software detects corruptions introduced into the I/O path between the database and storage. It stops corrupted data from being written to disk when a HARD check fails. Examples of the Exadata HARD checks include: 1) redo and block checksum, 2) correct log sequence, 3) block type validation, 4) block number validation, 5) Oracle data structures such as block magic number, block size, sequence#, and block header and tail data structures.
- Oracle Exadata Storage Server Software provides Automatic Hard Disk Scrub and Repair. This feature automatically inspects and repairs hard disks periodically when hard disks are idle. If bad sectors are detected on a hard disk, then Oracle Exadata Storage Server Software automatically sends a request to Oracle ASM to repair the bad sectors by reading the data from another mirror copy. By default, the hard disk scrub runs every two weeks. It's very lightweight and has enormous value add by fixing physical block corruptions even with infrequently access data.
- Oracle Data Integrity eXtensions (DIX) with T10 Data Integrity Field (DIF): Oracle Linux team has collaborated with hardware vendors and Oracle database development to extend Oracle data integrity extensions from Oracle's operating system (Linux) to various vendor's host adapter down to the storage device. With these extensions, DIX provides end to end data integrity for reads and writes through a checksum validation. The prerequisite is to leverage certified storage, HBA and disk firmware. An example of this partnership is DIX integration with Oracle Linux, Emulex or QLogic Host Bus Adapters and any T10 DIF capable storage arrays such as EMC VMAX.

  See Also:

  - "An Integrated End-to-End Data Integrity Solution for Oracle Products," an Oracle presentation at https://oss.oracle.com/~mkp/docs/OOW2011-DI.pdf
  - "Preventing Silent Data Corruption Using Emulex Host Bus Adapters, EMC VMAX and Oracle Linux," an EMC, Emulex and Oracle White Paper at http://www.oracle.com/us/technologies/linux/prevent-silent-data-corruption-1852761.pdf

Detecting and Monitoring Data Corruption

If corrupt data is written to disk or if a component failure causes good data to become corrupt after it is written, then it is critical to detect the corrupted blocks as soon as possible.

To monitor the database for errors and alerts:

- Use Enterprise Manager to monitor the availability of all discovered targets and detect errors and alerts. You can also review all targets in a single view from the HA Console.
- Query the V$DATABASE_BLOCK_CORRUPTION view that is automatically updated when block corruption is detected or repaired.
- Configure Data Recovery Advisor to automatically diagnose data failures, determine and present appropriate repair options, and perform repair operations at the user's request. See Section , "Use Data Recovery Adviser to Detect, Analyze and Repair Data Failures" for more information.

  Note: Data Recovery Advisor integrates with the Oracle Enterprise Manager Support Workbench (Support Workbench), the Health Monitor, and RMAN.

- Use Data Guard to detect physical corruptions and to detect lost writes.

  Data Guard can detect physical corruptions when the apply process stops due to a corrupted block in the redo steam or when it detects a lost write. Use Enterprise Manager to manage and monitor your Data Guard configuration. By taking advantage of Automatic Block Media Recovery, a corrupt block found on either a primary database or a physical standby database can be fixed automatically when the Active Data Guard option is used. For more information about Automatic Block Media Recovery, see Section , "Use Active Data Guard".
- Use SQL*Plus to detect data file corruptions and interblock corruptions

  Issue the ANALYZE TABLE tablename VALIDATE STRUCTURE CASCADE SQL*Plus statement. After determining the corruptions, the table can be re-created or another action can be taken.
- Backup and recovery strategy with Recovery Manager (RMAN) can detect physical block corruptions.

  A more intensive RMAN check using RMAN BACKUP VALIDATE CHECK LOGICAL can detect logical block corruptions.

  See Also:

  - Section , "Recovering from Data Corruption"
  - *Oracle Data Guard Concepts and Administration* for more information about Oracle Active Data Guard option and the Automatic Block Repair feature
  - *Oracle Database Backup and Recovery User's Guide* for information about Performing Block Media Recovery
  - Chapter 11, "Monitoring for High Availability" for more information about Enterprise Manager

Set DISK_ASYNCH_IO Initialization Parameter

Under most circumstances, Oracle Database automatically detects if asynchronous I/O is available and appropriate for a particular platform and enables asynchronous I/O through the DISK_ASYNCH_IO initialization parameter. However, for optimal performance, it is always a best practice to ensure that asynchronous I/O is actually being used. Query the V$IOSTAT_FILE view to determine whether asynchronous I/O is used:

```
SQL> select file_no,filetype_name,asynch_io from v$iostat_file;
```

To explicitly enable asynchronous I/O, set the DISK_ASYNCH_IO initialization parameter to TRUE:

```
ALTER SYSTEM SET DISK_ASYNCH_IO=TRUE SCOPE=SPFILE SID='*';
```

Note that if you are using Oracle ASM, it performs I/O asynchronously by default.

> **See Also:** *Oracle Database Reference* for more information about the DISK_ASYNCH_IO initialization parameter

Set LOG_BUFFER Initialization Parameter to 8 MB or Higher

Set the LOG_BUFFER initialization parameter to minimum of 8 MB.

Set LOG_BUFFER to a minimum of 64 MB for databases with flashback enabled and 4GB or higher SGAs.

Set LOG_BUFFER to a minimum of 256 MB if you are using Oracle Data Guard with asynchronous redo transport and have a high redo generation rate. Doing so will allow the asynchronous redo transport to read redo from the log buffer and avoid disk I/Os to online redo logs. Refer to Chapter 8, " Configuring Oracle Data Guard," for details.

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* for information about Configuring and Using the Redo Log Buffer
> - *Oracle Database Reference* for more information about LOG_BUFFER and valid values on 32-bit and 64-bit operating systems
> - For more information about using a buffer hit rate histogram for determining optimal size for log buffer to support redo transport, see "View X$LOGBUF_READHIST and In-Memory Log Buffer Hit Rate Histogram" in My Oracle Support Note 951152.1 at
>
>   https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=951152.1

Use Automatic Shared Memory Management and Avoid Memory Paging

For any systems with 4 GB or more memory, disable Automatic Memory Management by setting MEMORY_TARGET=0 and enable Automatic Shared Memory Management by setting SGA_TARGET.

The sum of SGA and PGA memory allocations on the database server should always be less than your system's physical memory, and conservatively should be less than 75% of total system memory. However, PGA_AGGREGATE_TARGET is not a hard limit, and for some Data Warehouse or reporting applications, the PGA memory can grow to be 3 X PGA_AGGREGATE_TARGET.

Monitor PGA memory and host-based memory utilization using Oracle Enterprise Manager, or by querying v$pgastat and operating systems statistics, to get an accurate understanding of memory utilization.

Avoid memory paging by adjusting the number of databases and applications, or reducing the allocated memory settings. Set PGA_AGGREGATE_LIMIT to specify a hard limit on PGA memory usage. If the PGA_AGGREGATE_LIMIT value is exceeded, Oracle Database first aborts the calls of sessions that are consuming the

most untunable PGA memory. Then, if the total PGA memory usage is still over the limit, the sessions that are using the most untunable memory will be terminated.

Notes:

- Memory consumption for parallel operations is tracked as a unit.
- SYS and fatal background processes other than job queue processes are not subject to being aborted or killed.

On Linux Operating systems it is recommended that you configure HugePages so that ASM and database instances can use it for their SGA. HugePages is a feature integrated into the Linux kernel from release 2.6. This feature provides the alternative to the 4K page size providing bigger pages. Using HugePages has the benefit of saving memory resources by decreasing page table overhead while making sure the memory is not paged to disk. This contributes to faster overall memory performance. Next to this overall node stability will benefit from using HugePages.

Ensuring the entire SGA of a database instance is stored in HugePages can be accomplished by setting the `init.ora` parameter `use_large_pages=only`. Setting this parameter will ensure that an instance will start only when it can get all of its memory for SGA from HugePages. For this reason the setting `use_large_pages=only` is recommended for database instances.

For ASM instances leave `use_large_pages=true` (the default value). This setting still ensures that HugePages are used when available, but also ensures that ASM as part of Grid Infrastructure starts when HugePages are not or insufficiently configured.

Use Automatic Shared Memory Management, as HugePages are not compatible with Automatic Memory Management.

See Also:

- *Oracle Database Administrator's Guide* for more information
- For information about configuring HugePages, see "Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration" and "HugePages on Linux: What It Is... and What It Is Not..." in My Oracle Support Notes 401749.1 and 361323.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=401749.1

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=361323.1

Disable Parallel Recovery for Instance Recovery

When the value of `RECOVERY_ESTIMATED_IOS` in the `V$INSTANCE_RECOVERY` view is small (for example, < 5000), then the overhead of parallel recovery may outweigh any benefit. This typically occurs with a very aggressive setting of `FAST_START_MTTR_TARGET`. In this case, set `RECOVERY_PARALLELISM` to 1 to disable parallel recovery.

See Also:

- Section , "Set FAST START MTTR TARGET Initialization Parameter"
- *Oracle Database Reference* for more information about the `RECOVERY_PARALLELISM` parameter

Recommendations to Improve Manageability

To improve Oracle Database manageability:

- Use Oracle Clusterware with Oracle RAC or Oracle Restart
- Use Data Recovery Adviser to Detect, Analyze and Repair Data Failures
- Use Automatic Performance Tuning Features
- Use a Server Parameter File
- Use Automatic Undo Management
- Use Locally Managed Tablespaces
- Use Automatic Segment Space Management
- Use Temporary Tablespaces and Specify a Default Temporary Tablespace
- Use Resumable Space Allocation
- Use Database Resource Manager
- Use Oracle Multitenant Best Practices

Use Oracle Clusterware with Oracle RAC or Oracle Restart

Configure Oracle Clusterware with Oracle Real Application Clusters (Oracle RAC) or Oracle Restart to automatically restart key application and Oracle services such as the Oracle ASM instance, listeners, application agents, and processes.

Oracle Restart enhances the availability of a single-instance (nonclustered) Oracle database and its components. Oracle Restart is used in single-instance environments only. For Oracle Real Application Clusters (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware.

If you configure Oracle Restart, it automatically restarts the database, the listener, and other Oracle components after a hardware or software failure or whenever the database's host computer restarts. It also ensures that the Oracle components are restarted in the proper order, in accordance with component dependencies.

Oracle Restart runs out of the Oracle Grid Infrastructure home, which you install separately from Oracle Database homes.

See Also:

- Chapter 5, "Configuring Oracle Database with Oracle Clusterware"
- *Oracle Database Administrator's Guide* for information about configuring Oracle Restart

Use Data Recovery Adviser to Detect, Analyze and Repair Data Failures

Use Data Recovery Advisor to quickly diagnose data failures, determine and present appropriate repair options, and execute repairs at the user's request. In this context, a data failure is a corruption or loss of persistent data on disk. By providing a centralized tool for automated data repair, Data Recovery Advisor improves the manageability and reliability of an Oracle database and thus helps reduce the Mean Time To Recover (MTTR). Data Recovery Advisor can diagnose failures based on symptoms, such as:

- Components that are not accessible because they do not exist, do not have the correct access permissions, are taken offline, and so on
- Physical corruptions such as block checksum failures, invalid block header field values, and so on
- Logical corruptions caused by software bugs
- Incompatibility failures caused by an incorrect version of a component
- I/O failures such as a limit on the number of open files exceeded, channels inaccessible, network or I/O errors, and so on
- Configuration errors such as an incorrect initialization parameter value that prevents the opening of the database

If failures are diagnosed, then they are recorded in the Automatic Diagnostic Repository (ADR). Data Recovery Advisor intelligently determines recovery strategies by:

- Generating repair advice and repairing failures only after failures have been detected by the database and stored in ADR
- Aggregating failures for efficient recovery
- Presenting only feasible recovery options
- Indicating any data loss for each option

Typically, Data Recovery Advisor presents both automated and manual repair options. If appropriate, you can choose to have Data Recovery Advisor automatically perform a repair, verify the repair success, and close the relevant repaired failures.

Note: In the current release, Data Recovery Advisor only supports single-instance databases. Oracle RAC databases are not supported. See *Oracle Database Backup and Recovery User's Guide* for more information about Data Recovery Advisor supported database configurations.

See Also:

- Section , "Recovering from Data Corruption" for more information about using Data Recovery Advisor
- *Oracle Database Backup and Recovery User's Guide* for information about diagnosing and repairing failures with Data Recovery Advisor

Use Automatic Performance Tuning Features

Effective data collection and analysis is essential for identifying and correcting performance problems. Oracle provides several tools that gather information regarding database performance.

The Oracle Database automatic performance tuning features include:

- Automatic Workload Repository (AWR)
- Automatic Database Diagnostic Monitor (ADDM)
- SQL Tuning Advisor
- SQL Access Advisor

- Active Session History Reports (ASH)

When using Automatic Workload Repository (AWR), consider the following best practices:

- Create a baseline of performance data to be used for comparison purposes should problems arise. This baseline should be representative of the peak load on the system.
- Set the AWR automatic snapshot interval to 10-20 minutes to capture performance peaks during stress testing or to diagnose performance issues.
- Under usual workloads a 60-minute interval is sufficient.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about Managing the Automatic Workload Repository

Use a Server Parameter File

The server parameter file (SPFILE) enables a single, central parameter file to hold all database initialization parameters associated with all instances of a database. This provides a simple, persistent, and robust environment for managing database parameters. An SPFILE is required when using Oracle Data Guard broker.

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for information about managing initialization parameters with an SPFILE
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information about initialization parameters with Real Application Clusters
> - *Oracle Data Guard Broker* for information about other prerequisites for using the broker

Use Automatic Undo Management

With automatic undo management, the Oracle Database server effectively and efficiently manages undo space, leading to lower administrative complexity and cost. When Oracle Database internally manages undo segments, undo block and consistent read contention are eliminated because the size and number of undo segments are automatically adjusted to meet the current workload requirement.

To use automatic undo management, set the following initialization parameters:

- `UNDO_MANAGEMENT`

  Set this parameter to `AUTO`.

- `UNDO_RETENTION`

  Specify the desired time in seconds to retain undo data. Set this parameter to the same value on all instances.

- `UNDO_TABLESPACE`

  Specify a unique undo tablespace for each instance.

Advanced object recovery features, such as Flashback Query, Flashback Version Query, Flashback Transaction Query, and Flashback Table, require automatic undo management. The success of these features depends on the availability of undo information to view data as of a previous point in time.

By default, Oracle Database automatically tunes undo retention by collecting database usage statistics and estimating undo capacity needs. Unless you enable retention guarantee for the undo tablespace (by specifying the RETENTION GUARANTEE clause on either the CREATE DATABASE or the CREATE UNDO TABLESPACE statement), Oracle Database may reduce the undo retention below the specified UNDO_RETENTION value.

> **Note:** By default, ongoing transactions can overwrite undo data even if the UNDO_RETENTION parameter setting specifies that the undo data should be maintained. To guarantee that unexpired undo data is not overwritten, you must enable RETENTION GUARANTEE for the undo tablespace.

If there is a requirement to use Flashback technology features, the best practice recommendations is to enable RETENTION GUARANTEE for the undo tablespace and set a value for UNDO_RETENTION based on the following guidelines:

1. Establish how long it would take to detect when erroneous transactions have been carried out. Multiply this value by two.
2. Use the Undo Advisor to compute the minimum undo tablespace size based on setting UNDO_RETENTION to the value recommended in step 1.
3. If the undo tablespace has the AUTOEXTEND option disabled, allocate enough space as determined in step 2 or reduce the value of the UNDO_RETENTION parameter.
4. If the undo tablespace has the AUTOEXTEND option enabled, make sure there is sufficient disk space available to extend the data files to the size determined in step 2. Make sure the autoextend MAXSIZE value you specified is large enough.

With the RETENTION GUARANTEE option, if the tablespace is configured with less space than the transaction throughput requires, then the following sequence of events occurs:

1. If you have an autoextensible file, then the file automatically grows to accommodate the retained undo data.
2. A warning alert reports the disk is at 85% full.
3. A critical alert reports the disk is at 97% full.
4. Transactions receive an out-of-space error.

See Also:

- *Oracle Database 2 Day DBA* for information about computing the minimum undo tablespace size using the Undo Advisor
- *Oracle Database Administrator's Guide* for more information about the UNDO_RETENTION setting and the size of the undo tablespace

Use Locally Managed Tablespaces

Locally managed tablespaces perform better than dictionary-managed tablespaces, are easier to manage, and eliminate space fragmentation concerns. Locally managed tablespaces use bitmaps stored in the data file headers and, unlike dictionary managed tablespaces, do not contend for centrally managed resources for space allocations and de-allocations.

See Also: *Oracle Database Administrator's Guide* for more information about locally managed tablespaces

Use Automatic Segment Space Management

Automatic segment space management simplifies space administration tasks, thus reducing the chance of human error. An added benefit is the elimination of performance tuning related to space management. It facilitates management of free space within objects such as tables or indexes, improves space utilization, and provides significantly better performance and scalability with simplified administration. The automatic segment space management feature is enabled by default for all tablespaces created using default attributes.

> **See Also:** *Oracle Database Administrator's Guide* for more information about automatic segment space management

Use Temporary Tablespaces and Specify a Default Temporary Tablespace

Temporary tablespaces improve the concurrency of multiple sort operations, reduce sort operation overhead, and avoid data dictionary space management operations. This is a more efficient way of handling temporary segments, from the perspective of both system resource usage and database performance.

The best practice is to specify a default temporary tablespace for the entire database to ensure that temporary segments are used for the most efficient sort operations, whether individual users have been assigned a temporary tablespace.

To Specify a Default Temporary Tablespace ...	Then ...
When creating the database ...	Use the `DEFAULT TEMPORARY TABLESPACE` clause of the `CREATE DATABASE` statement
After database creation ...	Use the `ALTER DATABASE` statement

Using the default temporary tablespace ensures that all disk sorting occurs in a temporary tablespace and that other tablespaces are not mistakenly used for sorting.

> **See Also:** *Oracle Database Administrator's Guide* for more information about managing tablespaces

Use Resumable Space Allocation

Resumable space allocation provides a way to suspend and later resume database operations if there are space allocation failures. The affected operation is suspended instead of the database returning an error. No processes must be restarted. When the space problem is resolved, the suspended operation is automatically resumed.

To use resumable space allocation, you can set it at the system level with the `RESUMABLE_TIMEOUT` initialization parameter, or enable it at the session level using clauses of the `ALTER SESSION` statement (for example, issue the `ALTER SESSION ENABLE RESUMABLE` statement). The default for a new session is resumable mode disabled, unless you explicitly set the `RESUMABLE_TIMEOUT` initialization parameter to a nonzero value.

> **See Also:** *Oracle Database Administrator's Guide* for more information about managing resumable space allocation

Use Database Resource Manager

Oracle Database Resource Manager (the Resource Manager) gives database administrators more control over resource management decisions, so that resource

allocation can be aligned with the business objectives of an enterprise. The Resource Manager provides the ability to prioritize work within the Oracle Database server. Availability of the database encompasses both its functionality and performance. If the database is available but users are not getting the level of performance they need, then availability and service level objectives are not being met. Application performance, to a large extent, is affected by how resources are distributed among the applications that access the database. The main goal of the Resource Manager is to give the Oracle Database server more control over resource management decisions, thus circumventing problems resulting from inefficient operating system management and operating system resource managers.

When you use the Resource Manager:

- Use Enterprise Manager to manage resource plans.
- When you test with the Resource Manager, ensure there is sufficient load on the system to make CPU resources scarce.

See Also:

- *Oracle Database Administrator's Guide* for more information about Oracle Database Resource Manager
- For information about configuring and troubleshooting Database Resource Manager, see "Resource Manager Training (11.2 features included)" in My Oracle Support Note 1119407.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1119407.1

Use Oracle Multitenant Best Practices

All of the best practices identified in this section for non-container databases (non-CDBs) also apply to Oracle Multitenant container databases (CDBs). The following best practices also apply:

- Create CDBs with the AL32UTF8 character set. This will allow any pluggable database (PDB) to be plugged into this CDB regardless of its character set.
- If creating a CDB to be used as a destination for a PDB upgrade, ensure that the `COMPATIBLE` parameter is set to the same value for both the source CDB and the destination PDB. This allows for easy unplug/plug fallback to the previous release. When all of the PDBs targeted for the destination CDB have been plugged in and are fully tested, raise the `COMPATIBLE` setting.
- The Database Creation Assistant (DBCA) enforces enabling of all database options when a CDB is created to allow any PDB regardless of option usage to be plugged into this new CDB. Do not disable the database options from the CDB.
- We recommend enabling Flashback Database; however, note that at this time flashback is performed at the CDB level so that all PDBs in the container will be flashed back to the same point in time.

5

Configuring Oracle Database with Oracle Clusterware

Oracle Clusterware enables servers to communicate with each other, so that they appear to function as a collective unit. Oracle Clusterware provides the infrastructure necessary to run Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node. The Grid Infrastructure is the software that provides the infrastructure for an enterprise grid architecture. In a cluster, this software includes Oracle Clusterware and Oracle ASM. For a standalone server, the Grid Infrastructure includes Oracle Restart and Oracle ASM. Oracle Database 12*c* combines these infrastructure products into one software installation called the Grid Infrastructure home. Oracle Restart provides managed startup and restart of a single-instance (non-clustered) Oracle Database, Oracle ASM instance, service, listener, and any other process running on the server. If an interruption of a service occurs after a hardware or software failure, Oracle Restart automatically takes the necessary steps to restart the component.

This chapter contains the following topics:

About Oracle Clusterware Best Practices

Oracle Clusterware, as part of Oracle Grid Infrastructure, is software that manages the availability of user applications and Oracle databases. Oracle Clusterware is the only clusterware needed for most platforms on which Oracle RAC operates. You can also use clusterware from other vendors in addition to Oracle Clusterware on the same system, if needed. However, adding unnecessary layers of software for functionality that is provided by Oracle Clusterware adds complexity and cost and can reduce system availability, especially for planned maintenance.

> **Note:** Before installing Oracle RAC or Oracle RAC One Node, you must first install Oracle Grid Infrastructure, which consists of Oracle Clusterware and Oracle Automatic Storage Management (ASM). After Oracle Clusterware and Oracle ASM are operational, you can use Oracle Universal Installer to install the Oracle Database software with the Oracle RAC components.

Oracle Clusterware includes a high availability framework that provides an infrastructure to manage any application. Oracle Clusterware ensures that the

applications it manages start when the system starts and monitors the applications to ensure that they are always available. If a process fails then Oracle Clusterware attempts to restart the process using agent programs (agents). Oracle clusterware provides built-in agents so that you can use shell or batch scripts to protect and manage an application. Oracle Clusterware also provides preconfigured agents for some applications (for example for Oracle TimesTen In-Memory Database). If a node in the cluster fails, then you can program processes that normally run on the failed node to restart on another node. The monitoring frequency, starting, and stopping of the applications and the application dependencies are configurable.

Oracle RAC One Node is a single instance of an Oracle RAC database that runs on one node in a cluster. For information about working with Oracle RAC One Node, see Section , "Configuring Oracle Database with Oracle RAC One Node".

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about Making Applications Highly Available Using Oracle Clusterware
- *Oracle Database 2 Day + Real Application Clusters Guide* for more information about installing Oracle Grid Infrastructure for a cluster

Client Configuration and Migration Concepts

Oracle provides the following features to enable you to migrate client connections between nodes. These features minimize the impact of a node failure from the client perspective:

- Client-side load balancing and connection load balancing. For more information see Section , "Use Client-Side and Server-Side Load Balancing."
- Single Client Access Name (SCAN)
- Fast Application Notification (FAN)
- Fast Connection Failover (FCF) (ideally used by FAN-enabled clients)
- Application Continuity and Transaction Guard

Tip: For more information about using and configuring Fast Connection Failover (FCF), see Chapter 10, "Client Failover Best Practices for Highly Available Oracle Databases."

Services

Services de-couple any hardwired mapping between a connection request and an Oracle RAC instance. Services are an entity defined for an Oracle RAC database that allows the workload for an Oracle RAC database to be managed. Services divide the entire workload executing in the Oracle Database into mutually disjoint classes. Each service represents a workload with common attributes, service level thresholds, and priorities. The grouping is based on attributes of the work that might include the application being invoked, the application function to be invoked, the priority of execution for the application function, the job class to be managed, or the data range used in the application function or job class.

To manage workloads or a group of applications, you can define services that you assign to a particular application or to a subset of an application's operations. You can also group work by type under services. For example, online users can use one service,

while batch processing can use another and reporting can use yet another service to connect to the database.

Oracle recommends that all users who share a service have the same service level requirements. You can define specific characteristics for services and each service can represent a separate unit of work. There are many options that you can take advantage of when using services. Although you do not have to implement these options, using them helps optimize application performance.

Fast Application Notification (FAN)

Fast Application Notification (FAN) is a notification mechanism that is integrated into Oracle Clusterware to notify registered clients about configuration and service level information that includes service status changes, such as UP or DOWN events. Applications can respond to FAN events and take immediate action. FAN UP and DOWN events can apply to instances, services, and nodes.

The use of Fast Application Notification (FAN) requires the use of Services.

For cluster configuration changes, the Oracle RAC high availability framework publishes a FAN event immediately when a state change occurs in the cluster. Instead of waiting for the application to poll the database and detect a problem, applications can receive FAN events and react immediately. With FAN, in-flight transactions can be immediately terminated and the client notified when the instance fails.

FAN also publishes load balancing advisory events. Applications can take advantage of the load balancing advisory FAN events to direct work requests to the instance in the cluster that is currently providing the best service quality.

> **See Also:** *Oracle Database Administrator's Guide* for more information about Overview of Fast Application Notification

Single Client Access Name (SCAN)

Single Client Access Name (SCAN) was introduced with Oracle Real Application Clusters (Oracle RAC) 11*g* Release 2 and provides a single name for clients to access an Oracle Database running in a cluster. The benefit is clients using SCAN do not need to change their connect string if you add or remove nodes in the cluster. Having a single name to access the cluster allows clients to use the EZConnect client and the simple JDBC thin URL to access any database running in the clusters independently of which server(s) in the cluster the database is active. SCAN provides load balancing and failover of client connections to the database. The SCAN is a virtual IP name, similar to the names used for virtual IP addresses, such as `node1-vip`. However, unlike a virtual IP, the SCAN is associated with the entire cluster, rather than an individual node, and associated with multiple IP addresses, not just one address. In Oracle Real Application Clusters 12*c* SCAN supports multiple subnets in the cluster (one SCAN per subnet).

> **See Also:**
>
> - *Oracle Clusterware Administration and Deployment Guide* for more information about Single Client Access Name (SCAN)
> - *Oracle Database Net Services Administrator's Guide* for more information about EZConnect

Application Continuity and Transaction Guard

Application Continuity (AC) is a new technology that protects applications from instance and session failures by re-playing affected "in-flight" transactions on another database instance in the cluster.

It enables the replay, in a non-disruptive and rapid manner, of a request against the database after a recoverable error that makes the database session unavailable. The request can contain transactional and non-transactional calls to the database. After a successful replay, the application can continue where the database session left off.

Users are not left in doubt whether their transactions have gone through successfully

Transaction Guard is a reliable protocol and tool that returns the outcome of the last in-flight transaction after an outage that makes the database session unavailable. Without Transaction Guard, applications and users who attempt to retry operations following an outage can cause logical corruption by committing duplicate transactions or committing transactions out of order.

Use UCP version 12.1.0.2 or later for Application Continuity and Transaction Guard.

> **See also:** *Oracle Database Development Guide* for information about Application Continuity and Transaction Guard

Oracle Clusterware Configuration Best Practices

Use the following Oracle Clusterware configuration best practices:

- Use the Cluster Verification Utility (CVU)
- Use a Local Home for Oracle Database and Oracle Clusterware with Oracle ASM
- Ensure Services are Highly Available
- Client Configuration and FAN Best Practices
- Connect to Database Using Services and Single Client Access Name (SCAN)
- Use Client-Side and Server-Side Load Balancing
- Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks with Oracle ASM
- Use Company Wide Cluster Time Management
- Verify That Oracle Clusterware, Oracle RAC, and Oracle ASM Use the Same Interconnect Network
- Use Redundant Interconnect with Highly Available IP (HAIP)
- Configure Failure Isolation with Intelligent Management Platform Interface (IPMI)
- Use Jumbo Frames for Cluster Interconnect Network

Use the Cluster Verification Utility (CVU)

Cluster Verification Utility (CVU) can verify the primary cluster components during installation and you can use the utility to verify configuration and components. A component can be basic, such as free disk space, or it can be complex, such as checking Oracle Clusterware integrity. For example, CVU can verify multiple Oracle Clusterware subcomponents across Oracle Clusterware layers. Additionally, CVU can check disk space, memory, processes, and other important cluster components.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for information about the Cluster Verification Utility

Use a Local Home for Oracle Database and Oracle Clusterware with Oracle ASM

All rolling patch features require that the software home being patched is local, not shared. The software must be physically present in a local file system on each node in the cluster and it is not on a shared cluster file system.

The reason for this requirement is that if a shared cluster file system is used, patching the software on one node affects all of the nodes, and would require that you shut down all components using the software on all nodes. Using a local file system allows software to be patched on one node without affecting the software on any other nodes.

Note the following when you install Oracle Grid Infrastructure and configure Oracle Clusterware:

- Oracle RAC databases require shared storage for the database files.
- Configure Oracle Cluster Registry (OCR) and voting files to use Oracle ASM. For more information, see Section , "Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks with Oracle ASM"
- Oracle recommends that you install Oracle Database on local homes, rather than using a shared home on shared storage. It is recommended to *not* use a shared file system for the Oracle Database Home (using a shared home prevents you from doing rolling upgrades, as all software running from that shared location must be stopped before it can be patched).
- Oracle Clusterware and Oracle ASM are both installed in one home on a non shared file system called the Grid Infrastructure home (`Grid_home`).

> **See Also:** *Oracle Database 2 Day + Real Application Clusters Guide* for more information about installing Oracle ASM

Ensure Services are Highly Available

For cases where a service only has one preferred instance, ensure that the service is started immediately on an available instance after it is brought down on its preferred instance. Starting the service immediately ensures that affected clients can instantaneously reconnect and continue working. Oracle Clusterware handles this responsibility and it is of utmost importance during unplanned outages.

Even though you can rely on Oracle Clusterware to start the service during planned maintenance as well, it is safer to ensure that the service is available on an alternate instance by manually starting an alternate preferred instance ahead of time. Manually starting an alternate instance eliminates the single point of failure with a single preferred instance and you have the luxury to do this because it is a planned activity. Add at least a second preferred instance to the service definition and start the service before the planned maintenance. You can then stop the service on the instance where maintenance is being performed with the assurance that another service member is available. Adding one or more preferred instances does not have to be a permanent change. You can revert it back to the original service definition after performing the planned maintenance.

Manually relocating a service rather than changing the service profile is advantageous in cases such as the following:

- If you are using Oracle XA, then use of manual service relocation is advantageous because, although the XA specification allows for a transaction branch to be suspended and resumed by the TM, if connection load balancing is utilized then any resumed connection could land on an alternate Oracle instance to the one that the transaction branch started on and there is a performance implication if a single distributed transaction spans multiple database instances.

- If an application is not designed to work properly with multiple service members, then application errors or performance issues can arise.

As with all configuration changes, you should test the effect of a service with multiple members to assess its viability and impact in a test environment before implementing the change in your production environment.

> **See Also:** The Technical Article, "XA and Oracle controlled Distributed Transactions" on the Oracle Real Application Clusters website at
> http://www.oracle.com/technetwork/database/clustering/overview/index.html

Client Configuration and FAN Best Practices

The ability to migrate client connections to and from the nodes on which you are working is a critical aspect of planned maintenance. Migrating client connections should always be the first step in any planned maintenance activity requiring software shutdown (for example, when performing a rolling upgrade). The potential for problems increases if there are still active database connections when the service switchover commences.

An example of a best-practice process for client redirection during planned maintenance could involve the following steps:

1. Client is configured to receive FAN notifications and is properly configured for run time connection load balancing and Fast Connection Failover.
2. Oracle Clusterware stops services on the instance to be brought down or relocates services to an alternate instance.
3. Oracle Clusterware returns a `Service-Member-Down` event.
4. Client that is configured to receive FAN notifications receives a notification for a `Service-Member-Down` event and moves connections to other instances offering the service.

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for an Introduction to Automatic Workload Management.
> - Detailed information about client failover best practices in an Oracle RAC environment are available in the "Automatic Workload Management with Oracle Real Application Clusters 11*g*" Technical Article on the Oracle Technology Network at
>
>   http://www.oracle.com/technetwork/database/clustering/overview/index.html

Connect to Database Using Services and Single Client Access Name (SCAN)

With Oracle Database 12*c*, application workloads can be defined as services so that they can be automatically or manually managed and controlled. For manually managed services, DBAs control which processing resources are allocated to each service during both normal operations and in response to failures. Performance metrics are tracked by service and thresholds set to automatically generate alerts should these thresholds be crossed. CPU resource allocations and resource consumption controls are managed for services using Database Resource Manager.

Oracle tools and facilities such as Job Scheduler, Parallel Query, and Oracle Streams Advanced Queuing also use services to manage their workloads.

With Oracle Database 12*c*, you can define rules to automatically allocate processing resources to services. Oracle RAC in Oracle Database 12*c* instances can be allocated to process individual services or multiple services, as needed. These allocation rules can be modified dynamically to meet changing business needs. For example, you could modify these rules at quarter end to ensure that there are enough processing resources to complete critical financial functions on time. You can also define rules so that when instances running critical services fail, the workload is automatically shifted to instances running less critical workloads. You can create and administer services with the SRVCTL utility or with Oracle Enterprise Manager.

You should make application connections to the database by using a VIP address (preferably SCAN) in combination with a service to achieve the greatest degree of availability and manageability.

A VIP address is an alternate public address that client connections use instead of the standard public IP address. If a node fails, then the node's VIP address fails over to another node but there is no listener listening on that VIP, so a client that attempts to connect to the VIP address receives a connection refused error (`ORA-12541`) instead of waiting for long TCP connect timeout messages. This error causes the client to quickly move on to the next address in the address list and establish a valid database connection. New client connections can initially try to connect to a failed-over-VIP, but when there is no listener running on that VIP the "no listener" error message is returned to the clients. The clients traverse to the next address in the address list that has a non-failed-over VIP with a listener running on it.

The Single Client Access Name (SCAN) is a fully qualified name (hostname+domain) that is configured to resolve to all three of the VIP addresses allocated for the SCAN. The addresses resolve using Round Robin DNS either on the DNS server, or within the cluster in a GNS configuration. SCAN listeners can run on any node in the cluster, multiple SCAN listeners can run on one node.

In Oracle Database 12*c* and later, by default, instances register with SCAN listeners as remote listeners.

SCANs are cluster resources. SCAN vips and SCAN listeners run on random cluster nodes. SCANs provide location independence for the databases, so that client configuration does not have to depend on which nodes are running a particular database. For example, if you configure policy managed server pools in a cluster, then the SCAN enables connections to databases in these server pools regardless of which nodes are allocated to the server pool.

SCAN names functions like a virtual cluster address. SCANs are resolved to three SCAN VIPs that may run on any node in the cluster. So unlike a VIP address per node as entry point, clients connecting to the SCAN no longer require any updates on how they connect when a virtual IP addresses is added, changed, or removed from the cluster. The SCAN addresses resolve to the cluster, rather than to a specific node address.

During Oracle Grid Infrastructure installation, SCAN listeners are created for as many IP addresses as there are addresses assigned to resolve to the SCAN. Oracle recommends that the SCAN resolves to three addresses, to provide high availability and scalability. If the SCAN resolves to three addresses, then there are three SCAN listeners created.

Oracle RAC provides failover with the node VIP addresses by configuring multiple listeners on multiple nodes to manage client connection requests for the same database service. If a node fails, then the service connecting to the VIP is relocated transparently

to a surviving node. If the client or service are configured with transparent application failover options, then the client is reconnected to a surviving node. When a SCAN Listener receives a connection request, the SCAN Listener checks for the least loaded instance providing the requested service. It then re-directs the connection request to the local listener on the node where the least loaded instance is running. Subsequently, the client is given the address of the local listener. The local listener finally creates the connection to the database instance.

Clients configured to use IP addresses for Oracle Database releases before Oracle Database 11*g* Release 2 can continue to use their existing connection addresses; using SCANs is not required: in this case, the pre-Database 11*g* Release 2 client would use a TNS connect descriptor that resolves to the node-VIPs of the cluster. When an earlier version of Oracle Database is upgraded, it registers with the SCAN listeners, and clients can start using the SCAN to connect to that database. The database registers with the SCAN listener through the remote listener parameter in the init.ora file. The `REMOTE_LISTENER` parameter must be set to SCAN:PORT. Do not set it to a TNSNAMES alias with a single address with the SCAN as HOST=SCAN. Having a single name to access the cluster allows clients to use the EZConnect client and the simple JDBC thin URL to access any database running in the cluster, independently of which server(s) in the cluster the database is active. For example:

```
sqlplus system@sales1-scan:1521/oltp
```

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about automatic workload management
- *Oracle Real Application Clusters Administration and Deployment Guide* for an Overview of Connecting to Oracle Database Using Services and VIP Addresses
- *Oracle Clusterware Administration and Deployment Guide* for more information about Oracle Clusterware Network Configuration Concepts
- *Oracle Database Net Services Administrator's Guide* for more information about EZConnect

Use Client-Side and Server-Side Load Balancing

Client-side load balancing is defined in your client connection definition (tnsnames.ora file, for example) by setting the `LOAD_BALANCE` parameter to `LOAD_BALANCE=ON`. When you set this parameter to `ON`, Oracle Database randomly selects an address in the address list, and connects to that node's listener. This balances client connections across the available SCAN listeners in the cluster.

The SCAN listener redirects the connection request to the local listener of the instance that is least loaded and provides the requested service. When the listener receives the connection request, the listener connects the user to an instance that the listener knows provides the requested service. To see what services a listener supports, run the `lsnrctl` services command.

When clients connect using SCAN, Oracle Net automatically load balances client connection requests across the three IP addresses you defined for the SCAN, unless you are using EZConnect.

Server-side load balancing When you create an Oracle RAC database with DBCA, it automatically:

- Configures and enables server-side load balancing
- Sets the remote listener parameter to the SCAN listener (Note: If you do not use DBCA, you should set the `REMOTE_LISTENER` database parameter to scan_name:scan_port.)
- Creates a sample client-side load balancing connection definition in the tnsnames.ora file on the server

> **Note:** The following features do not work with the default database service. You must create cluster managed services to take advantage of these features. You can only manage the services that you create. Any service created automatically by the database server is managed by the database server.

To further enhance connection load balancing, use the Load Balancing Advisory and define the connection load balancing for each service. Load Balancing Advisory provides information to applications about the current service levels that the database and its instances are providing. The load balancing advisory makes recommendations to applications about where to direct application requests to obtain the best service based on the policy that you have defined for that service. Load balancing advisory events are published through ONS. There are two types of service-level goals for run-time connection load balancing:

`SERVICE_TIME`: Attempts to direct work requests to instances according to response time. Load balancing advisory data is based on elapsed time for work done in the service plus available bandwidth to the service. An example for the use of `SERVICE_TIME` is for workloads such as internet shopping where the rate of demand changes. The following example shows how to set the goal to `SERVICE_TIME` for connections using the online service:

```
srvctl modify service -d db_unique_name -s online -B SERVICE_TIME -j SHORT
```

`THROUGHPUT`: Attempts to direct work requests according to throughput. The load balancing advisory is based on the rate that work is completed in the service plus available bandwidth to the service. An example for the use of `THROUGHPUT` is for workloads such as batch processes, where the next job starts when the last job completes. The following example shows how to set the goal to `THROUGHPUT` for connections using the `sjob` service:

```
srvctl modify service -d db_unique_name -s sjob -B THROUGHPUT -j LONG
```

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Connection Load Balancing and Load Balancing Advisory
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about Configuring Your Environment to Use the Load Balancing Advisory
- *Oracle Database Net Services Administrator's Guide* for more information about configuring listeners
- *Oracle Database Reference* for more information about the `LOCAL_LISTENER` and `REMOTE_LISTENER` parameters

Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks with Oracle ASM

Configure Oracle Cluster Registry (OCR) and voting files to use Oracle ASM. It is recommended to mirror OCR and configure multiple voting disks using an Oracle ASM high redundancy disk group when available.

The Oracle Cluster Registry (OCR) contains important configuration data about cluster resources. Always protect the OCR by using Oracle ASM redundant disk groups for example. Oracle Clusterware uses the Oracle Cluster Registry (OCR) to store and manage information about the components that Oracle Clusterware controls, such as Oracle RAC databases, listeners, virtual IP addresses (VIPs), and services and any applications. To ensure cluster high availability when using a shared cluster file system, as opposed to when using Oracle ASM, Oracle recommends that you define multiple OCR locations.

In addition:

- You can have up to five OCR locations
- Each OCR location must reside on shared storage that is accessible by all of the nodes in the cluster
- You can replace a failed OCR location online if it is not the only OCR location
- You must update OCR through supported utilities such as Oracle Enterprise Manager, the Server Control Utility (SRVCTL), the OCR configuration utility (OCRCONFIG), or the Database Configuration Assistant (DBCA)

Each OCR location must reside on shared storage that is accessible by all of the nodes in the cluster and the voting disk also must reside on shared storage. For high availability, Oracle recommends that you have multiple voting disks on multiple storage devices across different controllers, where possible. Oracle Clusterware enables multiple voting disks, but you must have an odd number of voting disks, such as three, five, and so on. If you define a single voting disk, then you should use external redundant storage to provide redundancy.

Extended Oracle RAC requires a quorum (voting) disk that should be on an arbiter site at a location different from the main sites (data centers). For more information, see Section , "Add a Third Voting Disk to Host the Quorum Disk".

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about Adding, Replacing, Repairing, and Removing Oracle Cluster Registry Locations
- *Oracle Database 2 Day + Real Application Clusters Guide* for more information about managing OCR and voting disks
- *Oracle Clusterware Administration and Deployment Guide* for information about voting disks and oracle cluster registry requirements

Use Company Wide Cluster Time Management

The Cluster Time Synchronization Service (CTSS) is installed as part of Oracle Clusterware. By default the CTSS runs in observer mode if it detects a time synchronization service or a time synchronization service configuration, valid or broken, on the system. If CTSS detects that there is no time synchronization service or time synchronization service configuration on any node in the cluster, then CTSS goes into active mode and takes over time management for the cluster.

The Network Time Protocol (NTP) is a protocol that client and server applications use to maintain correct time settings on client machines. Each database server running Oracle Clusterware is an NPT client and must have NTP client software installed and configured to synchronize its clock to the network time server. The Windows Time service is not an exact implementation of the NTP, but it is based on the NTP specifications.

As a best practice use company-wide NTP.

See Also:

- *Oracle Database 2 Day + Real Application Clusters Guide* for information About Setting the Time on All Nodes
- *Oracle Clusterware Administration and Deployment Guide* for information about Cluster Time Synchronization Service (CTSS) and Cluster Time Management

Verify That Oracle Clusterware, Oracle RAC, and Oracle ASM Use the Same Interconnect Network

For efficient network detection and failover and optimal performance, Oracle Clusterware, Oracle RAC, and Oracle ASM should use the same dedicated interconnect subnet so that they share the same view of connections and accessibility.

Perform the following steps to verify the interconnect settings:

1. To verify the interconnect settings for an Oracle RAC or Oracle ASM instance, do either of the following:
   - Issue the command:

```
SQL> select * from v$cluster_interconnects;

NAME   IP_ADDRESS      IS_PUBLIC    SOURCE
-----  ----------      ---------    -------
bond0  192.168.32.87   NO           cluster_interconnects parameter
```

- Consult the alert log to verify the interconnect settings for the instance.

2. To verify the interconnect subnet used by the clusterware:

```
prompt> $GI_HOME/bin/oifcfg getif | grep cluster_interconnect

bond0 192.168.32.0 global cluster_interconnect
```

Note: Multiple interconnects can be specified for one instance. It is a best practice to use network bonding for redundancy to ensure high availability.

See: *Oracle Database Administrator's Guide* for information about Viewing the Alert Log

Use Redundant Interconnect with Highly Available IP (HAIP)

You can define multiple interfaces for Redundant Interconnect usage by classifying the interfaces as private either during installation or after installation using the `oifcfg setif` command. When you do, Oracle Clusterware creates from one to four (depending on the number of interfaces you define) highly available IP (HAIP) addresses, which Oracle Database and Oracle ASM instances use to ensure highly available and load balanced communications. With HAIP, by default, interconnect traffic is load balanced across all active interconnect interfaces, and corresponding HAIP address are failed over transparently to other adapters if one fails or becomes non-communicative. Oracle Clusterware automatically picks free link local addresses from reserved 169.254.*.* subnet for HAIP.

The Oracle software (including Oracle RAC, Oracle ASM, and Oracle ACFS, all Oracle Database 11*g* release 2 (11.2.0.2), or later), by default, uses these HAIP addresses for all of its traffic allowing for load balancing across the provided set of cluster interconnect interfaces. If a defined cluster interconnect interface fails or becomes non-communicative, then Oracle Clusterware transparently moves the corresponding HAIP address to a remaining functional interface.

Note: Oracle Databases releases before Oracle Database 11*g* release 2 cannot use the Redundant Interconnect Usage feature and must use Operating System based interface bonding technologies instead.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about Redundant Interconnect Usage with HAIP
- *Oracle Grid Infrastructure Installation Guide* for your platform for information about defining interfaces
- For more information, see "Grid Infrastructure Redundant Interconnect and ora.cluster_interconnect.haip" in My Oracle Support Note 1210883.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1210883.1

Configure Failure Isolation with Intelligent Management Platform Interface (IPMI)

Failure isolation is a process by which a failed node is isolated from the rest of the cluster to prevent the failed node from corrupting data. The ideal fencing involves an external mechanism capable of taking a node out of the cluster without cooperation either from Oracle Clusterware or from the operating system running on that node. To provide this capability, Oracle Clusterware 11*g* Release 2 (11.2) and later supports the Intelligent Management Platform Interface specification (IPMI), an industry-standard management protocol.

Typically, you configure failure isolation using IPMI during Grid Infrastructure installation, when you are provided with the option of configuring IPMI from the Failure Isolation Support screen. If you do not configure IPMI during installation, then you can configure it after installation using the Oracle Clusterware Control utility (`CRSCTL`).

> **See:** *Oracle Clusterware Administration and Deployment Guide* for information about IPMI and for information about Configuring IPMI for Failure Isolation

Use Jumbo Frames for Cluster Interconnect Network

Ethernet is a widely used networking technology for Cluster Interconnects. Ethernet's variable frame size of 46-1500 bytes is the transfer unit between the all Ethernet participants, such as the hosts and switches. The upper bound, in this case 1500, is called MTU (Maximum Transmission Unit). When an application sends a message greater than 1500 bytes (MTU), it is fragmented into 1500 byte, or smaller, frames from one end-point to another. In Oracle RAC, the setting of DB_BLOCK_SIZE multiplied by the MULTI_BLOCK_READ_COUNT determines the maximum size of a message for the Global Cache and the PARALLEL_EXECUTION_MESSAGE_SIZE determines the maximum size of a message used in Parallel Query. These message sizes can range from 2K to 64K or more, and hence will get fragmented more so with a lower/default MTU.

Jumbo Frames introduces the ability for an Ethernet frame to exceed its IEEE 802 specified Maximum Transfer Unit of 1500 bytes up to a maximum of 9000 bytes.

note: Jumbo Frames can be implemented for private Cluster Interconnects, and requires careful configuration and testing to realize its benefits also Oracle Engineered Systems may already have an optimal setting pre-configured.

Consider using InfiniBand on the interconnect for workloads that have high volume requirements. InfiniBand can also improve performance by lowering latency. When InfiniBand is in place the RDS protocol can be used to further reduce latency.

Oracle Clusterware Operational Best Practices

Operational best practices for Oracle Clusterware include:

- Capacity Planning
- Regularly Back Up OCR to Tape or Offsite
- Use Cluster Health Monitor for Troubleshooting

Capacity Planning

Proper capacity planning is a critical success factor for all aspects of Oracle clustering technology, but it is of particular importance for planned maintenance. You must

ensure that the work a cluster is responsible for can be done when a small part of the cluster, for example, a node, is unavailable. If the cluster cannot keep up after a planned or unplanned outage, the potential for cascading problems is higher due to system resource starvation.

When sizing your cluster, ensure that *n* percentage of the cluster can meet your service levels where *n* percentage represents the amount of computing resource left over after a typical planned or unplanned outage. For example, if you have a four-node cluster and you want to apply patches in a rolling fashion—meaning one node is upgraded at a time—then three nodes can run the work requested by the application.

One other aspect to capacity planning that is important during planned maintenance is ensuring that any work being done as part of the planned maintenance is separated from the application work when possible. For example, if a patch requires that a SQL script is run after all nodes have been patched, it is a best-practice to run this script on the last node receiving the patch before allowing the application to start using that node. This technique ensures that the SQL script has full use of the operating system resources on the node and it is less likely to affect the application. For example, the `CATCPU.SQL` script that must be run after installing the CPU patch on all nodes.

Regularly Back Up OCR to Tape or Offsite

Oracle Clusterware automatically creates Oracle Cluster Registry (OCR) backups every four hours on one node in the cluster, which is the OCR master node. Oracle always retains the last three backup copies of OCR. The `CRSD` process that creates the backups also creates and retains an OCR backup for each full day and after each week. You should use Oracle Secure Backup, or standard operating-system tools, or third-party tools to back up the backup files created by Oracle Clusterware as part of the operating system backup.

> **Note:** The default location for generating OCR backups on UNIX-based systems is `CRS_HOME/cdata/`*`cluster_name`* where *`cluster_name`* is the name of your cluster. The Windows-based default location for generating backups uses the same path structure. Backups are taken on the OCR master node. To list the node and location of the backup, issue the `ocrconfig -showbackup` command.

In addition to using the automatically created OCR backup files, you can use the `-manualbackup` option on the `ocrconfig` command to perform a manual backup, on demand. For example, you can perform a manual backup before and after you make changes to the OCR such as adding or deleting nodes from your environment, modifying Oracle Clusterware resources, or creating a database. The `ocrconfig -manualbackup` command exports the OCR content to a file format. You can then backup the export files created by `ocrconfig` as a part of the operating system backup using Oracle Secure backup, standard operating-system tools, or third-party tools.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about backing up the OCR

Use Cluster Health Monitor for Troubleshooting

Cluster Health Monitor (CHM) designed to detect and analyze operating system (OS) and cluster resource related degradation and failures in order to bring more explanatory power to many issues that occur in clusters where Oracle Clusterware and Oracle RAC are running such as node eviction. It continuously tracks the OS resource

consumption at each node, process, and device level. It collects and analyzes this cluster-wide data. In real time mode, when thresholds are hit, an alert is shown to the operator. For root cause analysis, historical data can be replayed to understand what was happening at the time of failure.

6

Configuring Oracle Database with Oracle RAC

This chapter contains the following topics:

- Configuring Oracle Database with Oracle RAC
- Configuring Oracle Database with Oracle RAC One Node
- Configuring Oracle Database with Oracle RAC on Extended Clusters

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide*

Configuring Oracle Database with Oracle RAC

The best practices discussed in this section apply to Oracle Database 12*c* with Oracle Real Application Clusters (Oracle RAC). These best practices build on the configuration best practices described in Chapter 4, "Configuring Oracle Database" and Chapter 5, "Configuring Oracle Database with Oracle Clusterware." These best practices are identical for the primary and standby databases if they are used with Data Guard in Oracle Database 12*c* with Oracle RAC and Data Guard MAA. Some best practices may use your system resources more aggressively to reduce or eliminate downtime. This can, in turn, affect performance service levels, so be sure to assess the impact in a test environment before implementing these practices in a production environment.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide*

Optimize Instance Recovery Time

Instance recovery is the process of recovering the redo thread from the failed instance. Instance recovery is different from crash recovery, which occurs when all instances accessing a database have failed. Crash recovery is the only type of recovery when an instance fails using a single-instance Oracle Database.

When using Oracle RAC, the SMON process in one surviving instance performs instance recovery of the failed instance.

In both Oracle RAC and single-instance environments, checkpointing is the internal mechanism used to bound Mean Time To Recover (MTTR). Checkpointing is the process of writing dirty buffers from the buffer cache to disk. With more aggressive checkpointing, less redo is required for recovery after a failure. Although the objective is the same, the parameters and metrics used to tune MTTR are different in a single-instance environment versus an Oracle RAC environment.

In a single-instance environment, you can set the FAST_START_MTTR_TARGET initialization parameter to the number of seconds the crash recovery should take. Note that crash recovery time includes the time to startup, mount, recover, and open the database.

Oracle provides several ways to help you understand the MTTR target your system is currently achieving and what your potential MTTR target could be, given the I/O capacity.

> **See Also:** The MAA white paper "Best Practices for Optimizing Availability During Unplanned Outages Using Oracle Clusterware and Oracle Real Application Clusters" for more information from the MAA Best Practices area for Oracle Database at
>
> http://www.oracle.com/goto/maa

Maximize the Number of Processes Performing Transaction Recovery

The FAST_START_PARALLEL_ROLLBACK parameter determines how many processes are used for transaction recovery, which is done after redo application. Optimizing transaction recovery is important to ensure an efficient workload after an unplanned failure. If the system is not CPU bound, setting this parameter to HIGH is a best practice. This causes Oracle to use four times the CPU_COUNT (4 X CPU_COUNT) parallel processes for transaction recovery. The default setting for this parameter is LOW, or two times the CPU_COUNT (2 X CPU_COUNT). Set the parameter as follows:

```
ALTER SYSTEM SET FAST_START_PARALLEL_ROLLBACK=HIGH SCOPE=BOTH;
```

More importantly, in addition to the system not being CPU bound, the system should not be I/O bound in order to set FAST_START_PARALLEL_ROLLBACK to HIGH, so that the workload of the database is not affected after an unplanned failure.

When FAST_START_PARALLEL_ROLLBACK is set to HIGH, a system with a large number of CPUs will spawn a lot of parallel recovery slaves which can substantially increase the IOPS rate. In this case the system should not be challenged for I/O before FAST_START_PARALLEL_ROLLBACK is set to HIGH.

> **See Also:** *Oracle Database VLDB and Partitioning Guide* for information about Parameters Affecting Resource Consumption for Parallel DML and Parallel DDL

Ensure Asynchronous I/O Is Enabled

Using asynchronous I/O is a best practice that is recommended for all Oracle Databases. For more information, see Section , "Set DISK_ASYNCH_IO Initialization Parameter".

Redundant Dedicated Connection Between the Nodes

Use redundant dedicated connections and sufficient bandwidth for public traffic, Oracle RAC interconnects, and I/O.

In Oracle terms, an extended cluster is a two or more node configuration where the nodes are separated in two physical locations. For an extended cluster and for other Oracle RAC configurations, separate dedicated channels on one fibre may be needed, or you can optionally configure Dense Wavelength Division Multiplexing (DWDM) to allow communication between the sites without using repeaters and to allow greater distances, greater than 10 km, between the sites. However, the disadvantage is that DWDM can be prohibitively expensive.

See Also: *Oracle Database 2 Day + Real Application Clusters Guide* for more information About Network Hardware Requirements

Configuring Oracle Database with Oracle RAC One Node

Oracle RAC One Node is a single instance of an Oracle Real Application Clusters (Oracle RAC) database that runs on one node in a cluster with an option to failover or migrate to other nodes in the same cluster. This option adds to the flexibility that Oracle offers for database consolidation. You can consolidate many databases into one cluster with minimal overhead while also providing the high availability benefits of failover protection, online rolling patch application, and rolling upgrades for the operating system and Oracle Clusterware.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Administering Oracle RAC One Node

Configuring Oracle Database with Oracle RAC on Extended Clusters

An Oracle RAC extended cluster is an architecture that provides extremely fast recovery from a site failure and allows for all nodes, at all sites, to actively process transactions as part of single database cluster. An extended cluster provides greater high availability than a local Oracle RAC cluster, but because the sites are typically in the same metropolitan area, this architecture may not fulfill all disaster recovery requirements for your organization.

The best practices discussed in this section apply to Oracle Database 11*g* with Oracle RAC on extended clusters, and build on the best practices described in Section , "Configuring Oracle Database with Oracle RAC."

Use the following best practices when configuring an Oracle RAC database for an extended cluster environment:

- Spread the Workload Evenly Across the Sites in the Extended Cluster
- Add a Third Voting Disk to Host the Quorum Disk
- Configure the Nodes to Be Within the Proximity of a Metropolitan Area
- Use Host-Based Storage Mirroring with Oracle ASM Normal or High Redundancy
- Additional Deployment Considerations for Extended Clusters

See Also:

- The white paper about extended clusters on the Oracle Real Application Clusters website at `http://www.oracle.com/technetwork/database/clustering/overview/index.html`
- *Oracle Database High Availability Overview* for a high-level overview, benefits, and configuration examples for Oracle RAC

Spread the Workload Evenly Across the Sites in the Extended Cluster

A typical Oracle RAC architecture is designed primarily as a scalability and availability solution that resides in a single data center. To build and deploy an Oracle RAC extended cluster, the nodes in the cluster are separated by greater distances. When configuring an Oracle RAC database for an extended cluster environment, you must:

- Configure one set of nodes at Site A and another set of nodes at Site B.
- Spread the cluster workload evenly across both sites to avoid introducing additional contention and latency into the design. For example, avoid client/server application workloads that run across sites, such that the client component is in site A and the server component is in site B.

Add a Third Voting Disk to Host the Quorum Disk

Most extended clusters have only two storage systems (one at each site). During normal processing each node writes and reads a disk heartbeat at regular intervals, but if the heartbeat cannot complete, all affected nodes are evicted from the cluster forcing them to restart their processes and retry to acquire access to the shared resources safely as a member. Thus, the site that houses the majority of the voting disks is a potential single point of failure for the entire cluster. For availability reasons, you should add a third site that can act as the arbitrator in case either: one site fails, or a communication failure occurs between the sites.

In some cases, you can also use standard NFS to support a third voting disk on an extended cluster. You can configure the quorum disk on inexpensive, low end, standard NFS mounted device somewhere on the network. Oracle recommends putting the NFS voting disk on a dedicated server which belongs to a production environment.

If you have an extended cluster and do not configure a third site, you must find out which of the two sites is the primary site. Then, if the primary site fails, you must manually restart the secondary site.

Note: Oracle Clusterware supports NFS, iSCSI, Direct Attached Storage (DAS), Storage Area Network (SAN) storage, and Network Attached Storage (NAS). If your system does not support NFS, use an alternative. For example, on Windows systems you can use iSCSI.

See Also: For more information, see the Technical Article "Using standard NFS to support a third voting file for extended cluster configurations" at

http://www.oracle.com/technetwork/database/clustering/overview/index.html

Configure the Nodes to Be Within the Proximity of a Metropolitan Area

Extended clusters provide the highest level of availability for server and site failures when data centers are in close enough proximity to reduce latency and complexity. The preferred distance between sites in an extended cluster is within a metropolitan area. High internode and interstorage latency can have a major effect on performance and throughput. Performance testing is mandatory to assess the impact of latency. In general, distances of 50 km or less are recommended.

Testing has shown the distance (greatest cable stretch) between Oracle RAC cluster nodes generally affects the configuration, as follows:

- Distances less than 10 km can be deployed using normal network cables.
- Distances equal to or more than 10 km require Dense Wavelength Division Multiplexing (DWDM) links. If a DWDM or CWDM is used then these should be directly connected using a dedicated switch on either side.

- Distances from 10 to 50 km require storage area network (SAN) buffer credits to minimize the performance impact due to the distance. Otherwise, the performance degradation due to the distance can be significant.
- For distances greater than 50 km, there are not yet enough proof points to indicate the effect of deployments. More testing is needed to identify what types of workloads could be supported and what the effect of the chosen distance would have on performance.

Some additional considerations are that Interconnect, SAN, and IP Networking must be kept on separate channels, each with required redundancy. Redundant connections must not share the same Dark Fiber (if used), switch, path, or even building entrances because cables can be cut.

Because Oracle Clusterware is used, a single subnet should be set up across the sites on which the public network should reside so that VIPs can fail over from one site to another.

Use Host-Based Storage Mirroring with Oracle ASM Normal or High Redundancy

Use host-based mirroring with Oracle ASM normal or high redundancy configured disk groups so that a storage array failure does not affect the application and database availability.

Oracle recommends host-based mirroring using Oracle ASM to internally mirror across the two storage arrays. Implementing mirroring with Oracle ASM provides an active/active storage environment in which system write I/Os are propagated to both sets of disks, making the disks appear as a single set of disks that is independent of location. Do not use array-based mirroring because only one storage site is active, which makes the architecture vulnerable to this single point of failure and longer recovery times.

The Oracle ASM volume manager provides flexible host-based mirroring redundancy options. You can choose to use external redundancy to defer the mirroring protection function to the hardware RAID storage subsystem. The Oracle ASM normal and high-redundancy options allow two-way and three-way mirroring, respectively.

> **Note:** Array based mirroring can be used in an Oracle RAC extended cluster. Using this approach has the result that the two mirror sites will be in an active-passive configuration and this will result in a complete outage if one site fails. Service becomes available if the remaining mirror site is brought up. For this reason array based mirroring is not recommended from an HA perspective. To work with two active sites, host based mirroring is recommended.

Beginning with Oracle Database Release 11*g*, Oracle ASM includes a preferred read capability that ensures that a read I/O accesses the local storage instead of unnecessarily reading from a remote failure group. When you configure Oracle ASM failure groups in extended clusters, you can specify that a particular node reads from a failure group extent that is closest to the node, even if it is a secondary extent. This is especially useful in extended clusters where remote nodes have asymmetric access for performance, thus leading to better usage and lower network loading. Using preferred read failure groups is most useful in extended clusters.

The `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter value is a comma-delimited list of strings that specifies the failure groups that should be preferentially read by the given instance. This parameter is instance specific, and it is

generally used only for clustered Oracle ASM instances. It's value can be different on different nodes. For example:

```
diskgroup_name1.failure_group_name1, ...
```

See Also:

- Section , "Use Automatic Storage Management (Oracle ASM) to Manage Database Files"
- *Oracle Automatic Storage Management Administrator's Guide* for information about configuring preferred read failure groups with the `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter

Additional Deployment Considerations for Extended Clusters

Consider the following additional factors when implementing an extended cluster architecture:

- Network, storage, and management costs increase.
- Write performance incurs the overhead of network latency. Test the workload performance to assess impact of the overhead.
- Because this is a single database without Oracle Data Guard, there is no protection from data corruption or data failures.
- The Oracle release, the operating system, and the clusterware used for an extended cluster all factor into the viability of extended clusters.
- When choosing to mirror data between sites:
  - Host-based mirroring requires a clustered logical volume manager to allow active/active mirrors and thus a primary/primary site configuration. Oracle recommends using Oracle ASM as the clustered logical volume manager.
  - Array-based mirroring allows active/passive mirrors and thus a primary/secondary configuration.
- Extended clusters need additional destructive testing, covering
  - Site failure
  - Communication failure
- For full disaster recovery, complement the extended cluster with a remote Data Guard standby database, because this architecture:
  - Maintains an independent physical replica of the primary database
  - Protects against regional disasters
  - Protects against data corruption and other potential failures
  - Provides options for performing rolling database upgrades and patch set upgrades

7

Configuring Backup and Recovery

A data protection plan is not complete without a sound backup and recovery strategy to protect against system and storage failures. Oracle delivers a comprehensive data protection suite for backup and recovery of Oracle database and unstructured, application files.

The primary focus of this chapter is best practice configuration for backup and recovery for the Oracle database. File system data protection offerings are introduced along with pointers on where to find more information.

This chapter contains the following topics:

- Oracle Database Backup and Recovery Products and Features
- Backup and Recovery Configuration and Administration Best Practices
- Backup to Disk Best Practices
- Backup to Tape Best Practices
- Backup and Recovery Operations and Maintenance Best Practices
- Backup Files Outside the Database

Table 7–1provides a quick reference summary of the Oracle backup and recovery suite.

Table 7–1 Backup and Recovery Summary

Technology	Recommended use with Oracle Database	Recommended use with File System Data	Comments
Zero Data Loss Recovery Appliance	Yes	No	Enterprise-wide Oracle Data Protection (Backup and Recovery) Solution
Recovery Manager (RMAN)	Yes	No	Native backup utility for the Oracle Database
Oracle Secure Backup	Yes	Yes	Tape backup management software
Oracle Database Backup Service	Yes	No	Backup to Oracle Public Cloud
Oracle Secure Backup Cloud Module for Amazon S3	Yes	No	Backup to Amazon S3 storage
Flashback Technologies	Yes	No	Logical error correction leveraging undo data of the Oracle Database
Flashback Database	Yes	No	Continuous Data Protection (CDP) leveraging flashback logs

Table 7–1 (Cont.) Backup and Recovery Summary

Technology	Recommended use with Oracle Database	Recommended use with File System Data	Comments
Automatic Clustered File System (ACFS) Snapshots (for file system clones and disaster recovery)	No	Yes	Read-only or read/write copy-on-write version of the file system. Replication available.
ZFS Snapshots (for database clones such as dev/test)	Yes	Yes	Read-only or read/write copy-on-write version of the database for testing and development
ZFS Snapshots for backup/restore	No	Yes	Read-only or read/write copy-on-write version of the file system for testing, development and backup

Oracle Database Backup and Recovery Products and Features

This section discusses the motivation and tools for maintaining good database backups, for using Oracle database recovery features, and for using backup options and strategies made possible with Oracle database features.

Understand When to Use Backups

Using backups to resolve an unscheduled outage of a production database may not allow you to meet your Recovery Time Objective (RTO) and Recovery Point Objective (RPO) or service-level requirements. For example, some outages are handled best by using Flashback Database or a standby database. However, some situations require using database backups, including the sample situations shown in Table 7–2.

Table 7–2 Sample Situations that Require Database Backup

Situations that require Database Backup	Description
Setting Up the Initial Data Guard Environment	During initial setup of a standby database, you can either use a backup of the primary database that is made accessible to the secondary site to create the initial standby database, or use RMAN network-enabled database duplication to create the standby database without the need for a pre-existing backup. To perform an over-the-network duplication you must include the RMAN `FROM ACTIVE DATABASE` option in the `DUPLICATE` command.
Recovering from Data Failures Using File or Block Media Recovery	When a block corruption, media failure, or other physical data failure occurs in an environment that does not include Data Guard, the only method of recovery is to restore from existing backups.
Resolving a Double Failure	A double failure scenario is a situation that affects the availability of both the production and the standby databases. An example of a double failure scenario is a site outage at the secondary site, which eliminates fault tolerance, followed by a media failure on the production database. Whether the standby must be re-created depends on the type of outage at the secondary site. If the secondary site outage was temporary and did not involve the physical destruction of files, then after the secondary site is brought back online it can continue to receive redo data from the production database. Otherwise, the resolution of this situation is to re-create the production database from an available backup and then re-create the standby database. Some multiple failures, or more appropriately disasters, such as a primary site outage followed by a secondary site outage, might require the use of backups that exist only in an offsite location. Developing and following a process to deliver and maintain backup tapes at an offsite location is necessary to restore service in this worst case scenario.

See Also:

- *Oracle Data Guard Concepts and Administration* for information about creating a standby database
- *Oracle Database Backup and Recovery User's Guide* for information about the `DUPLICATE` command

Use Zero Data Loss Recovery Appliance to Back Up Database Files

Zero Data Loss Recovery Appliance is a new data protection solution integrated with the Oracle Database that eliminates data loss exposure and dramatically reduces backup and data protection overhead on production servers. The recovery appliance easily protects all databases in the data center with a scalable cloud architecture, ensures end-to-end data validation, and automates the management of the entire data protection lifecycle for all Oracle databases through a unified Enterprise Manager interface.

The recovery appliance serves as a destination for real-time redo transport for all Oracle Database 11g and 12c databases, thus providing data loss protection until the last sub-second for transactional data.

In conjunction with Oracle Recovery Manager (RMAN), the Recovery Appliance minimizes the impact of running backups against the database by only requiring a single incremental Level 0 (full) database backup on day 1, and then Incremental Level 1 database backups thereafter.

When a database needs to be restored, the Recovery Appliance constructs a level 0 copy of the data file as it would have been at the time of the most recent incremental level 1 backup, thereby reducing the amount of data that needs to be recovered after the database restore is completed.

The Recovery Appliance also manages the transfer of data from the appliance to a tape library using Oracle Secure Backup (OSB) for purposes of tape vaulting, as well as replicating backups to another recovery appliance for faster offsite data protection.

Oracle ZDLRA offers many advantages over traditional backup methods including:

- Offloaded incremental forever backup strategy
- Real-time redo transport, providing data loss protection until the last sub-second of transactions
- End-to-end data backup and recovery service level agreement (SLA) validation
- Efficient use of tape resources
- Replication of backups to an offsite Recovery Appliance
- Centralized administration and monitoring using Oracle Enterprise Manager Cloud Control
- Reporting on capacity planning, database backups, and their recovery service level agreements

> **See also:** Zero Data Loss Recovery Appliance documentation library (especially the *Zero Data Loss Recovery Appliance Administrator's Guide* and *Zero Data Loss Recovery Appliance Protected Databases Configuration Guide*)

Use Recovery Manager (RMAN) to Backup Database Files

Recovery Manager (RMAN) is Oracle's utility to backup and recover the Oracle Database. Because of its tight integration with the database, RMAN determines automatically what files must be backed up. More importantly, RMAN knows what files must be restored for media-recovery operations. RMAN uses server sessions to perform backup and recovery operations and stores metadata about backups in a repository. RMAN offers many advantages over typical user-managed backup methods, including:

- Online database backups without placing tablespaces in backup mode
- Efficient block-level incremental backups
- Data block integrity checks during backup and restore operations
- Test backups and restores without actually performing the operation
- Synchronize a physical standby database with the primary database

RMAN automates backup and recovery. While user-managed methods require you to:

- Locate backups for each data file
- Copy backups to the correct place using operating system commands
- Choose which logs to apply

RMAN fully automates these backup and recovery tasks.

There are also capabilities of Oracle backup and recovery that are only available when using RMAN, such as automated tablespace point-in-time recovery and block media recovery.

See Also: For more information, see the following chapters:

- *Oracle Database Backup and Recovery User's Guide* for information about performing block media recovery
- *Oracle Database Backup and Recovery User's Guide* for information about performing RMAN Tablespace Point-in-Time Recovery (TSPITR)
- *Oracle Data Guard Concepts and Administration* for information about Using RMAN Incremental Backups to Roll Forward a Physical Standby Database

Use Oracle Secure Backup for Backups to Tape

Oracle Secure Backup delivers unified data protection for heterogeneous environments with a common management interface across the spectrum of servers. Protecting both Oracle databases and unstructured data, Oracle Secure Backup provides centralized tape backup management for your entire IT environment, including:

- Oracle database through the Oracle Secure Backup built-in integration with Recovery Manager (RMAN)
- File system data protection: For UNIX, Windows, and Linux servers
- Network Attached Storage (NAS) data protection leveraging the Network Data Management Protocol (NDMP)

Oracle Secure Backup is integrated with RMAN providing the media management layer (MML) for Oracle database tape backup and restore operations. The tight integration between these two products delivers high-performance Oracle database tape backup.

Specific performance optimizations between RMAN and Oracle Secure Backup that reduce tape consumption and improve backup performance are:

- Unused block compression: Eliminates the time and space usage needed to backup unused blocks
- Backup undo optimization: Eliminates the time and space usage needed to backup undo that is not required to recover the current backup.

You can manage the Oracle Secure Backup environment using the command line, the Oracle Secure Backup Web tool, and Oracle Enterprise Manager.

Using the combination of RMAN and Oracle Secure Backup provides an end-to-end tape backup solution, eliminating the need for third-party backup software.

Optionally, during Zero Data Loss Recovery Appliance deployment, Oracle Secure Backup can be configured and integrated automatically with the Recovery Appliance.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more information about unused block compression and backup undo optimization
- The Oracle Secure Backup platform and NAS and tape device compatibility matrixes in

  http://www.oracle.com/technetwork/database/availability/osb-12-1-platforms-2420299.pdf

Using Oracle Secure Backup for Backups to Amazon S3 Storage

Users can take advantage of the Internet-based data storage services offered by Amazon Web Services (AWS) Simple Storage Service (S3) for their database backup needs. The OSB Cloud Module enables RMAN to use S3 as a repository for Oracle Database backups. This provides an easy-to-manage, cost-efficient, and scalable alternative to maintaining in-house data storage and a local, fully configured backup infrastructure. RMAN with the OSB Cloud module is also the recommended means of backing up an Oracle Database that is running on AWS's Elastic Compute Cloud (EC2).

Users must establish an account with AWS to pay for their storage and network transfer costs as appropriate. Additionally, users must consider their security requirements and network resources, and configure their backups appropriately. Since backups to S3 may travel to AWS on the public Internet, and will be stored on a public cloud facility, it may be necessary to encrypt them to protect the data in transit and at rest at S3. This requirement may not apply for certain databases or in certain configurations, for example when the database runs on EC2 or inside AWS's Virtual Private Cloud (VPC).

Users must also configure the degree of network parallelism (using RMAN channels) to match the network capabilities between the database and S3 (which may include a portion of the public Internet). Multiple simultaneous channels may achieve the highest throughput overall, as RMAN takes full advantage of such parallelism.

Use Restore Points for Creating Database Snapshots

Oracle provides restore points and guaranteed restore points:

- **Restore points** protect against logical failures at risky points during database maintenance. Creating a normal restore point assigns a restore point name to a specific point in time or SCN, that is a snapshot of the data as of that time. Normal restore points are available with Flashback Table, Flashback Database, and all RMAN recovery-related operations.
- **Guaranteed restore points** are recommended for database-wide maintenance such as database or application upgrades, or running batch processes. Guaranteed restore points are integrated with Flashback Database and enforce the retention of all flashback logs required for flashing back to the guaranteed restore point. After maintenance activities complete and the results are verified, you should delete guaranteed restore points that are no longer needed to reclaim flashback log space.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for more information about using restore points and guaranteed restore points with a Flashback Database

Backup and Recovery Configuration and Administration Best Practices

This section describes best practices for determining backup frequency, using the RMAN recovery catalog, and for using Oracle database backup options such as Block Change Tracking.

Determine a Backup Frequency and Retention Policy

It is important to determine a backup frequency policy and to perform regular backups. A backup retention policy helps ensure that needed data is not destroyed.

Factors Determining Backup Frequency Frequent backups are essential for any recovery scheme. You should base the frequency and content of backups on the following criteria:

- **Criticality of the data:** The Recovery Point Objective (RPO) determines how much data your business can acceptably lose if a failure occurs. The more critical the data, the lower the RPO and the more frequently data should be backed up. If you are going to back up certain tablespaces more often than others, with the goal of getting better RPO for those tablespaces, then you also must plan for doing TSPITR as part of your recovery strategy. This requires considerably more planning and practice than DBPITR, because you must ensure that the tablespaces you plan to TSPITR are self-contained.
- **Estimated repair time**: The Recovery Time Objective (RTO) determines the acceptable amount of time needed for recovery. Repair time is dictated by restore time plus recovery time. The lower the RTO, the higher the frequency of backups, that is, backups are more current, thereby reducing recovery time.
- **Volume of changed data:** The rate of database change effects how often data is backed up:
  - For read-only data, perform backups frequently enough to adhere to retention policies.
  - For frequently changing data, perform backups more often to reduce the RTO.

To simplify database backup and recovery, the Oracle Suggested Recovery Appliance Backup Strategy for the Enterprise uses RMAN to send incremental database backups and redo using Data Guard transport to the Recovery Appliance, while the Oracle Suggested Backup Strategy uses the fast recovery area, incremental backups, and incrementally updated backup features for a particular database. After the initial image copy backup to the FRA, only the changed blocks are captured in the incremental backups thereafter and subsequently applied to the image copy, thereby updating the copy to the most current incremental backup time (that is, incrementally updating the backup).

See Also:

- *Oracle Database Backup and Recovery User's Guide* for information about Performing RMAN Tablespace Point-in-Time Recovery (TSPITR)
- *Oracle Database Backup and Recovery User's Guide* for information about Performing Database Point-in-Time Recovery
- *Oracle Database 2 Day DBA* for information about Using the Oracle Suggested Backup Strategy

Establishing a Backup Retention Policy A backup retention policy, which is implemented as a Protection Policy on the Recovery Appliance, is a rule set regarding which backups must be retained, on disk or other backup media, to meet recovery and other requirements. It may be safe to delete a specific backup because it has been superseded by more recent backups or because it has been stored on tape. You may also have to retain a specific backup on disk for other reasons such as archival or regulatory requirements. A backup that is no longer needed to satisfy the backup retention policy is said to be obsolete.

Base your backup retention policy on redundancy or on a recovery window:

- In a redundancy-based retention policy, specify a number *n* such that you always keep at least *n* distinct backups of each file in your database.

- In a recovery window-based retention policy, specify an earlier time interval, for example, one week or one month, and keep all backups required to let you perform point-in-time recovery to any point during that window.

Keeping Archival Backups Some businesses must retain some backups for much longer than their day-to-day backup retention policy. RMAN allows for this with the Long-term Archival Backup feature. Rather than becoming obsolete according to the database's backup retention policy, archival backups either never become obsolete or become obsolete when their time limit expires.

You can use the RMAN `BACKUP` command with the `KEEP` option to retain backups for longer than your ordinary retention policy. This option specifies the backup as an archival backup, which is a self-contained backup that is exempt from the configured retention policy. This allows you to retain certain backups for much longer than usual, when needed for such reasons as satisfying statutory retention requirements. Using the `KEEP FOREVER` option, a recovery catalog is required because the backup records eventually age out of the control file (otherwise, without a recovery catalog, loss may occur when you retain backups for much longer than usual using the database control file). Only the archived redo log files required to make an archival backup consistent are retained. For more information about the RMAN recovery catalog, see Section , "Use an RMAN Recovery Catalog".

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for information about Archival Backups for Long-Term Storage

Use an RMAN Recovery Catalog

To protect and keep backup metadata for longer retention times than can be accommodated by the control file, you can create a recovery catalog. When using the Recovery Appliance, the Recovery Appliance is the recovery catalog for all databases. When not using a Recovery Appliance, you should create the recovery catalog schema in a dedicated standalone database. Do not locate the recovery catalog with other production data. If you use Oracle Enterprise Manager, you can create the recovery catalog schema in the Oracle Enterprise Manager repository database.

The advantages of using a recovery catalog include:

- Storing backup information for a longer retention period than what can be feasibly stored in the control file. If the control file is too small to hold additional backup metadata, then existing backup information is overwritten, making it difficult to restore and recover using those backups.
- Stores metadata for multiple databases.
- Offloading backups to a physical standby database and using those backups to restore and recover the primary database. Similarly, you can back up a tablespace on a primary database and restore and recover it on a physical standby database. Note that backups of logical standby databases are not usable at the primary database.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for more information about RMAN repository and the recovery catalog

Create Backups in NOCATALOG Mode and Then RESYNC CATALOG When Not Using Recovery Appliance

When creating backups to disk or tape, use the target database control file as the RMAN repository so that the success of the backup does not depend on the

availability of the database connection to the recovery catalog. To use the target database control file as the RMAN repository, run RMAN with the `NOCATALOG` option. Immediately after the backup is complete, the new backup information stored in the target database control file should be synchronized to the recovery catalog using the `RESYNC CATALOG` command.

> **See Also:** *Oracle Database Backup and Recovery Reference* for more information about the `RESYNC CATALOG` command

Enable Block Change Tracking for Incremental Backups

Oracle database includes the `BLOCK CHANGE TRACKING` feature for incremental backups which improves incremental backup performance by keeping track of which database blocks have changed since the previous backup. If `BLOCK CHANGE TRACKING` is enabled then RMAN uses the block change tracking file to identify which blocks to include in an incremental backup. This avoids the need to scan every block in the data file, reducing the number of disk reads during backup.

Starting with Oracle Database 11*g*, you can enable `BLOCK CHANGE TRACKING` on both the primary and physical standby databases. You should enable change tracking for any database where incremental backups are being performed. For example, if backups have been completely offloaded to a physical standby database, then Block Change Tracking should be enabled for that database (this requires Active Data Guard). If backups are being performed on both the primary and physical standby databases, then enable Block Change Tracking for both databases.

> **See Also:**
>
> - *Oracle Database Backup and Recovery User's Guide* for information about Enabling and Disabling Block Change Tracking
> - *Oracle Database Backup and Recovery Basics* for more information about Block Change Tracking

Enable Autobackup for the Control File and Server Parameter File

You should configure RMAN to automatically back up the control file and the server parameter file (SPFILE) whenever the database structure metadata in the control file changes or when a backup record is added.

The control file autobackup option enables RMAN to recover the database even if the current control file, catalog, and SPFILE are lost. Enable the RMAN autobackup feature with the `CONFIGURE CONTROLFILE AUTOBACKUP ON` statement.

You should enable autobackup for both the primary and standby databases. For example, after connecting to the primary database, as the target database, and the recovery catalog, issue the following command:

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

> **See Also:**
>
> - *Oracle Database Backup and Recovery User's Guide* for information about Configuring Control File and Server Parameter File Autobackups
> - *Oracle Data Guard Concepts and Administration* for more information about RMAN Configurations at a Standby Database Where Backups are Performed

Offload Backups to a Physical Standby Database

In an Oracle Data Guard configuration you can offload the process of backing up control files, data files, and archived redo log files to a physical standby database system, thereby minimizing the effect of performing backups on the primary system. You can use these backups to recover the primary or standby database.

Note: Backups of logical standby databases are not usable on the primary database.

See Also: *Oracle Data Guard Concepts and Administration* for information about using RMAN to back up and restore files

Set UNDO Retention for Flashback Query and Flashback Table Needs

To ensure a database is enabled to use Flashback Query, Flashback Versions Query, and Flashback Transaction Query, implement the following:

- Set the `UNDO_MANAGEMENT` initialization parameter to `AUTO`. This ensures the database is using an undo tablespace.
- Set the `UNDO_RETENTION` initialization parameter to a value that allows `UNDO` to be kept for a length of time that allows success of your longest query back in time or to recover from human errors.
- Set the `RETENTION GUARANTEE` clause for the undo tablespace to guarantee that unexpired undo will not be overwritten.

The Flashback Table also relies on the undo data to recover the tables. Enabling Automatic Undo Management is recommended and the `UNDO_RETENTION` parameter must be set to a period for which the Flashback Table is needed. If a given table does not contain the required data after a Flashback Table, it can be flashed back further, flashed forward, or back to its original state, if there is sufficient `UNDO` data.

See Also:

- *Oracle Database Advanced Application Developer's Guide* for information about Flashback
- *Oracle Database Advanced Application Developer's Guide* for information about Flashback Query

Backup to Disk Best Practices

Review the following priorities to determine your disk backup strategy:

- Overall backup time
- Impact to resource consumption
- Space used by the backup
- Recovery time

Table 7–3 compares different backup alternatives against the different priorities you might have. Using Table 7–3 as a guide, you can choose the best backup approach for your specific business requirements. You might want to minimize backup space while sacrificing recovery time. Alternatively, you might choose to place a higher priority on recovery and backup times while space is not an issue.

Table 7–3 Comparing Backup to Disk Options

Backup Option	Overall Backup Time 1: Fastest 5: Slowest	Impact on Resource Consumption 1: Lowest 5: Highest	Space Used by Backup 1: Least 5: Most	Restore Time 1: Fastest 5: Slowest
Oracle Suggested Recovery Appliance Backup Strategy	1	1	1 (backups stored in a compressed format)	2
Data file image copy	5	5	5	1 No restore (switch to fast recovery area copy)
Full or level 0 backup set	4	4	3	3
Differential incremental backup set (level 1); applied to previous level 0 and level 1 backups during recovery	1	1	1	5
Cumulative incremental backup set (level 1); applied to previous level 0 backup during recovery	2	2	2	4
Incrementally updated backup (level 1); incremental applied to image copy after backup	3	1	5	1 No restore (switch to fast recovery area copy)

Backup to Disk: Best Practices for Optimizing Recovery Times If restore time is your primary concern then perform either a database copy or an incremental backup with immediate apply of the incremental to the copy. These are the only options that provide an immediate usable backup of the database, which you then must recover only to the time of the failure using archived redo log files created since the last incremental backup was performed.

Backup to Disk: Best Practices for Minimizing Space Usage If space usage is your primary concern then perform Oracle Suggested Recovery Appliance Backup Strategy, as this practice stores the database backups off the machine where the database is located.

Alternatively, an incremental backup with a deferred apply of the incremental to the copy. If you perform a cumulative level 1 incremental backup, then it stores only those blocks that have been changed since the last level 0 backup:

- With a cumulative incremental backup apply only the last level 1 backup to the level 0 backup.
- With a differential incremental backup apply all level 1 backups to the level 0 backup.

A cumulative incremental backup usually consumes more space in the fast recovery area than a differential incremental backup.

Backup to Disk: Best Practices for Minimizing System Resource Consumption (I/O and CPU) If system resource consumption is your primary concern then an Oracle Suggested Recovery Appliance Backup Strategy or incremental backup with a Block Change Tracking enabled consumes the least amount of resources on the database.

Example

For many applications, only a small percentage of the entire database is changed each day even if the transaction rate is very high. In many cases, applications repeatedly modify the same set of blocks; so, the total unique, changed block set is small.

For example, a database contains about 600 GB of user data, not including temp files and redo logs. Every 24 hours, approximately 2.5% of the database is changed, which is approximately 15 GB of data. In this example, with the database being stored on the source system, MAA testing recorded the following results:

- Level 0 backup takes 180 minutes, including `READ`s from the data area and `WRITE`s to the fast recovery area
- Level 1 backup takes 20 minutes, including `READ`s from the data area and `WRITE`s to the fast recovery area
- Rolling forward and merging an existing image copy in the fast recovery area with a newly created incremental backup takes only 45 minutes, including `READ`s and `WRITE`s from the fast recovery area.

In this example, the level 0 backup (image copy) takes 180 minutes. This is approximately the same amount of time it takes to perform a full backup set.

Subsequent backups are level 1 (incremental), which take 20 minutes, so the potential impact on the data area is reduced. That backup is then applied to the existing level 0 backup, which takes 45 minutes. This process does not perform I/O to the data area, so there is no impact (assuming the fast recovery area and data area use separate storage). The total time to create the incremental backup and apply it to the existing level 0 backup is 65 minutes (20+45).

The result is the same using incrementally updated backups or full backup sets, a full backup of the database is created. The incremental approach takes 115 minutes less time (64% less) than simply creating a full backup set. In addition, the I/O impact is less, particularly against the data area which should have less detrimental effect on production database performance.

Thus, for this example when you compare always taking full backups versus starting with a level 0 backup, performing only incremental backups, and then rolling forward the level 0 backup, the net savings are:

- 115 minutes or 64% time savings to create a complete backup
- Reduced I/O on the database during backups

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for more information about backing up the database

Backup to Tape Best Practices

Recovery Manager (RMAN) provides automated disk backup for the Oracle database and is integrated with media management products such as Oracle Secure Backup for backup to tape. Whether your Oracle database backup strategy uses disk, tape, or

both, the combination of RMAN and Oracle Secure Backup delivers a comprehensive solution to meet your specific requirements.

Initial RMAN Oracle Secure Backup Configuration

When installing Oracle Secure Backup, the System Backup Tape (SBT) libraries for RMAN tape backups are automatically linked. Using Oracle Enterprise Manager Database Control you can manage the Oracle Secure Backup backup domain from tape vaulting to backup and restore operations. The tight integration between RMAN, Oracle Enterprise Manager Database Control, and Oracle Secure Backup makes initial configuration a simple process.

Perform the following four steps to perform initial configuration and prepare to backup a database to tape:

1. Define your Oracle Secure Backup Administrative Server in Oracle Enterprise Manager Database Control enabling the Oracle Secure Backup domain to be managed through Oracle Enterprise Manager.
2. Pre-authorize an Oracle Secure Backup user for use with RMAN allowing the RMAN backup/restore be performed without having to explicitly login to Oracle Secure Backup.
3. Set-up media policies in Oracle Secure Backup to be used for RMAN backups.
4. Establish RMAN backup settings such as parallelism and compression.

Note: If you use Oracle Secure Backup or tape-side compression, do not also use RMAN compression.

See Also: *Oracle Secure Backup Administrator's Guide* for more information about using Recovery Manager with Oracle Secure Backup

Define Oracle Secure Backup Media Policies for Tape Backups

Once backup data stored on tape is no longer needed, its lifecycle is complete and the tape media can be reused. Management requirements during a tape's lifecycle (retention period) may include duplication and vaulting across multiple storage locations. Oracle Secure Backup provides effective media lifecycle management through user-defined media policies, including:

- Retention
- Tape duplication
- Vaulting: rotation of tapes between multiple locations

Media lifecycle management may be as simple as defining appropriate retention settings or more complex to include tape duplication with the original and duplicate(s) having different retention periods and vaulting requirements. Oracle Secure Backup media families, often referred to as tape pools, provide the media lifecycle management foundation.

The best practice recommendation is to leverage content-managed media families which use defined RMAN retention parameters associated with the database to determine when the tape may be reused (effectively an expired tape). A specific expiration date is not associated with content-managed tapes as is done with time-managed. The expiration or recycling of these tapes is based on the attribute

associated with the backup images on the tape. All backup images written to content-managed tapes automatically have an associated "content-manages reuse" attribute. Since the recycling of content-managed tapes adheres to user-defined RMAN retention settings, RMAN instructs Oracle Secure Backup when to change the backup image attribute to "deleted".

The RMAN `DELETE OBSOLETE` command communicates which backup pieces (images) are no longer required to meet the user-defined RMAN retention periods. Once Oracle Secure Backup receives this communication, the backup image attribute is changed to "deleted". The actual backup image is not deleted but the attribute is updated within the Oracle Secure Backup catalog. Once all backup images on tape have a deleted attribute, Oracle Secure Backup considers the tape eligible for reuse, similar to that of an expired time-managed tape.

Oracle Secure Backup provides policy-based media management for RMAN backup operations through user-defined Database Backup Storage Selectors. One Database Backup Storage Selector (SSEL) may apply to multiple databases or multiple SSELs may be associated with a single database. For example, you would create two SSEL for a database when using RMAN duplexing and each copy should be written to a different media family. The SSEL contains the following information:

- Database name / ID or applicable to all databases
- Hostname or applicable to all hosts
- Content: archive logs, full, incremental, autobackup or applicable to all
- RMAN copy number (applicable when RMAN duplexing is configured)
- Media family name
- Name(s) of devices to which operations are restricted (if no device restrictions are configured, Oracle Secure Backup uses any available device)
- Wait time (duration) for available tape resources
- Encryption setting

Oracle Secure Backup automatically uses the storage selections defined within a SSEL without further user intervention. To override the storage selections for one time backup operations or other exceptions, define alternate media management parameters in the RMAN backup script. For more information, see *Oracle Secure Backup Administrator's Guide*.

Create Tape Backups from the Fast Recovery Area

You can easily backup the Fast Recovery Area (FRA) to tape with the RMAN command: `BACKUP RECOVERY AREA`. Using this disk to tape backup method instead of performing a separate backup of the production database to tape provides a few distinct advantages:

- Saves tape consumption by creating an optimized backup of the Fast Recovery Area (FRA) thereby eliminating unnecessary backup of files already protected on tape
- Enables RMAN to use better restore intelligence from disk then tape as necessary, otherwise, RMAN would restore from the most recent backup regardless of media type
- Reduces I/O on the production database since the FRA uses a separate disk group

Upon restoration, RMAN automatically selects the most appropriate backup to restore from disk or tape. If the required backup is on tape, RMAN would restore or recovery

the database directly from tape media through integration with Oracle Secure Backup. As RMAN has intimate knowledge of what files are necessary for recovery, restoration from disk or tape is an automated process.

While it is possible to backup the FRA or other RMAN disk backup to tape outside of RMAN by performing a file system backup of the disk area using the media management software, it is not recommended. If RMAN is not aware of the tape backup then restoration is an error-prone, manual process:

1. DBA must determine what files are needed for the restoration.
2. Media manager administrator would then restore designated files from tape backups to a disk location.
3. Once files on disk, DBA would initiate an RMAN restore or recovery from the disk location.

The combination of RMAN and Oracle Secure Backup provides an integrated Oracle database tape backup solution.

> **See Also:** *Oracle Database 2 Day DBA* for more information about the Fast Recovery Area

Managing Offsite Backup Tapes

Backup tapes are highly portable and often stored at offsite locations for disaster recovery purposes. These tapes are first created from within a tape device but often are removed from the hardware device. Once backup tapes are removed from a hardware device the tapes may be stored in an on-site or offsite location. You can effectively manage tape movement between multiple locations using Oracle Secure Backup rotation policies.

For Oracle database restoration, a restore request is submitted from RMAN to Oracle Secure Backup. If the tapes are within the library, the restore begins immediately assuming device availability. However, if the tapes needed for restore could be offsite; you may want to confirm the location of tapes before you issue the restore command. With RMAN and Oracle Secure Backup you can easily do so by issuing the following RMAN command(s):

- `RESTORE DATABASE PREVIEW` command provides a list of tapes needed for restoration which are offsite.
- `RESTORE DATABASE PREVIEW RECALL` command initiates a recall operation through Oracle Secure Backup to return the tapes from offsite to the tape device for restoration. Once the tapes are on-site, you can begin the RMAN restore operation.

> **See Also:**
>
> - *Oracle Database Backup and Recovery User's Guide* for more information about recalling offsite backups with RMAN
> - *Oracle Database Backup and Recovery User's Guide* for more information about previewing backups used in restore operations

Backup and Recovery Operations and Maintenance Best Practices

This section outlines procedures to regularly check for corruption of data files using the Data Recovery Advisor, for testing recovery procedures, and for backing up the recovery catalog database.

Use Read Only Tablespaces

Historical data that is not subject to change can be stored in Read Only tablespaces. A Read Only Tablespace can be backed up once, and then no additional backups need to be taken for those data files. This can reduce the amount of data that needs to be backed up during incremental level 0 or full backups.

Do Not Compress Data That Has Already Been Compressed

If the database contains compressed data such as HCC objects, then RMAN compression should not be used in an attempt to further compress the data. When RMAN compression is used, RMAN backups typically become CPU bound as opposed to IO bound and the backup operations can take between 2 and 50 times longer to complete depending upon the database and RMAN compression algorithm being used.

Use Section Size Parameter to Break Up BigFile DataFiles Backup

Using the RMAN Section Size option can significantly speed up database backups by allocating different parts of a big file datafile to multiple RMAN channels for processing in parallel.

Diagnose Data Failures and Present Repair Options

Data Recovery Advisor is an Oracle Database tool that automatically diagnoses data failures, determines and presents appropriate repair options, and executes repairs at the user's request. In this context, a data failure is a corruption or loss of persistent data on disk. By providing a centralized tool for automated data repair, Data Recovery Advisor improves the manageability and reliability of an Oracle database and thus helps reduce the mean time to recover (MTTR).

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for information about using Data Recovery Advisor

Regularly Check Database Files for Corruption

When using Oracle Suggested Recovery Appliance Backup Strategy, regular checks of the database backups stored on the appliance are performed automatically. Alternatively, use the RMAN `VALIDATE` command to regularly check database files for block corruption that has not yet been reported by a user session or by normal backup operations. RMAN scans the specified files and checks for physical and logical errors, but does not actually perform the backup or recovery operation. Oracle database records the address of the corrupt block and the type of corruption in the control file. Access these records through the `V$DATABASE_BLOCK_CORRUPTION` view, which can be used by RMAN block media recovery.

To detect all types of corruption that are possible to detect, specify the `CHECK LOGICAL` option.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for information about Validating Database Files and Backups

Periodically Test Recovery Procedures

Complete, successful, and tested backups are fundamental to the success of any recovery. Create test plans for different outage types. Start with the most common outage types and progress to the least probable. Using the RMAN `DUPLICATE`

command is a good way to perform recovery testing, because it requires restoring from backups and performing media recovery.

Monitor the backup procedure for errors, and validate backups by testing your recovery procedures periodically. Also, validate the ability to restore the database using the RMAN command `RESTORE...VALIDATE`.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for information about `DUPLICATE` command
- *Oracle Database Backup and Recovery User's Guide* for information about validating backups before restoring them

Back Up the RMAN and Oracle Secure Backup Catalogs on a Regular Basis

Include the recovery catalog database in your backup and recovery strategy. If you do not back up the recovery catalog and a disk failure occurs that destroys the recovery catalog database, then you may lose the metadata in the catalog. Without the recovery catalog contents, recovery of your other databases is likely to be more difficult.

The Oracle Zero Data Loss Recovery Appliance automatically backs up the RMAN Catalog to disk and then to tape along with backing up the integrated Oracle Secure Backup catalog.

When the Recovery Appliance is not used, the Oracle Secure Backup catalog maintains backup metadata, scheduling and configuration details for the backup domain. Just as it's important to protect the RMAN catalog or control file, the Oracle Secure Backup catalog should be backed up on a regular basis.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for information about managing a recovery catalog
- *Oracle Secure Backup Administrator's Guide* for information about the Oracle Secure Backup administrative server backup catalog

Use Procedures to Backup Files Outside the Database

Oracle Secure Backup provides tape backup for non-Oracle files. Oracle Secure Backup does not have pro-active checking as the database does. Use the features available to backup files outside the database. For more information, see Section , "Backup Files Outside the Database".

Backup Files Outside the Database

Oracle IT environments include both database and application files that must be protected for short and long-term retention requirements. Differences exist between backing up database and unstructured files. In addition, managing backup and recovery often crosses organizational areas such as DBA for the database and system administration for file system data could cross organizational areas. The Oracle data protection suite offers a cohesive solution meeting your complete needs for Oracle database and non Oracle database storage.

ACFS Snapshots

An Oracle ACFS snapshot is an online, read-only, point in time copy of an Oracle ACFS file system. The snapshot copy is space-efficient and uses Copy-On-Write functionality. Before an Oracle ACFS file extent is modified or deleted, its current value is copied to the snapshot to maintain the point-in-time view of the file system.

Oracle ACFS snapshot can support the online recovery of files inadvertently modified or deleted from a file system. With up to 63 snapshot views supported for each file system, flexible online file recovery solutions spanning multiple views can be employed. An Oracle ACFS snapshot can also be used as the source of a file system backup, as it can be created on demand to deliver a current, consistent, online view of an active file system.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for more information About Oracle ACFS Snapshots
- *Oracle Automatic Storage Management Administrator's Guide* for information about Managing Oracle ACFS Snapshots with Oracle Enterprise Manager

Oracle ZFS Storage Appliance Snapshots

Oracle ZFS Storage Appliance provides an integrated high performance backup solution and is also a cost effective platform for disaster recovery for non-Database files. Ever-growing amounts of data present system and database administrators with many challenges—the thorniest of which are associated with the complex process of backup and recovery. Without reliable data protection and processes, mission-critical data is at risk.

Oracle ZFS Storage Appliance is an easy-to-deploy Unified Storage System that ensures that backup window and recovery time objectives (RTO) are met by providing timely recovery in the event of a disaster.

Oracle ZFS Storage Appliance supports unlimited snapshot capability. A snapshot similar to Oracle ACFS is a read-only, point-in-time copy of a file system (for information about Oracle ACFS, see Section , "ACFS Snapshots"). It is instantaneously created and no space is allocated initially. Blocks are allocated as changes are made to the base file system (copy-on-write). The snapshots are either initiated manually or can be automated by scheduling at specific intervals. The snapshot data can be directly accessed for any backup purposes. Any reads to the snapshot blocks are served by the base file system's block. When changes happen to the base file system, the older block is now referenced by the snapshot and the new changed block is referenced by the file system.

Oracle ZFS Storage Appliance is also recommended for development and test systems that are snapshots taken from the standby database in a Data Guard environment.

Snapshot rollback is the process to bring the base file system to the point in time when the snapshot is taken. The rollback process discards all the changes that happened to the base file system from the time of the snapshot to the time of rollback. This removes the need for a data restore process.

See Also:

- https://wikis.oracle.com/display/FishWorks/Documentation for documentation on Oracle ZFS Storage Appliance
- The MAA white paper "Oracle Database Cloning Solution Using Oracle's Sun ZFS Storage Appliance And Oracle Data Guard" from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa
- The MAA white paper "Oracle Database Cloning Using Oracle Recovery Manager and Sun ZFS Storage Appliance" from the MAA Best Practices area for Oracle Databases at

  http://www.oracle.com/goto/maa

Tape Backups

Oracle Secure Backup provides centralized tape backup management for heterogeneous file system data and the Oracle database. Oracle Secure Backup offers multiple backups levels with full, cumulative and differential incrementals. For more information, see Section , "Backup to Tape Best Practices". In addition, a full offsite backup level may be scheduled without interfering with the regular full/incremental schedule. File system backups can be performed at the file, directory, file system, or raw partition level meeting even the most stringent requirements within user-defined backup windows.

For file system backup operations, you define Oracle Secure Backup "datasets" which describes what to backup. A dataset is a textual description employing a lightweight language to communicate how to build and organize files to be protected. Being Oracle Database aware, Oracle Secure Backup can skip database files during file system backups by using the "exclude oracle database files" directive within the dataset.

> **See Also:** The *Oracle Secure Backup Administrator's Guide* for more information about file system backup operations

8

Configuring Oracle Data Guard

The proper configuration of Oracle Data Guard is essential to ensuring that all standby databases work properly and perform their roles within the necessary service levels after switchovers and failovers.

The best practices for Oracle Data Guard build on the best practices described in Chapter 4, "Configuring Oracle Database."

This chapter contains the following topics:

Oracle Data Guard Configuration Best Practices

Data Guard is the Oracle optimized solution for Data availability and protection. It excels at simple, fast, and reliable one-way replication of a complete Oracle Database to provide High Availability and Disaster Recovery. Data Guard offers various deployment options that address unplanned outages, pre-production testing, and planned maintenance. Active Data Guard, an extension of basic Data Guard capabilities, further enables production offload of read-only workload to a synchronized physical standby database, automatic repair of corrupt blocks, and offload of fast incremental backups.

The focus of Data Guard is High Availability and Data Recovery. Data Guard design principles are simplicity, high performance, and application transparency.

Data Guard is not intended to be a full-featured replication solution. Oracle GoldenGate is the solution recommended for advanced replication requirements, such as multi-master replication, granular replication of a subset of a database, many to one replication topologies, and data integration. Oracle GoldenGate also provides additional options for reducing downtime for planned maintenance and for heterogeneous platform migrations.

Depending upon your requirements, the most efficient solution to use may be using Data Guard alone, using Data Guard with Oracle GoldenGate in a complementary manner, or just using Oracle GoldenGate.

For more information about Data Guard and Oracle GoldenGate see the Product Technical Brief on Oracle Active Data Guard and Oracle GoldenGate at

http://www.oracle.com/technetwork/middleware/goldengate/overview/index.html

Table 8–1 provides a summary of the Data Guard deployment options that are appropriate, depending on your requirements. Two or more options may be used in combination to address multiple requirements. This chapter also presents the Best practices for implementing each option.

Table 8–1 Requirements and Data Guard Deployment Options

Requirement	Data Guard Deployment Options
Zero data loss protection and availability for Oracle Database	Data Guard Maximum Protection or Maximum Availability (SYNC transport) and Redo Apply (physical standby). Active Data Guard Far Sync.
Near-zero data loss (single-digit seconds) and availability for Oracle Database	Data Guard Maximum Performance (ASYNC transport) and Redo Apply
Multi-site protection, including topology with local zero data loss standby for HA and remote asynchronous standby for geographic disaster recovery for Oracle Database	Multi-standby Data Guard configuration and Redo Apply
Fastest possible database failover	Data Guard Fast-Start Failover with Oracle Data Guard broker for automatic failure detection and database failover. Automatic failover of accompanying client applications to the new production database is implemented using Oracle Fast Application Notification (FAN) and Oracle Client Failover Best Practices. For more information, see the MAA white paper "Client Failover Best Practices for Data Guard 11g Release 2" from the MAA Best Practices area for Oracle Database at http://www.oracle.com/goto/maa
Offload read-only queries and fast incremental backups to a synchronized standby database. Use the standby database to automatically repair corrupt blocks, transparent to the application and user	Active Data Guard. Active Data Guard can be purchased in either of the following ways: (1) standalone as an option license for Oracle Database Enterprise Edition, or (2) included with an Oracle GoldenGate license.

Table 8–1 (Cont.) Requirements and Data Guard Deployment Options

Requirement	Data Guard Deployment Options
Pre-production testing	Snapshot Standby. A snapshot standby is a physical standby database that is temporarily open read/write for test and other read/write activity independent of primary database transactions. A snapshot standby is easily converted back into a synchronized standby database when testing is complete. Snapshot Standby is an included feature of Data Guard Redo Apply and is an ideal complement for Oracle Real Application Testing.
Planned maintenance: certain platform migrations such as Windows to Linux, data center moves, patching and upgrading system software or Oracle Database	Data Guard switchover, planned role transition, using Redo Apply. Redo Apply and Standby-First Patch Apply for qualifying patches from 11.2.0.1 onward. SQL Apply and Data Guard Database Rolling Upgrades (10.1 onward). Data Guard Transient Logical Standby (Upgrades Made Easy) from 11.1.0.7 onward. For more information, see the MAA white paper, "Database Rolling Upgrades Made Easy by Using a Data Guard Physical Standby Database", from the MAA Best Practices area for Oracle Database at http://www.oracle.com/goto/maa
Data Protection for data residing outside of the Oracle Database	When practical, move operating system file system data into Oracle Database using Oracle Database File System (DBFS). Data Guard protects DBFS data in the same manner as any other Oracle data. Data that must remain in operating system files can be protected using Oracle ASM Cluster File System (Oracle ACFS) or storage mirroring, and Data Guard.

Note: Standby-First Patch allows you to apply a patch initially to a physical standby database while the primary database remains at the previous software release (this applies for certain types of patches and does not apply for Oracle patch sets and major release upgrades; use the Data Guard transient logical standby method for patch sets and major releases). Once you are satisfied with the change, then you perform a switchover to the standby database. The fallback is to switchback if required. For more information, see "Oracle Patch Assurance - Data Guard Standby-First Patch Apply" in My Oracle Support Note 1265700.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1265700.1

See Also:

- *Oracle Database High Availability Overview* for a description of the high availability solutions and benefits provided by Oracle Data Guard and standby databases
- *Oracle Data Guard Concepts and Administration* provides complete information about Oracle Data Guard
- *Oracle Data Guard Broker* for information about the DGMGRL command-line interface

Determine Protection Mode and Data Guard Transport

Oracle Data Guard Zero Data Loss protection provides both a guarantee of data protection and the simplest recovery. For these reasons a Zero Data Loss protection mode, either Oracle Data Guard Maximum Protection or Maximum Availability, is

recommended. While both modes use Oracle Data Guard synchronous redo transport by default, there are differences in the rule-sets used to govern behavior at failover time that must be considered, as described below. Oracle Data Guard synchronous redo transport, however, can impact primary database performance if round-trip network latency between primary and standby databases is too great (latency is a function of distance and how 'clean' the network is). If this is the case (testing is easy to do, a database administrator can change protection modes and transport methods dynamically), then use Oracle Data Guard Maximum Performance. Maximum Performance uses Oracle Data Guard asynchronous transport services and does not have any impact on primary database performance regardless of network latency. In an environment with sufficient bandwidth to accommodate redo volume, data loss potential is measured in single-digit seconds when using Maximum Performance.

To determine the appropriate data protection mode for your application, consult *Oracle Data Guard Concepts and Administration*.

Best practices for the protection mode:

- **Maximum Protection mode** guarantees that no data loss will occur if the primary database fails, even in the case of multiple failures (for example, the network between the primary and standby fails, and then at a later time, the primary fails). This is enforced by never signaling commit success for a primary database transaction until at least one synchronous Data Guard standby has acknowledged that redo has been hardened to disk. Without such an acknowledgment the primary database will stall and eventually shut down rather than allow unprotected transactions to commit. To maintain availability in cases where the primary database is operational but the standby database is not, the best practice is to always have a minimum of two synchronous standby databases in a Maximum Protection configuration. Primary database availability is not impacted if it receives acknowledgment from at least one synchronous standby database.
- **Maximum Availability mode** guarantees that no data loss will occur in cases where the primary database experiences the first failure to impact the configuration. Unlike the previous protection mode, Maximum Availability will wait a maximum of `NET_TIMEOUT` seconds for an acknowledgment from a standby database, after which it will signal commit success to the application and move to the next transaction. Primary database availability (thus the name of the protection mode) is not impacted by an inability to communicate with the standby (for example, due to standby or network outages). Oracle Data Guard will continue to ping the standby and automatically re-establish connection and resynchronize the standby database when possible, but during the period when primary and standby have diverged there will be data loss should a second failure impact the primary database. For this reason, it is a best practice to monitor protection level (simple to do using Enterprise Manager Grid Control) and quickly resolve any disruption in communication between primary and standby before a second failure can occur.
- **Maximum Performance mode** (the default mode) provides the highest level of data protection that is possible without affecting the performance or the availability of the primary database. This is accomplished by allowing a transaction to commit as soon as the redo data needed to recover that transaction is written to the local online redo log at the primary database (the same behavior as if there were no standby database). Oracle Data Guard transmits redo to the standby database directly from the primary log buffer asynchronous to the local online redo log write. There is never any wait for standby acknowledgment. Similar to Maximum Availability, it is a best practice to monitor protection level (simple to do using Enterprise Manager Grid Control) and quickly resolve any

disruption in communication between primary and standby before a second failure can occur.

> **See Also:** *Oracle Data Guard Concepts and Administration* for information about Data Guard Protection Modes

Use Redo Transport Services Best Practices

At a high level, the Redo Transport best practices for planning and implementing redo transport services for Oracle Data Guard are as follows:

- Use the `SYNC` redo transport mode for a high degree of synchronization between the primary and standby databases. Use `SYNC` redo transport for zero data loss protection where performance service levels can tolerate the impact caused by network latency and standby I/O performance.
  - Use `SYNC` redo transport mode with the `NOAFFIRM` attribute (default=AFFIRM) when using Maximum Availability mode. This feature is known as FASTSYNC and helps to minimize the impact of `SYNC` redo transport by acknowledging the receipt of redo once it has been successfully received and verified within standby memory, but before the redo has been written to the standby redo log. Use `LogXptMode=FASTSYNC` in Data Guard Broker. Zero data loss protection is still preserved when only the primary database fails.
- Use the `ASYNC` redo transport mode for minimal impact on the primary database, but with a lower degree of synchronization. Use `ASYNC` redo transport when zero data loss protection is not required and sub-second or seconds of potential data loss is acceptable, or when the performance impact caused by network latency makes it impractical to use `SYNC`.
- Optimize network throughput following the best practices described in Section , "Assess Performance with Proposed Network Configuration".

Assess Performance with Proposed Network Configuration

Oracle recommends that you conduct a performance assessment with your proposed network configuration and current, or anticipated, peak redo rate. The network effect between the primary and standby databases, and the effect on the primary database throughput must be understood. Because the network between the primary and standby databases is essential for the two databases to remain synchronized, the infrastructure must have the following characteristics:

- Sufficient bandwidth to accommodate the maximum redo generation rate and any other activity sharing the same network
- If using the `SYNC` transport, then minimal latency is necessary to reduce the performance impact on the primary database
  - Use `FASTSYNC` or `SYNC NOAFFIRM` to eliminate the additional latency due to standby I/O.
- Multiple network paths for network redundancy

In configurations that use a dedicated network connection, the required bandwidth is determined by the maximum redo rate of the primary database and the efficiency of the network. Depending on the data protection mode, there are other recommended practices and performance considerations. Maximum protection mode and maximum availability mode require `SYNC` transport.

The maximum performance protection mode uses `ASYNC` redo transport. Use `ASYNC` redo transport when data loss can be tolerated or when the performance impact

caused by network latency makes it impractical to use SYNC (use SYNC redo transport for zero data loss protection).

Unlike the ASYNC transport mode, the SYNC transport mode can affect the primary database performance due to the incurred network latency. Distance and network configuration directly influence latency, while high latency can slow the potential transaction throughput and quicken response time. The network configuration, number of repeaters, the overhead of protocol conversions, network congestion, and the number of routers also affect the overall network latency and transaction response time.

Active Data Guard Far Sync

SYNC transport over WAN distances or on an underperforming network often has too large an impact on primary database performance to support zero data loss protection. Oracle 12*c* Active Data Guard Far Sync provides the ability to perform a zero data loss failover to a remote standby database without requiring a second standby database or complex operation. Far Sync enables this by deploying a Far Sync instance (a lightweight Oracle instance) at a distance that is within an acceptable range of the primary for SYNC transport. A Far Sync instance receives redo from the primary using SYNC transport and forwards the redo to up to 29 remote standby databases using ASYNC transport.

There are few new configuration best practices necessary in addition to those that would apply to any SYNC redo transport destination (see Section , "Assess Performance with Proposed Network Configuration"). They are:

- Standby Redo Logs (SRLs) should be placed on storage with sufficient IOPS (writes per second) capacity to support peak primary database redo rates in addition to any I/O activity using the same shared storage. This is an important consideration. For example:
  - If the Far Sync instance has lower performing disks than the primary it will not be able to forward redo to remote destinations as fast as it is received, and an archive log gap may form.
  - In the case of redo gap resolution scenarios, due to planned maintenance on the standby or network outages, for example, there will be additional I/O requests for gap resolution on top of peak redo coming in.
  - Lower performing disks at the Far Sync instance will delay acknowledgement to the primary database, increasing the total round-trip time between primary and standby and impacting application response time. This impact can be eliminated by using Fast Sync between the primary and the Far Sync instance.
- The Far Sync instance should have the same number of redo log groups as the primary plus one for each thread as described in standard MAA Best Practices.
- The SRLs of an alternate Far Sync instance should be manually cleared prior to use in order to achieve the best return to SYNC transport when the alternate Far Sync is acitvated. For example:

  ```
  ALTER DATABASE CLEAR LOGFILE GROUP 4, GROUP 5, GROUP 6, GROUP 7;
  ```

- Performance testing has shown that a small Far Sync instance SGA does not impact performance of the Far Sync instance nor the primary database. The MAA recommendation is to configure the minimum SGA required for Far Sync to function.
  - In order to achieve the smallest possible SGA, set CPU_COUNT=1 or 2.

  - MAA testing determined that a 300MB SGA (with CPU_COUNT=1) on Linux was sufficient for Far Sync

- When using RMAN, configure the RMAN archive log deletion policy at the Far Sync instance to `SHIPPED TO ALL STANDBY` or `APPLIED ON ALL STANDBY`. Backing up the archive logs at the Far Sync instance is not necessary provided there is a proper backup plan at the primary or standby site.
- Configure Far Sync instances for both the primary and standby databases to allow for zero data loss protection to be maintained following role transitions. The second Far Sync instance configured in proximity to the standby database will be idle until the standby becomes primary, enabling `SYNC` redo transport in the reverse direction.
  - Note that in a Data Guard Broker configuration, a switchover (planned role transition) cannot occur while in Maximum Availability mode unless the protection mode can be enforced from the target standby site. If the standby does not have its own Far Sync instance it will have to be configured to ship `ASYNC` to the original primary after roles are reversed. This will prevent a switchover from occurring unless the protection mode for the primary database is first dropped from Maximum Availability to Maximum Performance.
- Fast Sync improved performance between 5% and 12% depending on the network latency between the primary database and Far Sync instance.
- Multiple Far Sync instances servicing multiple Data Guard configurations can share the same physical server, cluster, or virtual machine.

Offloading to Far Sync

A Far Sync instance also offloads from the primary any overhead of resolving gaps in redo received by the remote standby database (for example, following network or standby database outages) and can conserve WAN bandwidth by performing redo transport compression without impacting primary database performance (Note that redo compression requires that the Advanced Compression Option be licensed).

Redo Transport Encryption can additionally be offloaded to the Far Sync instance. Including Advanced Security Option (ASO) encryption during MAA testing showed no impact to the performance of the primary nor currency of the standby databases.

Oracle recommends using ASO for encryption because it is tested and integrated with Oracle Net and Data Guard. (Note that Oracle Advanced Security Option is a licensed option).

Far Sync High Availability

In a Far Sync configuration, high availability or uninterrupted redo shipment can be achieved in multiple ways. In this sense, HA refers to maintaining data protection. An outage of a Far Sync Instance does not affect the availability of the production database. Each approach has special considerations and is described in the sections that follow.

Far Sync using Oracle Real Application Clusters - Oracle RAC The Far Sync instance can be placed on an Oracle RAC cluster. In this configuration only one instance is used at a time while other instances remain available in case of node failure. The characteristics of this approach include:

- Lowest data loss potential and brown-out when the active Far Sync instance or node fails.

- The ability to resume zero data loss protection quickly after Far Sync instance failure.
- By itself, this solution does not address cluster failure.

Far Sync HA using Alternate Destinations and Multiple Far Sync instances Configuring two separate Far Sync instances on distinct physical machines, each serving as an alternate destination for the other, provides Far Sync high availability in a non-Oracle RAC environment. Each destination defined on the primary database contains the `ALTERNATE` keyword assigning the other Far Sync instance as the alternate. When the active Far Sync instance enters an error state the alternate destination pointing to the alternate Far Sync instance is enabled automatically. By defining a Far Sync instance as an alternate destination, Maximum Availability protection will be maintained after a briefly dropping to a resynchronization state while the new destination is prepared.

The characteristics of this approach include:

- Retains zero data loss coverage after Far Sync transport failures (instance or network outages).
- Failure testing has shown
  - During Far Sync instance failures a performance brownout of approximately 3.5 seconds while `SYNC` redo transport starts (network sync service - NSS).
  - During network failures a short brownout equal to the setting of the destination's `net_timeout` parameter was observed.
- HA for machine outage assuming each Far Sync instance is on separate hardware.
- HA for site outage assuming Far Sync instances are deployed in separate sites.
- Higher application brown-out and resynchronization time during Far Sync outages compared with Far Sync with Oracle RAC.

HA Using the Terminal Standby as an Alternate Destination When an alternate Far Sync instance and Far Sync with Oracle RAC are not feasible, it is possible to create an alternate `LOG_ARCHIVE_DEST_N` pointing directly to the terminal standby (the terminal failover target). To avoid performance impact on the primary, use Data Guard asynchronous redo transport (`ASYNC`). `ASYNC` can achieve near-zero data loss protection (sub-seconds to seconds of exposure) but it is unable to provide a zero data loss guarantee. In this configuration the protection level must be dropped to Maximum Performance prior to a switchover (planned event) as the level must be enforceable on the target in order to perform the transition. Changing protection levels and transport methods is a dynamic operation that does not require downtime. The characteristics of this approach include:

- No additional hardware or Far Sync instances to manage.
- Loss of zero data loss coverage during a far sync instance outage. Data protection level drops to `UNSYNCHRONIZED` with `ASYNC` transport until the Far Sync instance can resume operation and the standby become fully synchronized.

Choosing a Far Sync Deployment Topology All configurations for Far Sync high availability perform equally with regard to receiving and sending redo. The choice of configuration should be based on application tolerance to the maximum data loss (RPO) and application brownout period of the different failure scenarios.

- An Oracle RAC Far Sync alone provides the lowest impact but requires an Oracle RAC license and has no coverage for cluster or site outage.

- Alternate Far Sync instances provide the ability to place each instance on separate physical database servers and require no additional licensing. This provides another level of protection by deploying the Far Sync instances in different sites. There is, however, slightly increased application brownout and longer resynchronization time while transport transitions from one Far Sync instance to the other.
- Terminal Standby Alternate configurations require that the application accept that there is no zero data loss protection while the Far Sync instance is not available, but requires no additional hardware to implement.

The most critical applications are well served by a pair of Oracle RAC Far Sync instances configured as alternates for each other and deployed at different locations. This provides the most robust Far Sync HA (instance, node, cluster, and site failure) protection.

Applications where data protection is critical but where cost is an important consideration are best served by deploying a pair of single node Far Sync instances, each as an alternate for the other.

Applications that can tolerate increased data loss potential during a Far Sync instance outage and where low cost is the main consideration are best served by configuring the terminal standby as an alternate location using `ASYNC` redo transport.

> **See also:** Oracle MAA white paper "Oracle Active Data Guard Far Sync Zero Data Loss at Any Distance" for details about Far Sync and `FASTSYNC`

General Data Guard Configuration Best Practices

Use the following configuration best practices for Data Guard:

- ❏ Use Oracle Data Guard Broker with Oracle Data Guard
- ❏ Use Recovery Manager to Create Standby Databases
- ❏ Use Flashback Database for Reinstatement After Failover
- ❏ Use FORCE LOGGING Mode
- ❏ Use a Simple, Robust Archiving Strategy and Configuration
- ❏ Use Standby Redo Logs and Configure Size Appropriately
- ❏ Use Data Guard Transport and Network Configuration Best Practices
- ❏ Use Data Guard Redo Apply Best Practices
- ❏ Implement Multiple Standby Databases

Use Oracle Data Guard Broker with Oracle Data Guard

Use Oracle Data Guard broker to create, manage, and monitor a Data Guard configuration. You can perform all Data Guard management operations locally or remotely through the Oracle Data Guard broker's easy-to-use interfaces: the Data Guard management pages in Oracle Enterprise Manager, which is the broker's graphical user interface (GUI), and the Data Guard command-line interface called DGMGRL.

The broker's interfaces improve usability and centralize management and monitoring of the Data Guard configuration. Available as a feature of the Enterprise Edition and

Personal Edition of the Oracle database, the broker is also integrated with the Oracle database and Oracle Enterprise Manager.

The benefits of using Oracle Data Guard broker include:

- Enhanced disaster protection.
- Higher availability and scalability with Oracle Real Application Clusters (Oracle RAC) Databases.
- Automated creation of a Data Guard configuration.
- Easy configuration of additional standby databases.
- Simplified, centralized, and extended management.
- Simplified switchover and failover operations.
- Fast Application Notification (FAN) after failovers.
- Built-in monitoring and alert and control mechanisms.
- Robust verification of Data Guard configuration using validate database command.
- Transparent to application.

> **See Also:** *Oracle Data Guard Broker* for more information about the benefits of using Data Guard Broker

Use Recovery Manager to Create Standby Databases

Oracle recommends that you use the Recovery Manager (RMAN) utility to simplify the process of creating a physical standby database.

You can either create a standby database from backups of your primary database, or create a standby database over the network:

- Use the RMAN `DUPLICATE TARGET DATABASE FOR STANDBY` command to create a standby database from backups of your primary database.

  You can use any backup copy of the primary database to create the physical standby database if the necessary archived redo log files to completely recover the database are accessible by the server session on the standby host. RMAN restores the most recent data files unless you execute the `SET UNTIL` command.

- Use the RMAN `FROM ACTIVE DATABASE` option to create the standby database over the network if a preexisting database backup is not accessible to the standby system.

  RMAN copies the data files directly from the primary database to the standby database. The primary database must be mounted or open.

You must choose between active and backup-based duplication. If you do not specify the `FROM ACTIVE DATABASE` option, then RMAN performs backup-based duplication. Creating a standby database over the network is advantageous because:

- You can transfer redo data directly to the remote host over the network without first having to go through the steps of performing a backup on the primary database. (Restoration requires multiple steps including storing the backup locally on the primary database, transferring the backup over the network, storing the backup locally on the standby database, and then restoring the backup on the standby database.)

- With active duplication you can backup a database (as it is running) from Oracle ASM, and restore the backup to a host over the network and place the files directly into Oracle ASM.

  Before this feature, restoration required you to backup the primary and copy the backup files on the primary host file system, transfer the backup files over the network, place the backup files on the standby host file system, and then restore the files into Oracle ASM.

  See Also:

  - *Oracle Data Guard Concepts and Administration* for information about using RMAN to Back Up and Restore Files
  - *Oracle Data Guard Concepts and Administration* for information about Creating a Standby Database with Recovery Manager
  - *Oracle Database Backup and Recovery User's Guide*

Use Flashback Database for Reinstatement After Failover

Enable Flashback Database on both the primary and standby database so that, in case the original primary database has not been damaged, you can reinstate the original primary database as a new standby database following a failover. If there is a failure during the switchover process, then it can easily be reversed when Flashback Database is enabled. For more information, see Section , "Enable Flashback Database".

Use FORCE LOGGING Mode

When the primary database is in `FORCE LOGGING` mode, all database data changes are logged. `FORCE LOGGING` mode ensures that the standby database remains consistent with the primary database. If this is not possible because you require the load performance with `NOLOGGING` operations, then you must ensure that the corresponding physical standby data files are subsequently synchronized. To synchronize the physical standby data files, either apply an incremental backup created from the primary database or replace the affected standby data files with a backup of the primary data files taken after the nologging operation. Before the file transfer, you must stop Redo Apply on the physical standby database.

You can enable force logging immediately by issuing an `ALTER DATABASE FORCE LOGGING` statement. If you specify `FORCE LOGGING`, then Oracle waits for all ongoing unlogged operations to finish.

See Also:

- *Oracle Database Administrator's Guide* for information about Specifying FORCE LOGGING Mode
- *Oracle Data Guard Concepts and Administration* for information about Enable Forced Logging

Use a Simple, Robust Archiving Strategy and Configuration

This archiving strategy is based on the following assumptions:

- Each database uses a fast recovery area.
- The primary database instances archive remotely to only one apply instance.

Table 8–2 describes the recommendations for a robust archiving strategy when managing a Data Guard configuration through SQL*Plus. All of the following items

are handled automatically when Oracle Data Guard broker is managing a configuration.

Table 8–2 Archiving Recommendations

Recommendation	Description

Start archiving on the primary and standby databases	Maintaining a standby database requires that you enable and start archiving on the primary database, as follows:
	`SQL> SHUTDOWN IMMEDIATE`
	`SQL> STARTUP MOUNT;`
	`SQL> ALTER DATABASE ARCHIVELOG;`
	`SQL> ALTER DATABASE OPEN;`
	Archiving must also be enabled on the standby database to support role transitions. To enable archiving on the standby database:
	`SQL> SHUTDOWN IMMEDIATE;`
	`SQL> STARTUP MOUNT;`
	`SQL> ALTER DATABASE ARCHIVELOG;`

Use a consistent log format (`LOG_ARCHIVE_FORMAT`).	The `LOG_ARCHIVE_FORMAT` parameter should specify the thread, sequence, and resetlogs ID attributes, and the parameter settings should be consistent across all instances. For example: `LOG_ARCHIVE_FORMAT=arch_%t_%S_%r.arc` **Note:** If the fast recovery area is used, then this format is ignored.
Perform remote archiving to only one standby instance and node for each Oracle RAC standby database.	All primary database instances archive to one standby destination, using the same net service name. Oracle Net Services connect-time failover is used to automatically switch to the "secondary" standby host when the "primary" standby instance has an outage. If the archives are accessible from all nodes because Oracle ASM or some other shared file system is being used for the fast recovery area, then remote archiving can be spread across the different nodes of an Oracle RAC standby database.
Specify role-based destinations with the `VALID_FOR` attribute	The `VALID_FOR` attribute enables you to configure destination attributes for both the primary and the standby database roles in one server parameter file (SPFILE), so that the Data Guard configuration operates properly after a role transition. This simplifies switchovers and failovers by removing the need to enable and disable the role-specific parameter files after a role transition.

The following example illustrates the recommended initialization parameters for a primary database communicating to a physical standby database. There are two instances, `SALES1` and `SALES2`, running in maximum protection mode.

```
*.DB_RECOVERY_FILE_DEST=+RECO
*.LOG_ARCHIVE_DEST_1='SERVICE=SALES_stby SYNC AFFIRM NET_TIMEOUT=30
    REOPEN=300 VALID_FOR=(ONLINE_LOGFILES, ALL_ROLES) DB_UNIQUE_NAME=SALES_stby'
*.LOG_ARCHIVE_DEST_STATE_1=ENABLE
```

The fast recovery area must be accessible to any node within the cluster and use a shared file system technology such as automatic storage management (Oracle ASM), a cluster file system, a global file system, or high availability network file system (HA NFS). You can also mount the file system manually to any node within the cluster very quickly. This is necessary for recovery because all archived redo log files must be accessible on all nodes.

On the standby database nodes, recovery from a different node is required when a failure occurs on the node applying redo and the apply service cannot be restarted. In that case, any of the existing standby instances residing on a different node can initiate managed recovery. In the worst case, when the standby archived redo log files are inaccessible, the managed recovery process (MRP) on the different node fetches the archived redo log files using the FAL server to retrieve from the primary node directly.

When configuring hardware vendor shared file system technology, verify the performance and availability implications. Investigate the following issues before adopting this strategy:

- Is the shared file system accessible by any node regardless of the number of node failures?
- What is the performance impact when implementing a shared file system?
- Is there any effect on the interconnect traffic?

Use Standby Redo Logs and Configure Size Appropriately

You should configure standby redo logs on all primary and standby databases for improved availability and performance.

For each redo log thread (a thread is associated with an OracleRAC database instance), number of Standby Redo Logs = number of Redo Log Groups + 1

The additional standby redo log eliminates the possibility of a standby database waiting on standby redo log. For example, if a primary database has two instances (threads) and each thread has three online log groups, then you should pre-configure 8 standby redo logs on the primary database and each standby database. Furthermore, if the primary or standby databases are not a symmetrical Real Application Cluster (example 8-node primary Oracle RAC cluster compared to 2-node standby Oracle RAC cluster), then the primary and standby databases should still have an equal number of standby redo logs and all threads should be represented.

The statements in Example 8–1 create three standby logs per thread.

Example 8–1 Create Standby Log Members

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 1G;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 1G;
```

Consider the following additional guidelines when creating standby redo logs:

- Create the same number of standby redo logs on both the primary and standby databases.
- Create all online redo logs and standby redo logs for both primary and standby databases so that they are the same size.
- Create standby redo logs in the first available ASM high redundancy disk group, or ensure that the logs are protected using external storage redundancy.
- In an Oracle RAC environment, create standby redo logs on a shared disk.
- In an Oracle RAC environment, assign a thread when the standby redo log is created as described in Example 8–1.
- Do not multiplex the standby redo logs.

To check the number and group numbers of the redo logs, query the `V$LOG` view:

```
SQL> SELECT * FROM V$LOG;
```

To check the results of the `ALTER DATABASE ADD STANDBY LOGFILE THREAD` statements, query the `V$STANDBY_LOG` view:

```
SQL> SELECT * FROM V$STANDBY_LOG;
```

> **See Also:** *Oracle Data Guard Concepts and Administration* for information about managing standby redo logs

Use Data Guard Transport and Network Configuration Best Practices

The best practices for Data Guard transport and network configuration include:

- Set the LOG_ARCHIVE_MAX_PROCESSES Parameter
- Set the Network Configuration and Highest Network Redo Rates

Set the LOG_ARCHIVE_MAX_PROCESSES Parameter

In most cases the default for `LOG_ARCHIVE_MAX_PROCESSES` is sufficient. However, in a Data Guard configurations that have multiple standby databases it may be necessary to increase the number of archive processes. The value of the `LOG_ARCHIVE_MAX_PROCESSES` initialization parameter must be at least one greater than the total number of all remote destinations. Use the following equation when setting the `LOG_ARCHIVE_MAX_PROCESSES` parameter for highly available environments:

```
LOG_ARCHIVE_MAX_PROCESSES = sum(remote_destinations) + count(threads)
```

You can adjust these parameter settings after evaluating and testing the initial settings in your production environment.

> **See Also:** *Oracle Database Administrator's Guide* for more information about Adjusting the Number of Archiver Processes

Set the Network Configuration and Highest Network Redo Rates

To set the network configuration and highest network redo rates:

- Properly Configure TCP Send / Receive Buffer Sizes
- Increase SDU Size
- Set TCP.NODELAY to YES
- Determine When to Use Redo Transport Compression

Properly Configure TCP Send / Receive Buffer Sizes

To achieve high network throughput, especially for a high-latency, high-bandwidth network, the minimum recommended setting for the sizes of the TCP send and receive socket buffers is the bandwidth-delay product (BDP) of the network link between the primary and standby systems. Settings higher than the BDP may show incremental improvement. For example, in the MAA Linux test lab, simulated high-latency, high-bandwidth networks realized small, incremental increases in throughput when using TCP send and receive socket buffer settings up to three times the BDP.

BDP is product of the network bandwidth and latency. Socket buffer sizes are set using the Oracle Net parameters `RECV_BUF_SIZE` and `SEND_BUF_SIZE`, so that the socket buffer size setting affects only Oracle TCP connections. The operating system may impose limits on the socket buffer size that must be adjusted so Oracle can use larger values. For example, on Linux, the parameters `net.core.rmem_max` and

net.core.wmem_max limit the socket buffer size and must be set larger than RECV_BUF_SIZE and SEND_BUF_SIZE.

Set the send and receive buffer sizes at either the value you calculated or 10 MB (10,485,760 bytes), whichever is larger. For example, if bandwidth is 622 Mbits and latency is 30 ms, then you would calculate the minimum size for the RECV_BUF_SIZE and SEND_BUF_SIZE parameters as follows: 622,000,000 / 8 x 0.030 = 2,332,500 bytes. Then, multiply the BDP 2,332,500 x 3 for a total of 6,997,500.

In this example, you would set the initialization parameters as follows:

```
RECV_BUF_SIZE=10485760

SEND_BUF_SIZE=10485760
```

Increase SDU Size

With Oracle Net Services it is possible to control data transfer by adjusting the size of the Oracle Net setting for the session data unit (SDU). Oracle internal testing has shown that setting the SDU to its maximum value of 65535 can improve performance for the SYNC transport. You can set SDU on a per connection basis using the SDU parameter in the local naming configuration file (TNSNAMES.ORA) and the listener configuration file (LISTENER.ORA), or you can set the SDU for all Oracle Net connections with the profile parameter DEFAULT_SDU_SIZE in the SQLNET.ORA file.

Note that the ASYNC transport uses the new streaming protocol and increasing the SDU size from the default has no performance benefit.

> **See Also:** *Oracle Database Net Services Reference* for more information about the SDU and DEFAULT_SDU_SIZE parameters

Set TCP.NODELAY to YES

To preempt delays in buffer flushing in the TCP protocol stack, disable the TCP Nagle algorithm by setting TCP.NODELAY to YES in the SQLNET.ORA file on both the primary and standby systems.

> **See Also:** *Oracle Database Net Services Reference* for more information about the TCP.NODELAY parameter

Determine When to Use Redo Transport Compression

In Oracle Database 11*g* Release 2 (11.2.0.2) redo transport compression is no longer limited to compressing redo data only when a redo gap is being resolved. When compression is enabled for a destination, all redo data sent to that destination is compressed.

In general, compression is most beneficial when used over low bandwidth networks. As the network bandwidth increases, the benefit is reduced. Compressing redo in a Data Guard environment is beneficial if:

- Sufficient CPU resources are available for the compression processing.
- The database redo rate is being throttled by a low bandwidth network.

Before enabling compression, assess the available CPU resources and decide if enabling compression is feasible. For complete information about enabling compression, see "Redo Transport Compression in a Data Guard Environment" in My Oracle Support Note 729551.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=729551.1

Use Data Guard Redo Apply Best Practices

To improve the Redo Apply rate of a physical standby database (and media recovery):

- Maximize I/O Rates on Standby Redo Logs and Archived Redo Logs
- Assess Recovery Rate
- Set DB_BLOCK_CHECKSUM=FULL and DB_BLOCK_CHECKING=MEDIUM or FULL
- Set DB_CACHE_SIZE to a Value Greater than on the Primary Database
- Assess Database Wait Events
- Tune I/O Operations
- Assess System Resources

See Also:

The MAA white paper "Active Data Guard 11g Best Practices (includes best practices for Redo Apply)" from the MAA Best Practices area for Oracle Database at

http://www.oracle.com/goto/maa

Maximize I/O Rates on Standby Redo Logs and Archived Redo Logs

Measure read I/O rates on the standby redo logs and archived redo log directories. Concurrent writing of shipped redo on a standby database might reduce the redo read rate due to I/O saturation. The overall recovery rate is always bounded by the rate at which redo can be read; so ensure that the redo read rate surpasses your required recovery rate.

Assess Recovery Rate

To obtain the history of recovery rates, use the following query to get a history of recovery progress:

```
SELECT * FROM V$RECOVERY_PROGRESS;
```

If your ACTIVE APPLY RATE is greater than the maximum redo generation rate at the primary database or twice the average generation rate at the primary database, then no tuning is required; otherwise follow the tuning tips below. The redo generation rate for the primary database can be monitored from Enterprise Manager or extracted from AWR reports under statistic REDO SIZE. If CHECKPOINT TIME PER LOG is greater than ten seconds, then investigate tuning I/O and checkpoints.

Set DB_BLOCK_CHECKSUM=FULL and DB_BLOCK_CHECKING=MEDIUM or FULL

Redo apply performance should be fast enough to keep up with most applications' redo generation rates but you can temporarily disable DB_BLOCK_CHECKING to speed up recovery. If you disable DB_BLOCK_CHECKING, you will disable in-memory block semantic checks as described in My Oracle Support note 1302539.1.

Note: To check for block corruption that was not preventable through the DB_BLOCK_CHECKING parameter, use:

- RMAN BACKUP command with the VALIDATE option
- DBVERIFY utility
- ANALYZE TABLE *tablename* VALIDATE STRUCTURE CASCADE SQL statement

Set the DB_LOST_WRITE_PROTECT parameter to FULL on the standby database to enable Oracle to detect writes that are lost in the I/O subsystem. The impact on redo apply is very small for OLTP applications and generally less than 5 percent.

See Also: Section , "Preventing Widespread Data Corruption"

Set DB_CACHE_SIZE to a Value Greater than on the Primary Database

Set DB_CACHE_SIZE to a value greater than that for the primary database. Set DB_KEEP_CACHE_SIZE and DB_RECYCLE_CACHE_SIZE to 0.

Having a large database cache size can improve media recovery performance by reducing the amount of physical data block reads. Because media recovery does not require DB_KEEP_CACHE_SIZE and DB_RECYCLE_CACHE_SIZE or require a large SHARED_POOL_SIZE, the memory can be reallocated to the DB_CACHE_SIZE.

Before converting the standby database into a primary database, reset these parameters to the primary database settings.

Assess Database Wait Events

With the Active Data Guard option and real-time query, you can use Statspack from the primary database to collect data from a standby database that is opened read-only and performing recovery. Any tuning or troubleshooting exercise should start with collecting Standby Statspack reports. For complete details about installing and using Standby Statspack, see "Installing and Using Standby Statspack in 11g" in My Oracle Support Note 454848.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=454848.1

If you do not have a license for the Active Data Guard option, you can determine the top system and session wait events by querying the standby database's V$SYSTEM_EVENT, V$SESSION_WAIT, and V$EVENT_HISTOGRAM and looking for the largest TIME_WAITED value. You may have to capture multiple snapshots of the query results and manually extract the difference to accurately assess a certain time period.

If recovery is applying a lot of redo data efficiently, the system is I/O bound and the I/O wait should be reasonable for your system. The vast majority of wait events related to parallel recovery coordinators and slaves apply to the coordinator. Slaves are either applying changes (clocking on CPU) or waiting for changes to be passed from the coordinator.

Typically, in a properly tuned system, the top wait event is db file parallel write followed by checkpoint completed. Consult the table below for tuning advice in cases where db file parallel write is not the top wait event. The database wait events are shown in Table 8–3 and Table 8–4.

Table 8–3 Parallel Recovery Coordinator Wait Events

Wait Name	Description	Tuning
`Log file sequential read`	The parallel recovery coordinator is waiting on I/O from the online redo log or the archived redo log.	Tune or increase the I/O bandwidth for the ASM diskgroup where the archive logs or online redo logs reside.
`Parallel recovery read buffer free`	This event indicates that all read buffers are being used by slaves, and usually indicates that the recovery slaves lag behind the coordinator.	Tune or increase the I/O bandwidth for the ASM diskgroup where data files reside.
`Parallel recovery change buffer free`	The parallel recovery coordinator is waiting for a buffer to be released by a recovery slave. Again, this is a sign the recovery slaves are behind the coordinator.	Tune or increase the I/O bandwidth for the ASM diskgroup where data files reside.
`Datafile init write`	The parallel recovery coordinator is waiting for a file resize to finish, as would occur with file auto extend.	Tune or increase the I/O bandwidth for the ASM diskgroup where data files reside.
`Parallel recovery control message reply`	The coordinator has sent a synchronous control messages to all slaves, and is waiting for all slaves to reply.	This is a non-tunable event.

When dealing with recovery slave events, it is important to know how many slaves were started. Divide the wait time for any recovery slave event by the number of slaves. Table 8–4 describes the parallel recovery slave wait events.

Table 8–4 Parallel Recovery Slave Wait Events

Wait Name	Description	Tuning
`Parallel recovery slave next change`	The parallel recovery slave is waiting for a change to be shipped from the coordinator. This is in essence an idle event for the recovery slave. To determine the amount of CPU a recovery slave is using, divide the time spent in this event by the number of slaves started and subtract that value from the total elapsed time. This may be close, because there are some waits involved.	Tune or increase the I/O bandwidth for the ASM diskgroup where the archive logs or online redo logs reside.
`DB File Sequential Read`	A parallel recovery slave (or serial recovery process) is waiting for a batch of synchronous data block reads to complete.	Tune or increase the I/O bandwidth for the ASM diskgroup where data files reside.

Table 8–4 (Cont.) Parallel Recovery Slave Wait Events

Wait Name	Description	Tuning
`Checkpoint completed`	Recovery is waiting for checkpointing to complete, and Redo Apply is not applying any changes currently.	Tune or increase the I/O bandwidth for the ASM diskgroup where data files reside. Also, increase the number of `db_writer_processes` until the `checkpoint completed` wait event is lower than the `db file parallel write` wait event. Consider also increasing the online log file size on the primary and standby to decrease the number of full checkpoints at log switch boundaries.
`Recovery read`	A parallel recovery slave is waiting for a batched data block I/O.	Tune or increase the I/O bandwidth for the ASM diskgroup where data files reside.

Tune I/O Operations

DBWR must write out modified blocks from the buffer cache to the data files. Always use native asynchronous I/O by setting `DISK_ASYNCH_IO` to `TRUE` (default). In the rare case that asynchronous I/O is not available, use `DBWR_IO_SLAVES` to improve the effective data block write rate with synchronous I/O.

Ensure that you have sufficient I/O bandwidth and that I/O response time is reasonable for your system either by doing some base I/O tests, comparing the I/O statistics with those for the primary database, or by looking at some historical I/O metrics. Be aware that I/O response time may vary when many applications share the same storage infrastructure such as with a Storage Area Network (SAN) or Network Attached Storage (NAS).

Assess System Resources

Use system commands such as UNIX `sar` and `vmstat` commands, or use system monitoring tools to assess the system resources. Alternatively, you can monitor using Oracle Enterprise Manager, AWR reports, or performance views such as `V$SYSTEM_EVENT`, `V$ASM_DISK` and `V$OSSTAT`.

1. If there are I/O bottlenecks or excessive wait I/O operations, then investigate operational or application changes that increased the I/O volume. If the high waits are due to insufficient I/O bandwidth, then add more disks to the relevant Oracle ASM disk group. Verify that this is not a bus or controller bottleneck or any other I/O bottleneck. The read I/O rate from the standby redo log should be greater than the expected recovery rate.
2. Check for excessive swapping or memory paging.
3. Check to ensure the recovery coordinator or MRP is not CPU bound during recovery.

Implement Multiple Standby Databases

You should deploy multiple standby databases for any of the following purposes. When desired, use standby databases for these purposes while reserving at least one standby database to serve as the primary failover target:

- To provide continuous protection following failover

  The standby databases in a multiple standby configuration that are not the target of the role transition (these databases are referred to as *bystander standby databases*) automatically apply redo data received from the new primary database.

- To achieve zero data loss protection while also guarding against widespread geographic disasters that extend beyond the limits of synchronous communication

  For example, one standby database that receives redo data synchronously is located 200 miles away, and a second standby database that receives redo data asynchronously is located 1,500 miles away from the primary.

- To perform rolling database upgrades while maintaining disaster protection throughout the rolling upgrade process

- To perform testing and other ad-hoc tasks while maintaining disaster-recovery protection

Use Multiple Standby Databases Best Practices

The *Oracle Database High Availability Overview* describes how a multiple standby database architecture is virtually identical to that of single standby database architectures. Therefore, the configuration guidelines for implementing multiple standby databases described in this section complement the existing best practices for physical and logical standby databases.

When deploying multiple standby databases, use the following best practices:

- Use Oracle Data Guard broker to manage your configuration and perform role transitions. However, if you choose to use SQL*Plus statements, see the MAA white paper "Multiple Standby Databases Best Practices" for best practices from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa

- If you are using Flashback Database for the sole purpose of reinstating databases following a failover, a `DB_FLASHBACK_RETENTION_TARGET` of 120 minutes is the minimum recommended value. When you use Flashback Database to quickly reinstate the original primary as the standby after a failover, instead of re-creating the entire standby database from backups or from the primary database, when using Fast-start Failover, ensure the `UNDO_RETENTION` and `DB_FLASHBACK_RETENTION_TARGET` initialization parameters are set to a minimum of 120 so that reinstatement is still possible after a prolonged outage. On a standby the flashback barrier cannot be guaranteed to be published every 30 minutes as it is on a primary. Thus, when enabling flashback database on a standby, the `DB_FLASHBACK_RETENTION_TARGET` should be a minimum of 120. Since the primary and standby should match, this implies the same for the primary.

- Enable supplemental logging in configurations containing logical standby databases. When creating a configuration with both physical and logical standby databases, issue the `ALTER DATABASE ADD SUPPLEMENTAL LOG DATA` statement to enable supplemental logging in the following situations:

  - When adding a logical standby database to an existing configuration consisting of all physical standby databases, you must enable supplemental logging on all existing physical standby databases in the configuration.
  - When adding a physical standby database to an existing configuration that contains a logical standby database, you must enable supplemental logging on the physical standby database when you create it.

  As part of the logical standby database creation supplemental logging is automatically enabled on the primary. Enabling supplemental logging is a control file change and therefore the change is not propagated to each physical standby database. Supplemental logging is enabled automatically on a logical standby database when it is first converted from a physical standby database to a logical standby database as part of the dictionary build process.

  To enable supplemental logging, issue the following SQL*Plus statement when connected to a physical standby database:

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE INDEX)
COLUMNS;
```

- If logical standby databases are not configured to perform real-time queries, then consider configuring SQL Apply to delay applying redo data to the logical standby database. By delaying the application of redo, you can minimize the need to manually reinstate the logical standby database after failing over to a physical standby database.

  To set a time delay, use the `DELAY=minutes` attribute of the `LOG_ARCHIVE_DEST_`*n* initialization parameter.

  See Also:

  - *Oracle Database High Availability Overview* to learn about the benefits of using multiple standby database and for implementation examples
  - *Oracle Database High Availability Overview* for an overview of multiple standby database architectures
  - The MAA white paper "Multiple Standby Databases Best Practices" from the MAA Best Practices area for Oracle Database at

    http://www.oracle.com/goto/maa

Oracle Multitenant Databases in a Data Guard Environment

All of the functionality of Data Guard is available when using Oracle Multitenant. All role transitions, Fast Start Failover, Transient Logical Rolling Upgrade, Oracle Active Data Guard, and so on, can be used with container databases (CDBs). Note that role transitions occur at the CDB level, all pluggable databases (PDBs) will transition over to the new primary database.

PDBs can be created in a variety of ways.

- Unplug a PDB from one CDB and plug in to a second
- Create as an empty PDB from the seed PDB
- Clone an existing PDB, either a PDB in the same container (local) or from another CDB (remote)

- Plug in an existing non-container database (non-CDB)

If a PDB is created from SEED, the standby database can copy its local files for the source PDB to create the new PDB. In addition if the standby database is running the Oracle Active Data Guard option, clones from local PDBs can also be created automatically on the standby database.

Prior to Oracle Database 12c version 12.1.0.2, when a PDB is created from a remote source either via plugin or remote clone operation, the files must be made available to the standby by some outside method. If the files are not available at the time the plugin redo is applied at the standby, redo apply will stop and cannot be restarted until the files are successfully added to the database controlfile.

In 12.1.0.2, the `CREATE PLUGGABLE DATABASE` statement has a new clause, `STANDBYS=NONE`, that allows for deferral of file instantiation on the standby allowing the physical standby database to continue to protect existing PDBs. The clause allows the general structure of the PDB to be created on all physical standbys but all files belonging to the PDB are marked as OFFLINE/RECOVER at the standby. The PDB cannot be opened on the standby with the files in this state although all other PDBs at the standby are unaffected.

At some point in the future, it is possible to copy the files to the standby database and enable recovery of the PDB and thus begin Data Guard protection of the PDB. Oracle provides tools to copy the files from the primary database to the standby database while the PDB is open and accessible. Enabling recovery of the PDB requires a bounce of the standby database into MOUNT mode and a brief stoppage of redo apply.

The reasons for requiring deferral include but are not limited to:

- Remote clone of a PDB. It is not possible to pre-copy the files to the physical standby database and ensure they will be in the correct state when the `CREATE PLUGGABLE DATABASE` statement redo is applied to the standby.
- The PDB is considered to be a test or short-lived PDB that will be dropped relatively quickly and thus does not need to be protected by Data Guard.
- Timing of the PDB creation does not allow for pre-instantiation of the files at the standby since that will prolong the application downtime associated with that PDB, but recovery will be required after the PDB has been created.
- Storage for the PDB on the standby environment is not immediately available.
- In some cases, the newly added PDB does not require higher level of data protection that comes with having a physical standby database and can be permanently disabled. The MAA team does not recommend this "subset standby" architecture where some PDBs in the same CDB have different HA SLAs.

It is also possible to disable recovery for a previously instantiated PDB. This requires a brief stoppage of redo apply. Potential reasons for removing a PDB from Data Guard protection include but are not limited to:

- Debugging operations of a particular PDB where redo being applied at the standby causes redo apply to fail, thus leaving the entire container database unprotected.
- Application activity against a single PDB causes redo apply on the standby to lag beyond SLA requirements.

MAA best practices recommend that you not run in this unprotected mode for an extended period of time. The implication of having some PDBs protected by a physical standby and others unprotected complicates the architecture and overall MAA solution. Refer to MAA's reference architectures for recommended architectures per

SLAs. You should review the requirements of the PDBs created with `STANDBYS=NONE` carefully and consolidate them with PDBs with similar requirements.

Outages cannot always be predicted. Prior to Oracle 12.1.0.2, Oracle does not have complete Data Guard role transition support for a configuration where files are missing on either the primary or standby databases, a concept called "subset standby." The Data Guard broker has been enhanced to report the files missing but not identify this as a reason to prevent a Data Guard role transition. If an issue arises prior to your opportunity to instantiate the files, Data Guard role transition will still work seamlessly to your standby site. The location of the files and the role of the respective database will determine PDB access. If a database in the primary role does not have the files for the PDB, it will not be able to open it, nor provide any access. If a database is in the standby role, the PDB can be opened accessed read only using the Active Data Guard option.

Oracle Data Guard Role Transition Best Practices

With proper planning and execution, Data Guard role transitions can effectively minimize downtime and ensure that the database environment is restored with minimal impact on the business. Using a physical standby database, MAA testing has determined that switchover and failover times with Oracle Data Guard 11*g* have been reduced to seconds. This section describes best practices for both switchover and failover.

Oracle Data Guard Switchovers Best Practices

A database switchover performed by Oracle Data Guard is a planned transition that includes a series of steps to switch roles between a standby database and a primary database. Following a successful switchover operation, the standby database assumes the primary role and the primary database becomes a standby database. Switchovers are typically completed in only seconds to minutes. At times the term *switchback* is also used within the scope of database role management. A switchback operation is a subsequent switchover operation to return the roles to their original state. While following best practices, switchover times of approximately 34 seconds for Oracle RAC and 19 seconds for a single instance database have been observed.

Data Guard enables you to change these roles dynamically by:

- Using Oracle Enterprise Manager
- Using the Oracle Data Guard broker's DGMGRL command-line interface
- Issuing SQL statements, as described in Section , "How to Perform Data Guard Switchover"

> **See Also:** *Oracle Data Guard Broker* for information about using Oracle Enterprise Manager or Oracle Data Guard broker's DGMGRL command-line interface to perform database switchover

To optimize switchover processing, perform the following steps before performing a switchover:

- Disconnect all sessions possible using the `ALTER SYSTEM KILL SESSION` SQL*Plus command.
- Stop job processing by setting the `AQ_TM_PROCESSES` parameter to `0`.
- Cancel any specified apply delay by using the `NODELAY` keyword to stop and restart log apply services on the standby database.

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE NODELAY DISCONNECT;
```

You can view the current delay setting on the primary database by querying the DELAY_MINS column of the V$ARCHIVE_DEST view.

- For physical standby databases in an Oracle RAC environment, ensure there is only one instance active for each primary and standby database.
- Configure the standby database to use real-time apply and, if possible, ensure the databases are synchronized before the switchover operation to optimize switchover processing.

  For the fastest switchover, use real-time apply so that redo data is applied to the standby database as soon as it is received, and the standby database is synchronized with the primary database before the switchover operation to minimize switchover time. To enable real-time apply use the following SQL*Plus statement:

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

- For a physical standby database, reduce the number of archiver (ARC*n*) processes to the minimum needed for both remote and local archiving. Additional archiver processes can take additional time to shut down, thereby increasing the overall time it takes to perform a switchover. After the switchover has completed you can reenable the additional archiver processes.
- Set the LOG_FILE_NAME_CONVERT initialization parameter to any valid value for the environment, or if it is not needed set the parameter to null.

  As part of a switchover, the standby database must clear the online redo log files on the standby database before opening as a primary database. The time needed to complete the I/O can significantly increase the overall switchover time. By setting the LOG_FILE_NAME_CONVERT parameter, the standby database can pre-create the online redo logs the first time the MRP process is started. You can also pre-create empty online redo logs by issuing the SQL*Plus ALTER DATABASE CLEAR LOGFILE statement on the standby database.

  See Also: Support notes for switchover best practices for Data Guard Physical Standby (11.2.0.2):

  - If using SQL*Plus, see "11.2 Data Guard Physical Standby Switchover Best Practices using SQL*Plus" in My Oracle Support Note 1304939.1 at

    https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1304939.1

  - If using the Oracle Data Guard broker or Oracle Enterprise Manager, see "11.2 Data Guard Physical Standby Switchover Best Practices using the Broker" in My Oracle Support Note 1305019.1 at

    https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1305019.1

  - The MAA white paper "Switchover and Failover Best Practices" from the MAA Best Practices area for Oracle Database at

    http://www.oracle.com/goto/maa

Oracle Data Guard Failovers Best Practices

A failover is typically used only when the primary database becomes unavailable, and there is no possibility of restoring it to service within a reasonable period. During a failover the primary database is taken offline at one site and a standby database is brought online as the primary database.

With Data Guard the process of failover can be completely automated using fast-start failover or it can be a manual, user driven process. Oracle recommends using fast-start failover to eliminate the uncertainty inherent in a process that requires manual intervention. Fast-start failover automatically executes a failover within seconds of an outage being detected. While following best practices, failover times of approximately 16 seconds for Oracle RAC and 9 seconds for a single instance database have been observed.

For more on Data Guard failover best practices, see:

- Comparing Fast-Start Failover and Manual Failover
- Failover Best Practices (Manual Failover and Fast-Start Failover)
- Fast-Start Failover Best Practices
- Manual Failover Best Practices

See Also: For a comprehensive review of Oracle Data Guard failover best practices, see:

- *Oracle Data Guard Broker* for information about Switchover and Failover Operations
- "Data Guard Fast-Start Failover" MAA white paper from the MAA Best Practices area for Oracle Database at

  `http://www.oracle.com/goto/maa`
- "11.2 Data Guard Physical Standby Switchover Best Practices using SQL*Plus" in My Oracle Support Note 1304939.1 at

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1304939.1`
- "11.2 Data Guard Physical Standby Switchover Best Practices using the Broker" in My Oracle Support Note 1305019.1 at

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1305019.1`

Comparing Fast-Start Failover and Manual Failover

There are two distinct types of failover: manual failover and fast-start failover. An administrator initiates manual failover when the primary database fails. In contrast, Data Guard automatically initiates a fast-start failover without human intervention after the primary database has been unavailable for a set period (the fast-start failover threshold).

Table 8–5 compares fast-start failover and manual failover.

Table 8–5 Comparing Fast-Start Failover and Manual Failover

Points of Comparison	Fast-Start Failover	Manual Failover
Benefits	Allows you to increase availability with less need for manual intervention, thereby reducing management costs.	Gives you control over exactly when a failover occurs and to which target standby database.
Failover triggers	The following conditions automatically trigger a fast-start failover: ■ Database instance failure (or last instance failure in an Oracle RAC configuration). ■ Shutdown abort (or a shutdown abort of the last instance in an Oracle RAC configuration). ■ Specific conditions that are detected through the database health-check mechanism (for example, data files taken offline due to I/O errors). Fast-start failover can be enabled for these conditions (`ENABLE FAST_START FAILOVER CONDITION`) and `ORA` errors raised by the Oracle server when they occur. See *Oracle Data Guard Broker* for a full list of conditions. ■ Both the observer and the standby database lose their network connection to the primary database. ■ Application initiated fast-start failover using the `DBMS_DG.INITIATE_FS_FAILOVER` PL/SQL procedure.	A manual failover is user initiated and involves performing a series of steps to convert a standby database into a primary database. A manual failover should be performed due to an unplanned outage such as: ■ Site disaster which results in the primary database becoming unavailable (all instances of an Oracle RAC primary database). ■ User errors that cannot be repaired in a timely fashion. ■ Data failures, which impact the production application.
Management	Use the following tools to manage fast-start failover failovers: ■ Oracle Enterprise Manager ■ The Oracle Data Guard broker command-line interface (DGMGRL) See Section , "How to Perform Data Guard Switchover".	Use the following tools to perform manual failovers: ■ Oracle Enterprise Manager ■ The Oracle Data Guard broker command-line interface (DGMGRL) ■ SQL statements See Section , "Best Practices for Performing Manual Failover".
Restoring the original primary database after failover	Following a fast-start failover, Oracle Data Guard broker can automatically reconfigure the original primary database as a standby database upon reconnection to the configuration (`FastStartFailoverAutoReinstate`), or you can delay the reconfiguration to allow diagnostics on the failed primary. Automatic reconfiguration enables Data Guard to restore disaster protection in the configuration quickly and easily, returning the database to a protected state as soon as possible.	After manual failover, you must reinstate the original primary database as a standby database to restore fault tolerance.
Restoring bystander standby databases after failover	Oracle Data Guard broker coordinates the role transition on all databases in the configuration. Bystanders that do no require reinstatement are available as viable standby databases to the new primary. Bystanders that require reinstatement are automatically reinstated by the observer.	A benefit of using Oracle Data Guard broker is that it provides the status of bystander databases and indicates whether a database must be reinstated. Status information is not readily available when using SQL*Plus statements to manage failover. See Section , "Restoring a Standby Database After a Failover".
Application failover	Oracle Data Guard broker automatically publishes FAN/AQ (Advanced Queuing) and FAN/ONS (Oracle Notification Service) notifications after a failover. Clients that are also configured for Fast Connection Failover can use these notifications to connect to the new primary database. You can also use the `DB_ROLE_CHANGE` system event to help user applications locate services on the primary database. (These events are also available for manual failovers performed by the broker. See *Oracle Data Guard Broker*.)	Oracle Data Guard broker automatically publishes FAN/AQ (Advanced Queuing) and FAN/ONS (Oracle Notification Service) notifications after a failover. Clients that are also configured for Fast Connection Failover can use these notifications to connect to the new primary database. You can also use the DB_ROLE_CHANGE system event to help user applications locate services on the primary database. (These events are also available for fast-start failovers performed by the broker. See *Oracle Data Guard Broker*.)

Failover Best Practices (Manual Failover and Fast-Start Failover)

To optimize failover processing:

- Enable Flashback Database to reinstate the failed primary databases after a failover operation has completed. Flashback Database facilitates fast point-in-time recovery, if needed.
- Use real-time apply with Flashback Database to apply redo data to the standby database as soon as it is received, and to quickly rewind the database should user error or logical corruption be detected.
- Consider configuring multiple standby databases to maintain data protection following a failover.
- Set the `LOG_FILE_NAME_CONVERT` parameter. As part of a failover, the standby database must clear its online redo logs before opening as the primary database. The time needed to complete this I/O can add significantly to the overall failover time. By setting the `LOG_FILE_NAME_CONVERT` parameter, the standby pre-creates the online redo logs the first time the MRP process is started. You can also pre-create empty online redo logs by issuing the SQL*Plus `ALTER DATABASE CLEAR LOGFILE` statement on the standby database.
- Use fast-start failover. If possible, ensure that the databases are synchronized before the switchover operation to optimize switchover processing. Real-time apply ensures that redo is applied as received and ensures the fastest switchover. Real-time apply is now the default in Oracle Database 12*c*, if standby redo logs are configured on the standby. The `USING CURRENT LOGFILE` clause is no longer required. For more information, see Section , "Fast-Start Failover Best Practices".
- For physical standby databases, do the following:
  - When transitioning from read-only mode to Redo Apply (recovery) mode, restart the database.
  - Go directly to the `OPEN` state from the `MOUNTED` state instead of restarting the standby database (as required in releases before Oracle Database 11*g* release 2).
  - See the MAA white paper "Oracle Data Guard Redo Apply and Media Recovery" to optimize media recovery for Redo Apply from the MAA Best Practices area for Oracle Database at

    `http://www.oracle.com/goto/maa`

Fast-Start Failover Best Practices

Fast-start failover automatically, quickly, and reliably fails over to a designated standby database if the primary database fails, without requiring manual intervention to execute the failover. You can use fast-start failover only in an Oracle Data Guard configuration that is managed by Oracle Data Guard broker.

The Oracle Data Guard configuration can be running in either the maximum availability or maximum performance mode with fast-start failover. When fast-start failover is enabled, the broker ensures fast-start failover is possible only when the configured data loss guarantee can be upheld. Maximum availability mode provides an automatic failover environment guaranteed to lose no data. Maximum performance mode provides an automatic failover environment guaranteed to lose no more than the amount of data (in seconds) specified by the `FastStartFailoverLagLimit` configuration property.

Use the following fast-start failover best practices in addition to the generic best practices listed in the Section , "Failover Best Practices (Manual Failover and Fast-Start Failover)":

- Run the fast-start failover observer process on a host that is not located in the same data center as the primary or standby database.

  Ideally, you should run the observer on a system that is equally distant from the primary and standby databases. The observer should connect to the primary and standby databases using the same network as any end-user client. If the designated observer fails, Oracle Enterprise Manager can detect it and automatically restart the observer. If the observer cannot run at a third site, then you should install the observer on the same network as the application. If a third, independent location is not available, then locate the observer in the standby data center on a separate host and isolate the observer as much as possible from failures affecting the standby database.

- Make the observer highly available by using Oracle Enterprise Manager to automatically restart the observer on the same host upon observer process death or fail over the observer to a designated alternate host when the primary observer host fails.

- After the failover completes, the original primary database is automatically reinstated as a standby database when a connection to it is reestablished, if you set the `FastStartFailoverAutoReinstate` configuration property to `TRUE`.

- Set the value of the `FastStartFailoverThreshold` property according to your configuration characteristics, as described in Table 8–6.

Table 8–6 Minimum Recommended Settings for FastStartFailoverThreshold

Configuration	Minimum Recommended Setting
Single-instance primary, low latency, and a reliable network	15 seconds
Single-instance primary and a high latency network over WAN	30 seconds
Oracle RAC primary	Oracle RAC miscount + reconfiguration time + 30 seconds

Test your configuration using the settings shown in Table 8–6 to ensure that the fast-start failover threshold is not so aggressive that it induces false failovers, or so high it does not meet your failover requirements.

Manual Failover Best Practices

You should perform a manual failover, which is user-driven, only in case of an emergency. The failover should be initiated due to an unplanned outage such as:

- Site disaster that results in the primary database becoming unavailable
- User errors that cannot be repaired in a timely fashion
- Data failures, to include widespread corruption, which affects the production application

Use the following manual failover best practices in addition to the generic best practices listed in Section , "Failover Best Practices (Manual Failover and Fast-Start Failover)":

- Reinstate the original primary database as a standby database to restore fault tolerance to your environment. The standby database can be quickly reinstated by

using Flashback Database. See Section , "Restoring a Standby Database After a Failover."

> **See Also:** For physical standby databases see the MAA white paper "Oracle Data Guard Redo Apply and Media Recovery" from the MAA Best Practices area for Oracle Database at
>
> http://www.oracle.com/goto/maa

Use Oracle Active Data Guard Best Practices

If you have a license for the Oracle Active Data Guard option then you can open a physical standby database for read-only access while Redo Apply on the standby database continues to apply redo data received from the primary database. All queries reading from the physical standby database execute in real time and return current results, providing more efficient use of system resources and additional assurance that the standby is healthy without compromising data protection or extending recovery time if a failover is required. Hence, this capability is referred to as **real-time query**.

> **Note:** A physical standby database can be open for read-only access while Redo Apply is active if a license for the Oracle Active Data Guard option has been purchased. This capability, known as real-time query also provides the ability to have block-change tracking on the standby database, thus allowing incremental backups to be performed on the standby.

To deploy real-time query:

- Ensure Active Data Guard is enabled.

  The easiest and best way to view the status of Oracle Active Data Guard is on the Data Guard overview page through Oracle Enterprise Manager.

  Alternatively, query the `v$database` view on the standby database and confirm the status of '`READ ONLY WITH APPLY`':

```
SQL> SELECT open_mode FROM V$DATABASE;
OPEN_MODE
--------------------
READ ONLY WITH APPLY
```

- Use real-time apply on the standby database so that changes are applied as soon as the redo data is received. Real-time apply is the default as of Oracle Database 12*c* provided standby redo logs are configured.
- Enable Flashback Database on the standby database to minimize downtime for logical corruptions.
- Monitor standby performance by using Standby Statspack. For complete details about installing and using Standby Statspack, see "Installing and Using Standby Statspack in 11g" in My Oracle Support Note 454848.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=454848.1
- When you deploy real-time query to offload queries from a primary database to a physical standby database, monitor the apply lag to ensure that it is within

acceptable limits. See *Oracle Data Guard Concepts and Administration* for information about Monitoring Apply Lag in a Real-time Query Environment.

- Create an Oracle Data Guard broker configuration to simplify management and to enable automatic apply instance failover on an Oracle RAC standby database.

> **See Also:** The "Active Data Guard 11g Best Practices (includes best practices for Redo Apply)" white paper available from the MAA Best Practices area for Oracle Database at
>
> http://www.oracle.com/goto/maa

Use Snapshot Standby Database Best Practices

Beginning with Oracle Database release 11*g*, you can convert a physical standby database into a fully updatable standby database called a snapshot standby database.

To convert a physical standby database into a snapshot standby database, issue the SQL*Plus `ALTER DATABASE CONVERT TO SNAPSHOT STANDBY` statement. This command causes Oracle Data Guard to perform the following actions:

1. Recover all available redo data
2. Create a guaranteed restore point
3. Activate the standby database as a primary database
4. Open the database as a snapshot standby database

To convert the snapshot standby back to a physical standby, issue the `ALTER DATABASE CONVERT TO PHYSICAL STANDBY` statement. This command causes the physical standby database to be flashed back to the guaranteed restore point that was created before the `ALTER DATABASE CONVERT TO SNAPSHOT STANDBY` statement was issued. Then, you must perform the following actions:

1. Restart the physical standby database
2. Restart Redo Apply on the physical standby database

To create and manage snapshot standby databases:

- Use the Oracle Data Guard broker to manage your Oracle Data Guard configuration, because it simplifies the management of snapshot standby databases. The broker will automatically convert a snapshot standby database into a physical standby database as part of a failover operation. Without the broker, this conversion must be manually performed before initiating a failover.
- Create multiple standby databases if your business requires a fast recovery time objective (RTO).
- Ensure the physical standby database that you convert to a snapshot standby is caught up with the primary database, or has a minimal apply lag. See Section , "Use Data Guard Redo Apply Best Practices" for information about tuning media recovery.
- Configure a fast recovery area and ensure there is sufficient I/O bandwidth available. This is necessary because snapshot standby databases use guaranteed restore points.

> **See Also:** *Oracle Data Guard Concepts and Administration* for complete information about creating a snapshot standby database

Assessing Data Guard Performance

To accurately assess the primary database performance after adding Data Guard standby databases, obtain a history of statistics from the V$SYSMETRIC_SUMMARY view or Automatic Workload Repository (AWR) snapshots before and after deploying Oracle Data Guard with the same application profile and load.

To assess the application profile, compare the following statistics:

- Physical reads per transaction
- Physical writes per transaction
- CPU usage per transaction
- Redo generated per transaction

To assess the application performance, compare the following statistics:

- Redo generated per second or redo rate
- User commits per second or transactions per second
- Database time per second
- Response time per transaction
- SQL service response time

If the application profile has changed between the two scenarios, then this is not a fair comparison. Repeat the test or tune the database or system with the general principles outlined in the *Oracle Database Performance Tuning Guide*.

If the application profile is similar and you observe application performance changes on the primary database because of a decrease in throughput or an increase in response time, then assess these common problem areas:

- CPU utilization

  If you are experiencing high load (excessive CPU usage of over 90%, paging and swapping), then tune the system before proceeding with Data Guard. Use the V$OSSTAT view or the V$SYSMETRIC_HISTORY view to monitor system usage statistics from the operating system.

- Higher I/O wait events

  If you are experiencing higher I/O waits from the log writer or database writer processes, then the slower I/O effects throughput and response time. To observe the I/O effects, look at the historical data of the following wait events:

  - `Log file parallel writes`
  - `Log file sequential reads`
  - `Log file parallel reads`
  - `Data file parallel writes`
  - `Data file sequential reads parallel writes`

With SYNC transport, commits take more time because of the need to guarantee that the redo data is available on the standby database before foreground processes get an acknowledgment from the log writer (LGWR) background process that the commit has completed. A LGWR process commit includes the following wait events:

- `Log File Parallel Write` (local write for the LGWR process)
- `SYNC Remote Write`

Longer commit times for the LGWR process can cause longer response time and lower throughput, especially for small time-sensitive transactions. However, you may obtain sufficient gains by tuning the log writer local write (`Log File Parallel Write` wait event).

To tune the disk write I/O (`Log File Parallel Write` or the RFS I/O), add more spindles or increase the I/O bandwidth.

To reduce the network time:

- Tune the Oracle Net send and receive buffer sizes
- Set SDU=65535 (for more information, see Section , "Set the Network Configuration and Highest Network Redo Rates")
- Increase the network bandwidth if there is saturation
- Possibly find a closer site to reduce the network latency

With `ASYNC` transport, the LGWR process never waits for the network server processes to return before writing a `COMMIT` record to the current log file. However, if the network server processes has fallen behind and the redo to be shipped has been flushed from the log buffer, then the network server process reads from the online redo logs. This causes more I/O contention and possibly longer wait times for the log writer process writes (`Log File Parallel Write`). If I/O bandwidth and sufficient spindles are not allocated, then the log file parallel writes and log file sequential reads increase, which may affect throughput and response time. In most cases, adding sufficient spindles reduces the I/O latency.

Note: To enable most of the statistical gathering and advisors, ensure the `STATISTICS_LEVEL` initialization parameter is set to `TYPICAL` (recommended) or `ALL`.

See Also:

- *Oracle Database Performance Tuning Guide* for general performance tuning and troubleshooting best practices
- *Oracle Database Performance Tuning Guide* for Overview of the Automatic Workload Repository (AWR) and on Generating Automatic Workload Repository Reports
- The MAA white paper "Data Guard Redo Transport & Network Best Practices" from the MAA Best Practices area for Oracle Database at

  `http://www.oracle.com/goto/maa`

9

Configuring Oracle GoldenGate

Oracle GoldenGate delivers low-impact, real-time data acquisition, distribution, and delivery across both homogeneous and heterogeneous systems. Oracle GoldenGate enables cost-effective and low-impact real-time data integration and continuous availability solutions across a wide variety of use cases. Oracle GoldenGate offers close integration with Oracle technologies and applications, support for additional heterogeneous systems, and improved performance.

This chapter contains the following topics:

Oracle GoldenGate Overview

Oracle GoldenGate captures primary database changes by reading redo records from a source database online redo log file, transforming those records into a platform independent trail file format, and transmitting the trail file to a target database(s). Oracle GoldenGate maintains a logical replica by converting the trail file into SQL and applying SQL to a target database. A target database is open read/write while synchronization occurs. Additional Oracle GoldenGate information can be found at

http://www.oracle.com/us/products/middleware/data-integration/goldengate/resources/index.html

Oracle GoldenGate is ideally used where its flexibility can address advanced requirements not addressed by other MAA features. Oracle GoldenGate is an important element in the MAA architecture, useful for the following purposes:

- Active-Active multi-master configurations used for data availability and to scale performance. An important consideration for such configurations is the ability to manage update conflicts either by avoiding them or by implementing a process for conflict detection and resolution.
- Offload operational reporting when read/write access to the reporting instance is required.
- Near zero downtime (one-way replication) or zero downtime (bi-directional replication) for planned maintenance tasks, including:
  - Database upgrades
  - Application upgrades that modify back-end database objects (requires the user to implement transformations to map old and new versions.

- Database consolidation
- Database and platform migrations

Oracle GoldenGate and Oracle RAC

Oracle Real Application Clusters (Oracle RAC) enables multiple instances that are linked by an interconnect to share access to an Oracle database. In an Oracle RAC environment, Oracle Database runs on two or more systems in a cluster while concurrently accessing a single shared database. The result is a single database system that spans multiple hardware systems, enabling Oracle RAC to provide high availability and redundancy during failures in the cluster. Oracle GoldenGate is integrated with Oracle RAC and Cluster Ready Services (CRS) using the Oracle Grid Infrastructure Agent such that during cluster node failures, GoldenGate will automatically restart on a surviving node.

See Also:

- Oracle Grid Infrastructure Downloads web page for additional information on the Oracle Grid Infrastructure Agent at http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/index.html
- MAA white papers "Oracle GoldenGate on Oracle Exadata Database Machine Configuration" at http://www.oracle.com/technetwork/database/features/availability/maa-wp-gg-oracledbm-128760.pdf and "Oracle GoldenGate With Oracle Real Application Clusters Configuration" at http://www.oracle.com/technetwork/database/features/availability/maa-goldengate-rac-2007111.pdf for best practices for configuring GoldenGate on Oracle RAC

Oracle GoldenGate and Oracle Data Guard/Oracle Active Data Guard

Oracle GoldenGate is Oracle's strategic logical replication product. Oracle Data Guard is Oracle's strategic physical replication product focused on data protection and data availability, and is the standard MAA recommendation for such purposes because of the advantages it offers over logical replication. Oracle Data Guard is also commonly used in place of storage-remote mirroring or host-based mirroring solutions for disaster protection. Oracle Data Guard also minimizes planned downtime by supporting database rolling upgrades, select migrations (for example, Windows to Linux), data center moves, and other types of planned maintenance. Oracle Active Data Guard, an extension to Oracle Data Guard, is the simplest, fastest, most efficient method of maintaining a synchronized physical replica of a source database open read-only for offloading read-only workload and backups. For a detailed discussion of Data Guard advantages for data protection, see the Product Technical Brief, "Oracle Active Data Guard and Oracle GoldenGate" available from the GoldenGate link at

http://www.oracle.com/technetwork/database/features/availability/index.html

Oracle GoldenGate is often used in Data Guard configurations in a complementary manner. Oracle GoldenGate is integrated with Data Guard using the Oracle Grid Infrastructure Agent, so that during a Data Guard role transition (switchover or failover) the GoldenGate processes will restart automatically on the primary database. The Data Guard protection mode can be either MaxAvailability/MaxProtection (zero data loss) or MaxPerformance (data loss).

See Also: Oracle Grid Infrastructure Downloads web page for additional information on the Oracle Grid Infrastructure Agent at http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/index.html

Oracle GoldenGate and Edition-Based Redefinition

Edition-based redefinition is a capability implemented entirely within Oracle Database that enables database objects that implement the back end of an application to be patched or upgraded without interrupting the availability of the application. Edition-based redefinition enables customers to implement application upgrades online with zero database downtime. Edition-based redefinition requires Oracle Database 11*g* Release 2, and it requires application changes; an application must be made editionable to upgrade online.

Oracle GoldenGate can also be used for online application upgrades. The application itself does not need to be modified to implement an upgrade, but the administrator must have sufficient knowledge of the differences between old and new versions of the application to implement mapping between versions using Oracle GoldenGate. User control over the application, and user preference for the second major distinction between these technologies determines which approach makes the most sense to achieve a zero downtime application upgrade.

The second major difference between these technologies is that edition-based redefinition uses only the single database that ordinarily supports the application. Oracle GoldenGate uses a second synchronized database to execute the upgrade.

- Using edition-based redefinition, the old version of the application is in the old edition and the new version of the application is in the new edition - both within the same database; the edition is the isolation mechanism. Data that is represented the same in the old and the new versions of the application is represented only once in table columns used by both versions; only data that is represented differently in the two application versions must exist twice. Synchronization is needed, therefore, only for that typically small proportion of the total data that differs between the two versions. Because a cross edition trigger fires within a transaction, potential conflicts between the old and the new representations are prevented before they can be committed, and there is no need for conflict-resolution.
- Using Oracle GoldenGate, the old version of the application runs on the original database and the new version of the application runs on a second database; the second database is the isolation mechanism. All data - both that which is represented the same in the old and the new versions of the application and that which is represented differently in the two application versions must exist twice. Synchronization is needed, therefore, for all the data. The synchronization is implemented using code that intervenes in the replay mechanism for the SQL that is constructed by mining the redo logs. It is, therefore, non-transactional; and conflicts between the old and the new representations cannot be prevented. Rather, conflict-resolution must be implemented as an explicit, post-processing step.

Oracle GoldenGate Configuration Best Practices

There are several best practices for configuring GoldenGate Extract, Data pump and Replicat for optimal performance and High Availability to reduce increased latencies caused by downtime of GoldenGate processes, including the following:

Oracle GoldenGate Integrated Extract and Integrated Replicat

With Oracle GoldenGate version 12.1.2, Replicat can now operate in integrated mode for improved scalability within Oracle target environments. The apply processing functionality within the Oracle database is leveraged to automatically handle referential integrity and data description language (DDL) so that the operations are applied in the correct order. Extract can also be used in integrated capture mode with an Oracle database, introduced with Oracle GoldenGate version 11.2.1. Extract integrates with an Oracle database log mining server to receive change data from the database in the form of logical change records (LCR). Extract can be configured to capture from a local or downstream mining database. Because integrated capture is fully integrated with the database, no additional setup is required to work with Oracle RAC, ASM, TDE, and data compression which greatly simplify setup without sacrificing performance.

The latest version of Oracle GoldenGate can be downloaded from My Oracle Support, Patches and Updates.

To make use of Integrated Extract you must use database release 11.2.0.3 or later. The specific patch numbers required for 11.2.0.3 are listed in My Oracle Support node 1557031.1. Integrated capture mode Extract can also be used to capture changes from Oracle versions starting with 10.2.0.4 with downstream mining using an 11.2.0.3 or higher, mining database.

To make use of Integrated Replicat use database release of 11.2.0.4 or later.

Use of a Clustered File System

Using a clustered file system is fundamental to the continuing availability of Oracle GoldenGate checkpoint and trail files in the event of a node failure. Ensuring the availability of the checkpoint files is essential to ensure that, after a failure occurs, the Extract process can continue mining from the last known archived redo log file position and Replicat processes can start applying from the same trail file position before a failure occurred. Using Oracle Database Filesystem (DBFS) or Oracle Automatic Storage Management Cluster File System (Oracle ACFS) allows a surviving database or ASM instance to be the source of an Extract process or destination for the Replicat processes.

Best practices for configuring DBFS or Oracle ACFS for use with Oracle GoldenGate are described in each of the following references.

Note: The best practices provided in the following documents apply to all supported Oracle GoldenGate platforms, including Oracle Exadata Database Machine.

See Also: MAA white papers "Oracle GoldenGate on Oracle Exadata Database Machine Configuration" at http://www.oracle.com/technetwork/database/features/availability/maa-wp-gg-oracledbm-128760.pdf and "Oracle GoldenGate With Oracle Real Application Clusters Configuration" at http://www.oracle.com/technetwork/database/features/availability/maa-goldengate-rac-2007111.pdf for DBFS configuration recommendations

Oracle GoldenGate Operational Best Practices

See the following documents for more information about Oracle GoldenGate management and operational Best Practices:

- "Administering Oracle GoldenGate for Windows and UNIX" at http://docs.oracle.com/goldengate/1212/gg-winux/GWUAD/index.html
- "Installing and Configuring Oracle GoldenGate for Oracle Database" at http://docs.oracle.com/goldengate/1212/gg-winux/GIORA/index.html
- Oracle MAA GoldenGate Best Practices white papers

  Oracle Database - http://www.oracle.com/technetwork/database/features/availability/oracle-database-maa-best-practices-155386.html

  Oracle Exadata - http://www.oracle.com/technetwork/database/features/availability/exadata-maa-best-practices-155385.html

10

Client Failover Best Practices for Highly Available Oracle Databases

This section describes Oracle Database 12c configuration best practices to automatically transition application connections from a failed primary database to a new primary database after a Data Guard / Active Data Guard role transition has occurred. This configuration also applies to client's connection to Real Application Clusters databases. In addition it also describes best practices for Application Continuity and Transaction Guard, new with Oracle Database 12c.

At a high level, automating client failover in a Data Guard configuration includes relocating database services to the new primary database as part of a Data Guard failover, notifying clients that a failure has occurred to break them out of TCP timeout, and redirecting clients to the new primary database.

The sections below describe how to create role-based database services for both OCI and JDBC applications in a Data Guard configuration. Subsequent sections provide detailed configuration steps for enabling OCI and JDBC clients to receive FAN notifications and reconnect to a new primary database.

Types of Failures

Unplanned failures of an Oracle Database instance fall into these general categories:

- A server failure or other fault that causes the crash of an individual Oracle instance in an Oracle RAC database. To maintain availability, application clients connected to the failed instance must quickly be notified of the failure and immediately establish a new connection to the surviving instances of the Oracle RAC database.
- A complete-site failure that results in both the application and database tiers being unavailable. To maintain availability users must be redirected to a secondary site that hosts a redundant application tier and a synchronized copy of the production database.
- A partial-site failure where the primary database, a single-instance database, or all nodes in an Oracle RAC database become unavailable but the application tier at the primary site remains intact.

Configure Fast Connection Failover as a best practice to fully benefit from fast instance and database failover and switchover with Oracle RAC and Oracle Data Guard. Fast Connection Failover enables clients, mid-tier applications, or any program that connects directly to a database to failover quickly and seamlessly to an available database service when a database service becomes unavailable.

This chapter contains the following topics:

See Also:

- The MAA white paper "Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 12c from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa

- "Application High Availability with Services and FAN" in *Oracle Database Administrator's Guide*

Automating Client Failover - JDBC, OCI, and ODP.Net

You can enable OCI and JDBC, and ODP.Net application clients to receive FAN notifications and quickly reconnect to a new primary database. The configuration best practices to enable fast connection failover differ depending on the client type.

See Also:

- *Oracle Database Administrator's Guide* for more information about Enabling Fast Connection Failover for Oracle Call Interface Clients
- *Oracle Call Interface Programmer's Guide* for more information about Transparent Application Failover in OCI
- "Client Failover Best Practices for Highly Available Oracle Databases - Oracle Database 12c" at
  http://www.oracle.com/technetwork/database/availability/client-failover-2280805.pdf
- "Client Failover Best Practices for Data Guard 11g Release 2" at
  http://www.oracle.com/technetwork/database/features/availability/maa-wp-11gr2-client-failover-173305.pdf

Configuring Fast Connection Failover for JDBC Clients

Prerequisites:

- The Universal Connection Pool (UCP) is enabled (UCP 12.1.0.2 or later)
- The application uses service names to connect to the database
- Oracle Notification Service (ONS) is configured and available on the node where JDBC is running

- The Java Virtual Machine (JVM) in which your JDBC instance runs must have oracle.ons.oraclehome set to point to your ORACLE_HOME

1. Enable Fast Connection Failover (FCF) and configure the JDBC application to connect to all ONS daemons for both the primary and standby clusters using the setONSConfiguration property. The setONSConfiguration property should point to all primary and standby ONS daemons.

```
pds.setONSConfiguration("nodes=adczatdb01:6200,adczatdb02:6200,slcc17adm01:6200
,slcc17adm02:6200");
pds.setFastConnectionFailoverEnabled(true);
```

2. By default the JDBC application will randomly pick three hosts from the setONSConfiguration property and create connections to those three ONS daemons. This default must be changed so that connections are made to all ONS daemons. This is done by setting the following property when the JDBC application is invoked to the total number of ONS daemons in the configuration:

```
java -Doracle.ons.maxconnections=4
```

3. The JDBC client must set the oracle.net.ns.SQLnetDef.TCP_CONNTIMEOUT_STR property. This property enables the JDBC client to quickly traverse an ADDRESS_LIST in the event of a failure. For example, if the client attempts to connect to a host that is unavailable, the connection attempt will be bounded to the time specified by the SQLnetDef.TCP_CONNTIMEOUT_STR property after which the client attempts to connect to the next host in the ADDRESS_LIST. The behavior continues for each host in the ADDRESS_LIST until a connection is made. Setting the property to a value of 3 seconds will suffice in most environments. It is important to note that the SQLnetDef.TCP_CONNTIMEOUT_STR property should be set on the data source and not on the Universal Connection Pool.

```
Properties prop = new Properties();  prop.put(oracle.net.ns.SQLnetDef.TCP_
CONNTIMEOUT_STR, ""+3000); // 3000ms
pds.setConnectionProperties(prop);
```

4. Set the thinForceDNSLoadBalancing property to get the correct behavior from SCAN load balancing:

```
// need to set oracle.jdbc.thinForceDNSLoadBalancing
prop.put("oracle.jdbc.thinForceDNSLoadBalancing","true");
```

5. Configure JDBC clients to use a connect descriptor that includes an address list that in turn includes the SCAN address for each site and connects to an existing service. Do not configure both TAF and JDBC FCF when using JDBC thick clients.

   The following URL we search both primary and standby sites looking for the appropriate service with very little overhead. This URL configuration is the recommended approach:

```
PoolDataSource pds = PoolDataSourceFactory.getPoolDataSource();
pds.setConnectionFactoryClassName("oracle.jdbc.pool.OracleDataSource");
pds.setUser("system");
 pds.setPassword("oracle");
 String dbURL =
 "jdbc:oracle:thin:@" +
 "(DESCRIPTION=" +
"(FAILOVER=on)" +
 "(ADDRESS_LIST=" +
 "(LOAD_BALANCE=on)" +
 "(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)" +
 "(ADDRESS=(PROTOCOL=TCP)(HOST=prmy-scan)(PORT=1521))"+
```

```
"(ADDRESS=(PROTOCOL=TCP)(HOST= stby-scan)(PORT=1521)))" +
 "(CONNECT_DATA=(SERVICE_NAME=oltpworkload)))"
 System.out.println("Url=" + dbURL);
 pds.setURL(dbURL);
```

The following URL should be used if it is very rare for the primary to ever run on the secondary site and you wish to have connections connect as fast as possible:

```
PoolDataSource pds = PoolDataSourceFactory.getPoolDataSource();
pds.setConnectionFactoryClassName("oracle.jdbc.pool.OracleDataSource");
pds.setUser("system");
 pds.setPassword("oracle");
 String dbURL =
 "jdbc:oracle:thin:@" +
 "(DESCRIPTION_LIST=" +
 "(LOAD_BALANCE=off)" +
 "(FAILOVER=on)" +
 "(DESCRIPTION=" +
 "(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)" +
 "(ADDRESS_LIST=" +
 "(LOAD_BALANCE=on)" +
 "(ADDRESS=(PROTOCOL=TCP)(HOST=prmy-scan)(PORT=1521)))" +
 "(CONNECT_DATA=(SERVICE_NAME=oltpworkload)))" +
 "(DESCRIPTION=" +
 "(ADDRESS_LIST=" +
 "(LOAD_BALANCE=on)" +
 "(ADDRESS=(PROTOCOL=TCP)(HOST= stby-scan)(PORT=1521)))" +
 "(CONNECT_DATA=(SERVICE_NAME=oltpworkload))))";
 System.out.println("Url=" + dbURL);
 pds.setURL(dbURL);
```

Note that if a switchover or failover occurs the above URL will force all connections to go through the old prmy-scan before using the stby scan where the primary currently runs.

See Also:

- *Oracle Database Administrator's Guide* for more information about Enabling Fast Connection Failover for JDBC Clients
- "Application High Availability with Services and FAN" in *Oracle Database Administrator's Guide*

Configuring Application Continuity

Application Continuity with Oracle Database 12c is used to mask outages for planned maintenance that is performed in rolling fashion across Oracle RAC instances or across a Data Guard primary and standby database. Application Continuity also masks unplanned outages of an Oracle RAC instance or a Data Guard primary database configured in Maximum Availability (zero data loss failover) with Data Guard Fast-Start Failover (automatic database failover). Use of Application Continuity for a Data Guard failover requires that both source and target databases be at Oracle 12.1.0.2 or later.

Application Continuity is available for:

- Oracle JDBC Replay Driver 12c or later. This is a JDBC driver feature provided with Oracle Database 12c for Application Continuity, referred to as the "replay driver" onwards (OCI support is planned for a future release).

- Oracle Universal Connection Pool, Oracle WebLogic Server 12c (12.1.2) or later, and third-party Java connection pools or standalone Java applications - using Oracle JDBC- Replay Driver 12c or later.
- Standard 3rd Party Java application servers using the pooled connection interface - including IBM WebSphere and Apache Tomcat, from 12.1.0.2
- Standard 3rd Party Java application servers supporting the Universal Connection Pool as the pooled data source - including IBM WebSphere, Apache Tomcat, and RedHat JBoss
- Third-party Java connection pools or standalone Java applications - using Oracle JDBC- Replay Driver 12c or later and embedding their own request boundaries

Application Continuity uses Transaction Guard to reliably determine if the last transaction was committed or not. Without Transaction Guard, applications and users who attempt to retry operations following an outage can cause logical corruption by committing duplicate transactions or committing transactions out of order.

The following sections describe the options to configure Application Continuity using the Oracle JDBC 12c Replay Driver depending on your configuration:

- Configuring Oracle UCP 12c
- Configuring Oracle WebLogic Server 12c
- Configuring Standalone Java Applications or Third-party Connection Pools
- Configuring Connections for High Availability (Failover and Failback)
- Configuring Services for Application Continuity
- Checking Resource Allocation

See Also:

- MAA white paper http://www.oracle.com/technetwork/database/database-cloud/private/application-continuity-wp-12c-1966213.pdf
- MAA white paper http://www.oracle.com/technetwork/database/database-cloud/private/transaction-guard-wp-12c-1966209.pdf

Configuring Oracle UCP 12c

To configure the Oracle JDBC 12c Replay Data Source as a connection factory on UCP PoolDataSource:

```
setConnectionFactoryClassName("oracle.jdbc.replay.OracleDataSourceImpl");
```

Configuring Oracle WebLogic Server 12c

To configure the Oracle 12c JDBC Replay Data Source use the Oracle WebLogic Server Administration Console.

Configuring Standalone Java Applications or Third-party Connection Pools

To configure the Oracle JDBC 12c Replay Data Source in the property file or in the thin JDBC application

```
replay datasource=oracle.jdbc.replay.OracleDataSourceImpl
```

Configuring Connections for High Availability (Failover and Failback)

1. The REMOTE_LISTENER setting for the database must include the addresses in the ADDRESS_LISTs for all URL used for client connection:
   - If any URL uses the SCAN Names, then REMOTE_LISTENERS must include the SCAN Name.
   - If any URL uses an ADDRESS_LIST of host VIPs, then REMOTE_LISTENERS must include an ADDRESS list including all SCAN VIPs and all host VIPs
2. Set RETRY_COUNT, CONNECT_TIMEOUT parameters in the URL to allow new incoming connections to retry. For a complete discussion on this attributes please consult the JDBC Application Configuration Section later in this paper. For example,

```
"jdbc:oracle:thin:@" +
 "(DESCRIPTION=" +
"(FAILOVER=on)" +
 "(ADDRESS_LIST=" +
 "(LOAD_BALANCE=on)" +
 "(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)" +
 "(ADDRESS=(PROTOCOL=TCP)(HOST=prmy-scan)(PORT=1521))"+
"(ADDRESS=(PROTOCOL=TCP)(HOST= stby-scan)(PORT=1521)))" +
 "(CONNECT_DATA=(SERVICE_NAME=oltpworkload)))"
```

Configuring Services for Application Continuity

1. Set the service attributes using SRVCTL / GDSCTL to use Application Continuity.
2. Set FAILOVER_TYPE to TRANSACTION to enable Application Continuity.
3. Set COMMIT_OUTCOME to TRUE to enable Transaction Guard (mandatory).
4. Review the following service attributes:
   - REPLAY_INITIATION_TIMEOUT : Set this to the duration in seconds after which replay is not started (e.g. 180, 300, 1800 seconds - the override to cancel replay). This timer starts at beginRequest. (default 300 seconds)
   - FAILOVER_RETRIES : Set this to specify the number of connection retries for each replay attempt. (default 30 retries, applied at replay driver)
   - FAILOVER_DELAY : Set this to specify the delay in seconds between connection retries (default 10 seconds, applied at replay driver)
   - AQ_HA_NOTIFICATIONS: Set this to TRUE to enable FAN (default TRUE)

The following is an example of configuring services for Application Continuity:

```
srvctl add service -db mts -service oltpworkload -role PRIMARY -notification TRUE
-session_state dynamic -failovertype transaction -failovermethod basic -commit_
outcome TRUE -failoverretry 30 -failoverdelay 10 -replay_init_time 900 -clbgoal
SHORT -rlbgoal SERVICE_TIME -preferred mts1,mts2  -retention 3600 -verbose
```

Checking Resource Allocation

- Ensure that the system has the necessary memory and CPU resources.
- Memory: The JDBC replay driver uses more memory than the base JDBC driver because the calls are retained until the end of a database request. If the number of calls retained is small, then the memory consumption of the replay driver is comparable to the base driver. At the end of a request, the calls are released to the

garbage collector. This action differs from the base driver that releases as calls are closed.

- For good performance, if there is sufficient memory, allocate 4 to 8 GB (or more) of memory for the Virtual Machine (VM), for example, by setting - Xms4096m for 4 GB.
- CPU: The JDBC replay driver uses some additional CPU for building proxy objects, managing queues, and for garbage collection. The server uses some additional CPU for managing the validation. CPU overhead is red.

Configuring Fast Connection Failover for OCI Clients

Prerequisites

- An Oracle RAC environment with Oracle Clusterware set up and enabled or a single node (non-Oracle RAC) database with Oracle Restart
- The application must have been linked with the threads library
- The OCI environment must be created in OCI_EVENTS and OCI_THREADED mode

Enable FAN for OCI clients by initializing the environment with the OCI_EVENTS parameter, as in the following example:

```
OCIEnvCreate(...OCI_EVENTS...)
```

1. Link the OCI client applications with thread library libthread or libpthread.
2. Your application will need the ability to check if an event has occurred by using code similar to that used in the following example:

```
void evtcallback_fn(ha_ctx, eventhp)
...
printf("HA Event received.\n");
  if (OCIHandleAlloc( (dvoid *)envhp, (dvoid **)&errhp, (ub4) OCI_HTYPE_ERROR,
                      (size_t) 0, (dvoid **) 0))
    return;
  if (retcode = OCIAttrGet(eventhp, OCT_HTYPE_EVENT, (dvoid *)&srvhp, (ub4 *)0,
                           OCI_ATTR_HA_SRVFIRST, errhp))
    checkerr (errhp, (sword)retcode;
  else {
     printf("found first server handle.\n");
     /*get associated instance name */
     if (retcode = OCIAttrGet(srvhp, OCI_HTYPE_SERVER, (dvoid *)&instname,
                              (ub4 *)&sizep, OCI_ATTR_INSTNAME, errhp))
       checkerr (errhp, (sword)retcode);
     else
       printf("instance name is %s.\n", instname);
```

3. Clients and applications can register a callback that is invoked whenever a high availability event occurs, as shown in the following example:

```
/*Registering HA callback function */
  if (checkerr(errhp, OCIAttrSet(envhp, (ub4) OCI_HTYPE_ENV,
                                 (dvoid *)evtcallback_fn, (ub4) 0,
                                 (ub4)OCI_ATTR_EVTCBK, errhp)))
  {
    printf("Failed to set register EVENT callback.\n");
    return EX_FAILURE;
  }
  if (checkerr(errhp, OCIAttrSet(envhp, (ub4) OCI_HTYPE_ENV,
```

```
                                    (dvoid *)evtctx, (ub4) 0,
                                    (ub4)OCI_ATTR_EVTCTX, errhp)))
  {
    printf("Failed to set register EVENT callback context.\n");
    return EX_FAILURE;
  }
return EX_SUCCESS;
```

After registering an event callback and context, OCI will call the registered function once for each high availability event.

4. Configure an Oracle Net alias that the OCI application will use to connect to the database. The Oracle Net alias should specify both the primary and standby SCAN hostnames. For best performance while creating new connections the Oracle Net alias should have LOAD_BALANCE=OFF for the DESCRIPTION_LIST so that DESCRIPTIONs are tried in an ordered list, top to bottom. With this configuration the second DESCRIPTION is only attempted if all connection attempts to the first DESCRIPTION have failed.

```
SALES=
 (DESCRIPTION_LIST=
  (LOAD_BALANCE=off)
  (FAILOVER=on)
  (DESCRIPTION=     (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_
COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=TCP)(HOST=prmy-scan)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME=oltpworkload)))
  (DESCRIPTION=
(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=TCP)(HOST=stby-scan)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME=oltpworkload))))
```

When a new connection is made using the above Oracle Net alias the following logic is used:

a. Oracle Net contacts DNS and resolves prmy-scan to a total of three IP addresses.

b. Oracle Net randomly picks one of the three IP address and attempts to make a connection. If the connection attempt to the IP address does not respond in three seconds (TRANSPORT_CONNECT_TIMEOUT) the next IP address is attempted. All three IP addresses will be tried a total of four times (initial attempt plus RETRY_COUNT in the above example).

c. If the connection to primary site is unsuccessful, it then contacts DNS and resolves stby-scan to three addresses.

d. The same sequence is performed for the standby stby-scan as it was for the prmy-scan.

The following is some additional information about the Oracle Net parameters used in the above alias:

- LOAD_BALANCE is ON by default for DESCRIPTION_LIST only. This parameter by default is OFF for an address list within a DESCRIPTION. Setting this ON for a SCAN-based address implies that new connections will be randomly assigned to one of the 3 SCAN-based IP addresses resolved by DNS.

- The RETRY_COUNT parameter specifies the number of times an address list is traversed before the new connection attempt is terminated. The default value is 0. With respect to SCAN, with FAILOVER = on, setting this RETRY_COUNT parameter to a value of 2, for example, means the three SCAN IP addresses are traversed thrice (i.e. 3*3=9 connect attempts), before the connection is terminated:
  - When the connection request initially comes in, the first randomly assigned IP address tries to service that request, followed by the two remaining IP addresses. (This behavior is controlled by the FAILOVER parameter.)
  - The retries then kick in and the list of three IP addresses is tried two more times. RETRY_COUNT is only supported at DESCRIPTION level in connect string, but not at global (i.e. sqlnet.ora) level.

Configuring Automatic Failover for ODP.Net Clients

Prerequisites:

- Namespace: Oracle.DataAccess.Client, Assembly: Oracle.DataAccess.dll.
- Microsoft .NET Framework Version 2.0 or later.
- Configure Oracle Net alias as described in Section , "Configuring Fast Connection Failover for OCI Clients."

1. Enable Fast Connection Failover for ODP.NET connection pools by subscribing to FAN high availability events. To enable Fast Connection Failover, include "HA Events=true" and "pooling=true" in the connection string, as shown in the following example where user_name is the name of the database user and password is the password for that user:

```
con.ConnectionString =
   "User Id=user_name;Password=password;Data Source=sales;" +
   "Min Pool Size=10;Connection Lifetime=120;Connection Timeout=60;" +
   "HA Events=true;Incr Pool Size=5;Decr Pool Size=2";
```

2. To take advantage of load balancing events with ODP.NET connection pools, set the load balancing attribute in the ConnectionString to TRUE (the default is FALSE). You can do this at connect time. This only works if you are using connection pools, or when the pooling attribute is set to TRUE which is the default.

   The following example demonstrates how to configure the ConnectionString to enable load balancing

```
con.ConnectionString =
  "User Id=user_name;Password=password;Data Source=odpapp;" +
  "Min Pool Size=10;Connection Lifetime=120;Connection Timeout=60;" +
  "Load Balancing=true;Incr Pool Size=5;Decr Pool Size=2";
```

3. Configure the service to enable FAN and HA events as described in Section , "Configuring Database Services."

See Also:

- "Client Failover Best Practices for Highly Available Oracle Databases - Oracle Database 12c" at http://www.oracle.com/technetwork/database/availability/client-failover-2280805.pdf
- "Client Failover Best Practices for Data Guard 11g Release 2" at http://www.oracle.com/technetwork/database/features/availability/maa-wp-11gr2-client-failover-173305.pdf
- *Oracle Database Administrator's Guide*

Configuring Oracle RAC Databases for Failover

Oracle Database 12*c* provides the infrastructure to make your application data highly available with Oracle Real Application Clusters (Oracle RAC) and with the Oracle Data Guard. At the database tier you must configure fast application failover.

Configuring Database Services

At a high level, automating client failover in an Oracle RAC configuration includes relocating database services to new or surviving instances, notifying clients that a failure has occurred to break the clients out of TCP timeout, and redirecting clients to a surviving instance (Oracle Clusterware sends FAN messages to applications; applications can respond to FAN events and take immediate action). For more information about FAN, see Section , "Client Configuration and Migration Concepts".

For services on an Oracle RAC database, Oracle Enterprise Manager or the SRVCTL utility are the recommended tools to manage services. A service can span one or more instances of an Oracle database and a single instance can support multiple services. The number of instances offering the service is managed by the database administrator independent of the application.

See Also:

- Section , "Client Configuration and Migration Concepts"
- Section , "Connect to Database Using Services and Single Client Access Name (SCAN)"

Optionally Configure FAN Server Side Callouts

Server-side callouts provide a simple, yet powerful integration mechanism with the High Availability Framework that is part of Oracle Clusterware. You can use server side callouts to log trouble tickets or page Administrators to alert them of a failure. For Up events, when services and instances are started, new connections can be created so the application can immediately take advantage of the extra resources

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for an Introduction to Automatic Workload Management.
- For more information about client failover best practices and details on deploying FAN server side callouts, see the Technical Article, "Automatic Workload Management with Oracle Real Application Clusters 11*g* Release 2" on the Oracle Technology Network at

  http://www.oracle.com/technetwork/database/clustering/overview/index.html

Configuring the Oracle Data Guard Environment

The following topics describe how to configure the Oracle Data Guard environment:

- Configuring Database Services
- Use Data Guard Broker

See Also: The MAA white paper "Client Failover Best Practices for Highly Available Oracle Databases - Oracle Database 12*c*" from the MAA Best Practices area for Oracle Database at

http://www.oracle.com/goto/maa

Configuring Database Services

In an Oracle Data Guard configuration you should only run primary application services on the primary database and run standby application services on the standby database. You can automatically control the startup of database services on primary and standby databases by assigning a database role to each service (roles include: `PRIMARY`, `PHYSICAL_STANDBY`, `LOGICAL_STANDBY`, and `SNAPSHOT_STANDBY`).

A database service automatically starts upon database startup if the management policy for the service is `AUTOMATIC` and if a role assigned to that service matches the current role of the database.

See Also:

- *Oracle Database Administrator's Guide* for information about Creating and Deleting Database Services with SRVCTL
- *Oracle Database Administrator's Guide* for information About Automatic Startup of Database Services

Use Data Guard Broker

The best practice is to configure Oracle Data Guard to manage the configuration with Oracle Data Guard Broker. Oracle Data Guard Broker is responsible for sending FAN events to client applications to clean up their connections to the down database and reconnect to the new production database.

Oracle Clusterware must be installed and active on the primary and standby sites for both single instance (using Oracle Restart) and Oracle RAC databases. Oracle Data Guard broker coordinates with Oracle Clusterware to properly fail over role-based services to a new primary database after a Data Guard failover has occurred.

See Also:

- Section , "Client Configuration and Migration Concepts" for more information about FAN.
- Section , "Use Oracle Data Guard Broker with Oracle Data Guard"

Client Transition During Switchover Operations

In Oracle Data Guard, the term "switchover" describes a planned event where a primary and standby database switch roles, usually to minimize the downtime while performing planned maintenance. The configuration best practices to address unplanned failovers also address most of the requirements for a planned switchover, except for several additional manual steps that apply to logical standby databases (SQL Apply).

Note: There are no additional considerations for switchovers using Oracle Active Data Guard.

The following steps describe the additional manual switchover steps for Oracle Data Guard 11*g* Release 2:

1. The primary database is converted to a standby database. This disconnects all sessions and brings the database to the mount state. Oracle Data Guard Broker shuts down any read/write services.
2. Client sessions receive a ORA-3113 and begin going through their retry logic (TAF for OCI and application code logic for JDBC).
3. The standby database is converted to a primary database and any existing sessions are disconnected. Oracle Data Guard Broker shuts down read-only services.
4. Read-only connections receive an ORA-3113 and begin going through their retry logic (TAF for OCI and application code logic for JDBC).
5. As the new primary and the new standby are opened, the respective services are started for each role and clients performing retries now see the services available and connect.

For logical standby switchover:

1. Ensure that the proper reconnection logic has been configured (for more information, see Section , "Automating Client Failover - JDBC, OCI, and ODP.Net" and Section , "Configuring Oracle RAC Databases for Failover"). For example, configure `TAF` and `RETRY_COUNT` for OCI applications and code retry logic for JDBC applications.
2. Stop the services that the primary application uses and the read-only applications enabled on the standby database.
3. Disconnect or shutdown the primary and read-only application sessions.
4. Once the switchover has completed, restart the services used by the primary application and the read-only application.
5. Sessions that were terminated reconnect once the service becomes available as part of the retry mechanism.
6. Restart the application if an application shuts down.

Note that FAN is not needed to transition clients during a switchover operation if the application performs retries. FAN is only needed to break clients out of TCP timeout, a state that should only occur during unplanned outages.

See Also: Section , "How to Perform Data Guard Switchover"

Preventing Login Storms

The process of failing over an application that has a large number of connections may create a login storm. A *login storm* is a sudden spike in the number of connections to a database instance, which drains CPU resources. As CPU resources are depleted, application timeouts and application response times are likely to increase.

To control login storms:

- Implement the Connection Rate Limiter

  The primary method of controlling login storms is to implement the Connection Rate Limiter feature of the Oracle listener. This feature limits the number of connections that can be processed in seconds. Slowing down the rate of connections ensures that CPU resources remain available and that the system remains responsive.

- Configure Oracle Database for shared server operations

  In addition to implementing the Connection Rate Limiter, some applications can control login storms by configuring Oracle Database for shared server operations. By using shared server, the number of processes that must be created at failover time are greatly reduced, thereby avoiding a login storm.

- Adjust the maximum number of connections in the mid tier connection pool

  If such a capability is available in your application mid tier, try limiting the number of connections by adjusting the maximum number of connections in the mid tier connection pool.

  See Also:

  - *Oracle Database Administrator's Guide* for more information about configuring and controlling shared server operations
  - The "Oracle Net Listener Connection Rate Limiter" white paper for information about the Connection Rate Limiter at

    `http://www.oracle.com/technetwork/database/enterprise-edition/oraclenetservices-connectionratelim-133050.pdf`
  - The "Best Practices for Optimizing Availability During Unplanned Outages Using Oracle Clusterware and Oracle Real Application Clusters" white paper for information and examples about listener connection rate throttling from the MAA Best Practices area for Oracle Database at

    `http://www.oracle.com/goto/maa`

Configuring Global Data Services

Global Data Services enables administrators to automatically and transparently manage client workloads across replicated databases that offer common services. A database service is a named representation of one or more database instances. Services enable you to group database workloads and route a particular work request to an

appropriate instance. A global service is a service provided by multiple databases synchronized through data replication.

Global Data Services provides dynamic load balancing, failover, and centralized service management for a set of replicated databases that offer common services. The set of databases can include Oracle RAC and noncluster Oracle databases interrelated through Oracle Data Guard, databases consolidated under Oracle Multitenant, Oracle GoldenGate, or any other replication technology.

The global services management framework is built around the following preexisting Oracle Database technologies:

- Oracle Real Application Clusters (Oracle RAC) - Enables dynamic load balancing and workload management in a cluster
- Oracle Active Data Guard - Enables high-performance farms of read-only databases
- Data Guard Broker - Enables creation, management, and monitoring of Data Guard configurations that include a primary database and up to 30 standby databases
- Oracle GoldenGate - Enables replication updates among multiple databases

Oracle GDS provides the following key capabilities for a set of replicated databases that are globally distributed or located within the same data center:

- Region-based workload routing
- Connect-time Load balancing
- Provides Run-time load balancing advisory for Oracle integrated clients
- Inter-database Service failover
- Replication lag based workload routing for Active Data Guard
- Role-based global Services for Active Data Guard
- Centralized workload management framework

At a high level, configuring the Global Data Services Framework involves the following:

- Installing a Global Service Manager. You must install at least one global service manager for each Global Data Services region. Global service managers can be hosted on physical or virtual environments. For high availability, Oracle recommends installing multiple (typically 3) global service managers in each region running on separate hosts.
- Creating a Global Data Services Catalog. The catalog must reside in a 12c database and it is recommended that database be outside the GDS configuration. GDS catalog may be co-hosted along with catalogs of RMAN or Oracle Enterprise Manager. Oracle recommends that you use Oracle high availability features such as Oracle Real Application Clusters (Oracle RAC) and Oracle Data Guard to protect the Global Data Services catalog against outages.
- Registering a Global Service Manager to the catalog.
- Adding a Global Services Pool. A GDS pool is a subset of databases that provide a set of global services that belong to an administrative domain. This simplifies service management and provides for high security by allowing each pool to be administered by a different administrator.

- Adding a Global Data Services Region. A region is a set of databases that share the same network proximity and network latency. A region normally corresponds to a local area network. For high availability purposes, each region in a GDS configuration should have a designated buddy region, which is a region that contains global service managers that can provide continued access to a GDS configuration if the global services managers in the local region become unavailable.
- Adding databases to the GDS pool. To be part of a Global Data Services pool, a database must use a server parameter file (SPFILE). An Oracle RAC database should also have SCAN set up.
- Add a service to a GDS pool. A global service name must be unique within a GDS pool and must also be unique within a GDS configuration. A global service cannot be created at a database if a local or global service with the same name already exists at that database.

See Also:

- *Oracle Database Global Data Services Concepts and Administration Guide*
- MAA white paper "Oracle Database 12c - Global Data Services" at http://www.oracle.com/technetwork/database/availability/global-data-services-12c-wp-1964780.pdf

Configuring the Database Client

When clients connect to the database they do so using an Oracle Net connect string that specifys a service. The connect string used when connecting to a global service is unique in the following ways:

- Service name parameter must specify a global service
- Multiple SCAN addresses are used to point to global service manager endpoints
- The database client's region may be specified in the connection data section

Consider the following connect string:

```
(DESCRIPTION=
  (FAILOVER=on)
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(host=sales-east1)(port=1522))
    (ADDRESS=(host=sales-east2)(port=1522))
    (ADDRESS=(host=sales-east3)(port=1522)))
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(host=sales-west1)(port=1522))
    (ADDRESS=(host=sales-west2)(port=1522))
    (ADDRESS=(host=sales-west3)(port=1522)))
  (CONNECT_DATA=
   (SERVICE_NAME=sales)
   (REGION=east)))
```

Client-side load balancing is enabled across the global service managers within each region by setting the LOAD_BALANCE parameter to ON in the address list for each region. Connect-time failover between regions is enabled by setting the FAILOVER parameter to ON.

11

Monitoring for High Availability

This chapter provides best practices for monitoring your system using Enterprise Manager and to monitor and maintain a highly available environment across all tiers of the application stack.

This chapter contains the following topics:

Overview of Monitoring and Detection for High Availability

Continuous monitoring of the host, network, database operations, application, and other system components ensures early detection of problems. Early detection improves the user's system experience because problems can be avoided or resolved faster. In addition, monitoring captures system metrics to indicate trends in system performance, growth, and recurring problems. This information can facilitate prevention, enforce security policies, and manage job processing. For the database server, a sound monitoring system must measure availability and detect events that can cause the database server to become unavailable, and provide immediate notification about critical failures to responsible parties.

The monitoring system itself must be highly available and adhere to the same operational best practices and availability practices as the resources it monitors. Failure of the monitoring system leaves all monitored systems unable to capture diagnostic data or alert the administrator about problems.

Enterprise Manager provides management and monitoring capabilities with many different notification options. Recommendations are available for methods of monitoring the environment's availability and performance, and for using the tools in response to changes in the environment.

Using Enterprise Manager for System Monitoring

A major benefit of Enterprise Manager is its ability to manage components across the entire application stack, from the host operating system to a user or packaged application. Enterprise Manager treats each of the layers in the application as a *target*. Targets—such as databases, application servers, and hardware—can then be viewed along with other targets of the same type, or can be grouped by application type. You can also review related targets in a single view from the High Availability Console (for

more information, Section , "Manage Database Availability with the High Availability Console"). Each target type has a default generated home page that displays a summary of relevant details for a specific target. You can group different types of targets by function; that is, as resources that support the same application.

Every target is monitored by an Oracle Management Agent. Every Management Agent runs on a host and is responsible for a set of targets. The targets can be on a host that is different from the one that is used by the Management Agent. For example, a Management Agent can monitor a storage array that cannot host an agent natively. When a Management Agent is installed on a host, the host is automatically discovered along with other targets that are on the machine.

Moreover, to help you implement the Maximum Availability Architecture (MAA) best practices, Enterprise Manager provides the MAA Advisor (for more information, see Section , "Configure High Availability Solutions with MAA Advisor"). The MAA Advisor page recommends Oracle solutions for most outage types and describes the benefits of each solution.

In addition to monitoring infrastructure with Enterprise Manager in the Oracle HA environment, Oracle Auto Service Request (ASR) can be used to resolve problems faster by using auto-case generation for Oracle's servers, storage systems, components, and Engineered Systems when specific hardware faults occur. For more information, see "Oracle Auto Service Request" in My Oracle Support Note 1185493.1 at

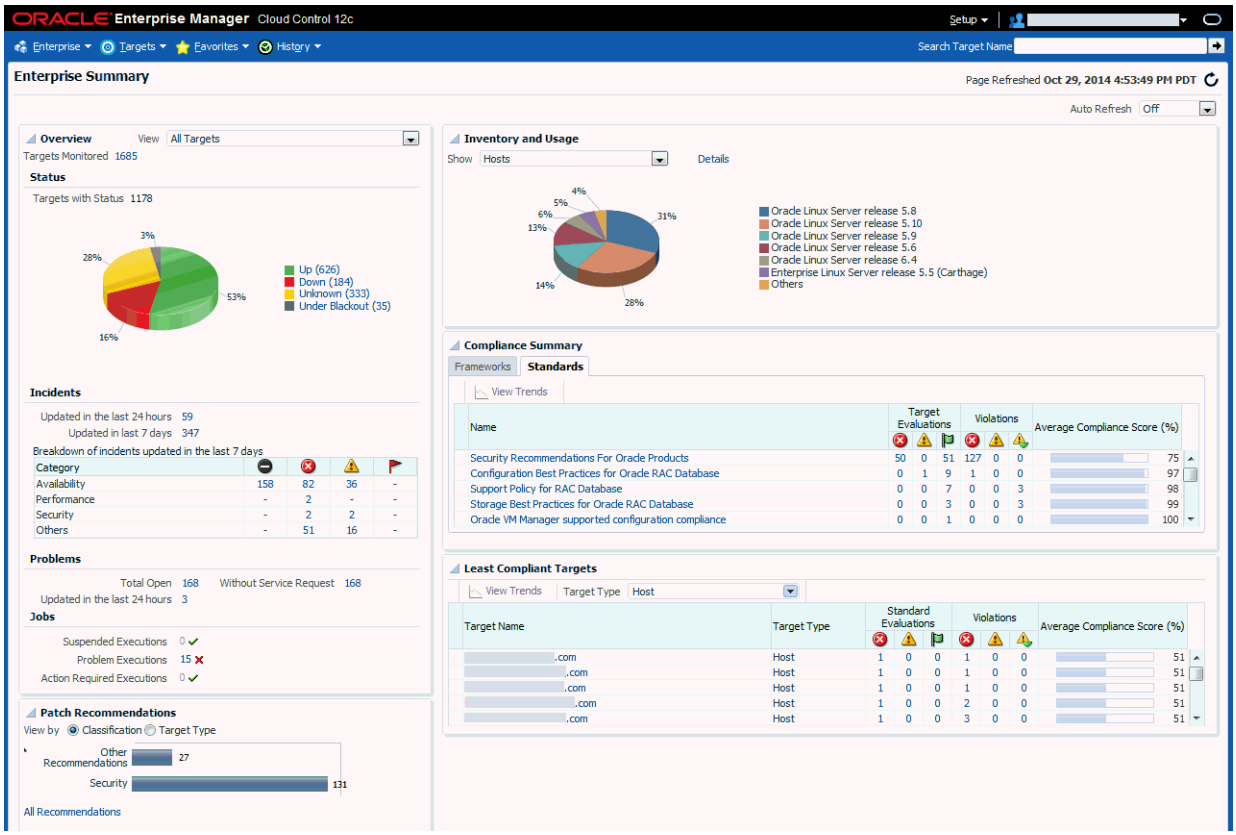
`https://support.oracle.com/rs?type=doc&id=1185493.1`

> **See Also:** *Enterprise Manager Cloud Control Introduction* for information about Enterprise Manager Architecture and the Oracle Management Agent

Oracle Enterprise Manager Home Page

Administrators can select the Enterprise Manager Home Page that provides the most relevant information based upon their role, choosing from a set of suggested pages such as the Enterprise Summary page shown in Figure 11–1 that shows the availability of all discovered targets. Administrators can also select any page in Enterprise Manager as their home page.

Figure 11–1 Enterprise Summary Page



The Enterprise Summary page provides administrators with a consolidated view of rollup information across a variety of areas that can affect availability, from critical incidents and failed jobs to recommended patches and compliance violations, providing the ability to drill into relevant details for further analysis, to take action to react to critical issues, and to proactively maintain environments to ensure availability.The Enterprise Summary page includes the following information:

- A snapshot of the current availability of all targets. The Status pie chart gives the administrator an immediate indication of how many targets are available (Up), unavailable (Down), or have lost communication with the console (Unknown), or have specifically had their monitoring suspended for reasons such as maintenance operations (Under Blackout). Click on any of the statuses to drill down into a list of the targets in that state, and drill further into individual targets to analyze and take corrective action.
- An overview of how many incidents and problems are known in the entire monitored system. Drill down to view matching incidents in the Incident Manager by clicking the links. The administrator can access the Incident Manager directly from any Enterprise Manager page by selecting Enterprise > Monitoring > Incident Manager or by pressing Ctrl+Shift+I.
- The number of suspended, problem (stopped/failed), and action required executions for all Enterprise Manager jobs. Click the number next to the status group to view a list of those jobs.
- Patch recommendations, viewable by classification or target type. Drill down to view details in the Patches & Updates page for the selected set of patches or to view all recommendations.

- A Compliance Summary displaying by compliance framework and compliance standard the results of compliance standard evaluations and the severity and total number of related violations for all managed targets. Drill down to view the Compliance Reports page and to determine the source and type of violation.
- A view of the least compliant targets by selected target type, including the result of compliance standard evaluations and the severity and total number of related violations. Drill down to determine the source and type of violation.

The following sections provide best practices, configuration recommendations, and relevant links to additional information regarding these capabilities. Tailoring the monitoring to the business needs and details of your environment helps to ensure that the data displayed on this page helps your administrators effectively manage the availability of your environments.

Your ability to use these capabilities is dependent upon ensuring the availability of the Enterprise Manager system itself. See "EM Operational Considerations and Troubleshooting Whitepaper Master Index" in My Oracle Support Note (Doc ID 1940179.1 for best practices for configuring, operating, and diagnosing Enterprise Manager to ensure the availability of the Enterprise Manager system.

> **See Also:** "EM Operational Considerations and Troubleshooting Whitepaper Master Index" in My Oracle Support Note (Doc ID 1940179.1

Configure Metrics and Incident Rule Sets

Enterprise Manager takes a comprehensive approach to monitoring, management, and resolution of issues, organizing the process into three levels: event management, incident management, and problem management.

- An event is something that happens on a managed target, often indicating something abnormal has occurred. Metric alerts, availability alerts, compliance violations, and job events are examples of events.
- An incident is composed of one or more significant events that need to be managed together because of the potential impact of the event(s) to the business. Incident resolution is focused on mitigating the business impact. Incidents are created automatically or manually and are managed via the Incident Manager.
- A problem is the root cause of an incident. Problem resolution is focused on resolving the root cause. Administrators use Support Workbench to open a Service Request (SR) for the problem with Oracle Support using the details from the Automatic Diagnostic Repository (ADR), and manage the problem via the Incident Manager.

Incident rule sets and incident rules provide the means to automate actions taken by Enterprise Manager with respect to events, incidents, and problems. There are two types of rule sets: enterprise rule sets and private rule sets. Enterprise rule sets provide a complete set of actions, while private rule sets can only send e-mail notifications to their owners. Enterprise rule sets are evaluated in order, may be evaluated in multiple passes, and are evaluated before private rule sets. The order is important as only one incident will be created for an event, so the first rule that matches the event will be the one that creates the incident. Each matching rule with a workflow or notification action will execute, so if there are multiple matching rules for an event, the last one that executes will determine the final value of the priority, assignment, etc., and administrators may receive multiple notifications for the same event. Rule sets and rules can be enabled and disabled.

Metric alerts and availability alerts are types of events generated by a combination of factors and are defined on specific metrics. A metric is a data point sampled by a Management Agent and sent to the Oracle Management Repository to determine the health of a target. An availability alert could be an evaluation of the availability of a component through a simple heartbeat test. A metric alert could be an evaluation of a specific performance measurement such as "disk busy" or percentage of processes waiting for a specific wait event.

There are four states that can be checked for any metric: error, warning, critical, and clear. The administrator must make policy decisions such as:

- What objects should be monitored (databases, nodes, listeners, or other services)?
- What instrumentation should be sampled (such as availability, CPU percent busy)?
- How frequently should the metric be sampled?
- What should be done when the metric exceeds a predefined threshold?

All of these decisions are predicated on the business needs of the system. For example, all components might be monitored for availability, but some systems might be monitored only during business hours. Systems with specific performance problems can have additional performance tracing enabled to debug a problem.

Incident Rules can take action on events such as metric alerts and can be defined to operate on all targets, targets of a specific type, targets in a specific group, or individual targets. For example, an administrator can create an incident rule that monitors the availability of database targets and generates an e-mail message if a database fails. After that incident rule is enabled, it is evaluated for all existing databases and any database created in the future. Access these rules by navigating to **Setup**, selecting Incidents, and then choosing **Incident Rules**.

The rules monitor issues that require immediate attention, such as those that can affect service availability, and Oracle or application errors. Service availability can be affected by an outage in any layer of the application stack: node, database, listener, and critical application data. A service availability failure, such as the inability to connect to the database, or the inability to access data critical to the functionality of the application, must be identified, reported, and reacted to quickly. Potential service outages such as a full archive log directory also must be addressed correctly to avoid a system outage.

Enterprise Manager provides a default incident rule set that provides a strong starting framework for monitoring availability. While you cannot modify Oracle's default incident rule sets, you can create your own copy of the out of box incident rule set and modify the rules to conform to the policies of each individual site. You can also create incident rule sets consisting of incident rules for site-specific targets or applications. Additionally, you can configure notification schedules to notify users during specific time periods to create an automated coverage policy.

Implement a strategy for managing incident rules using four logical categories:

- High Availability

  Create enterprise rule set(s) as applicable that focus on the availability of mission critical systems. Notify administrators when warning and critical thresholds are exceeded on the metrics that you identify. Locate the rule set(s) above the Oracle out of box rule sets so that any incident creation that you specify in the rule set(s) takes precedence.

- Key Performance Indicators

Create enterprise rule set(s) as applicable that focus on the key performance indicators that are representative of the performance and throughput of the overall system. Notify administrators when warning and critical thresholds are exceeded on the metrics that you identify. Monitor to ensure required service level is achieved. Locate the rule set(s) above the Oracle out of box rule sets and below the High Availability rule set(s) so that any incident creation that you specify in the rule set(s) will occur if not created by High Availability rule set(s) and will take precedence over the out of box rule sets.

- ADR Incidents

  Make use of the ADR Incidents that are automatically generated by Oracle out of box. Incorporate notifications for these incidents into the rule sets in above categories as appropriate to notify administrators of relevant incidents and operational errors. Enable DB Alert Log metrics if/as necessary to address any additional required error monitoring.

- Administrator's Choice

  Each administrator creates private rule set(s) managed by the individual administrator, deciding on what is of most interest to them. Focus on providing more details and insight to ensure successful administration and operations, addressing deeper investigation, longer term, or lower priority concerns that would not warrant waking an administrator in the middle of the night. These rules may change more frequently.

Consider the following example. The Fast Recovery Area (FRA) is filling up on a database that is part of a mission critical system. Having sufficient space for the FRA is critical to availability. The database is associated with a group of targets that are associated with a high availability rule set. The rule set includes a rule that evaluates events of type Metric Alert for the Recovery Area Free Space (%) metric and sends e-mail notifications to administrators. The rule is configured to evaluate whether the severity is warning or critical, and sends an appropriate notification via the appropriate mechanism. When Enterprise Manager detects that the FRA crosses the warning threshold, Enterprise Manager raises an event of type Metric Alert. The rules in the high availability rule set are evaluated. When Enterprise Manager processes the rule for the FRA, it determines that the severity of the Metric Alert event is warning, and sends an email to the administrators on their standard email accounts because an action has been specified to send an email to notify administrators via their standard email address when the FRA crosses the warning threshold. Had the FRA crossed the critical threshold, a rule action would have sent an email to the administrators' pager addresses. Unlike most metrics, the Recovery Area Free Space (%) warning and critical thresholds cannot be edited. You can create a Metric Extension if different thresholds are required.

Use the following best practices:

- When creating enterprise rule sets, place those above the out of the box Oracle incident rule set so that your custom rule sets fire first.
- Use monitoring templates to configure metrics for each target type. This allows standardization of metrics for similar targets in environments. You may have different templates for the same target types in different environments (such as development and production) and within different applications or hardware configurations (such as where thresholds would be different due to storage space or processing differences).
- Use administration groups and template collections to ensure that monitoring templates are automatically applied to targets added to the administration groups. This removes the need to manually apply the monitoring templates to the targets.

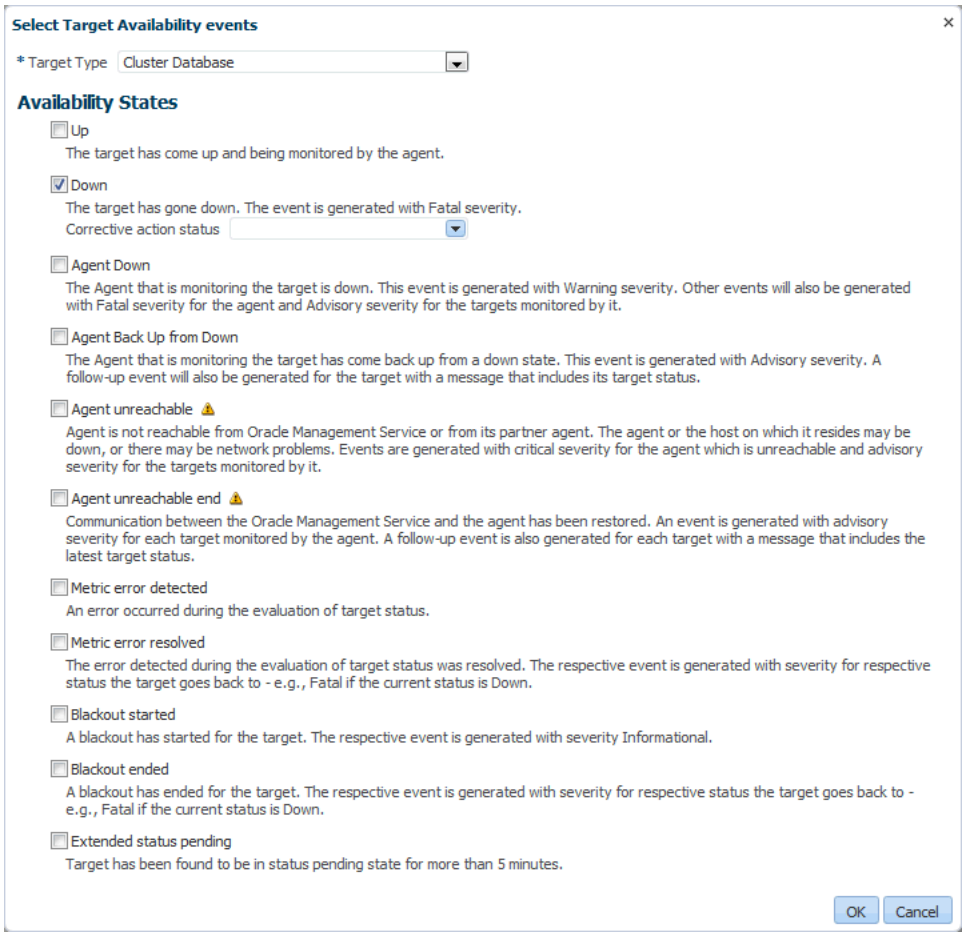
- Ensure that metric thresholds are set appropriately for each target type in each environment to suit your availability requirements. Consider setting thresholds for the metrics in Table 11–1, Table 11–2, Table 11-4, and Table 11-5. The frequency of the monitoring is determined by the service-level agreement (SLA) for each component.
- Create incident rules that notify administrators appropriately for each environment based on the key metric alert and target availability events.
- Use Beacon functionality to track the performance of individual applications. A Beacon can be set to perform a user transaction representative of normal application work. Enterprise Manager can then break down the response time of that transaction into its component pieces for analysis. In addition, an alert can be triggered if the execution time of that transaction exceeds a predefined limit.
- Configure multiple e-mail addresses for administrators who have pagers, cell phones, or mobile devices that can receive messages via e-mail. Use the Email Type of Pager with the email addresses for these devices and the Email Type of Email for standard email addresses. Within Incident Rules, differentiate between critical and warning messages, adding administrators to the Page field to send notifications for critical alerts to email addresses associated with these devices to ensure critical notifications are received as quickly as possible, and adding administrators to the Email To field to send notification for warning events to the regular e-mail address.
- Add Notification Methods and use them in each Incident Rule when notifications are sent. By default, the easiest method for alerting an administrator to a potential problem is to send e-mail as discussed previously. Supplement this notification method by adding a callout to an SNMP trap or operating system script that sends an alert by some method other than e-mail. This avoids problems that might occur if a component of the e-mail system fails. Set additional Notification Methods by using the **Setup** link at the top of any Enterprise Manager page and selecting Notification Methods for scripts and SNMPv1 Traps, or selecting SNMPv3 Traps.
- Create or modify Incident Rules to notify the administrator when there are errors in computing target availability. Select "Metric error detected" in the Availability States for a Target Type on the Select Target Availability Events form when editing or creating a rule of type Target Availability. This might generate a false positive reading on the availability of the component, but it ensures the highest level of notification to system administrators. See Figure 11-2 Setting Incident Rules for Availability for an example of the Select Target Availability events form where this option is available to be selected.

See Also:

- *Oracle Enterprise Manager Cloud Control Administrator's Guide* for information about Monitoring and using Metrics
- *Oracle Enterprise Manager Cloud Control Administrator's Guide* for information about Incident Management including Events, Incidents, Problems, Rule Sets, and Rules
- For more information about Rule Sets and Incident Rules in Enterprise Manager 12c, see "12c Cloud Control: What are Incident Management Rule Sets and Incident Rules?" in My Oracle Support Note (Doc ID) 1556225.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1556225.1
- *Oracle Enterprise Manager Cloud Control Administrator's Guide* for information about Notifications
- *Oracle Enterprise Manager Cloud Control Administrator's Guide* for information about Administration Groups
- *Oracle Enterprise Manager Cloud Control Administrator's Guide* for information about Monitoring Templates
- *Oracle Enterprise Manager Cloud Control Administrator's Guide* for information about configuring Service Tests and Beacons

Figure 11–2 shows the Select Target Availability events page for choosing availability states for a Cluster Database, with the Down option chosen.

Figure 11–2 Setting Incident Rules for Availability



Use the metrics shown in Table 11–1 to monitor space management conditions that have the potential to cause a service outage.

Table 11–1 Recommendations for Monitoring Space

Metric	Recommendation
Tablespace Space Used (%)	Set this database-level metric to check the Available Space Used (%) for each tablespace. For cluster databases, this metric is monitored at the cluster database target level and not by member instances. This metric enables the administrator to choose the threshold percentages that Enterprise Manager tests against, and the number of samples that must occur in error before a message is generated and sent to the administrator. If the percentage of used space is greater than the values specified in the threshold arguments, then a warning or critical alert is generated. The recommended default settings are 85% for a warning and 97% for a critical space usage threshold, but you should adjust these values appropriately, depending on system usage. Also, you can customize this metric to monitor specific tablespaces. Note: there is an Enterprise Manager Job in the Job Library named: `DISABLE TABLESPACE USED (%) ALERTS FOR UNDO AND TEMP TABLESPACES` Use this Job to disable alerts for all `UNDO` and `TEMP` tablespaces. This job is useful if you encounter too many alerts on `TEMP` and `UNDO` tablespaces. Beginning with Database Plugin 12.1.0.6, monitoring of TEMP and UNDO tablespaces has been separated into two new metrics: Tablespace Space Used (%) (Temp) and Tablespace Space Used (%) (Undo). For more information, see Testing and Troubleshooting the 'Tablespace Space Full (%)' metric in Enterprise Manager 12.1.0.4 in My Oracle Support Note (Doc ID) 1927636.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1927636.1.
Dump Area Used (%)	Set this metric to monitor the dump directory destinations. Dump space must be available so that the maximum amount of diagnostic information is saved the first time an error occurs. The recommended settings are 70% for a warning and 90% for a critical threshold error, but these should be adjusted depending on system usage. Set this metric in the Dump Area metric group.
Recovery Area Free Space (%)	This is a database-level metric that is evaluated by the server every 15 minutes or during a file creation, whichever occurs first. The metric is also printed in the alert log. For cluster databases, this metric is monitored at the cluster database target level and not by member instances. The Critical Threshold is set for < 3% and the Warning Threshold is set for < 15%. You cannot customize these thresholds. An alert is returned the first time the alert occurs, and the alert is not cleared until the available space rises above 15%.
File system Space Available(%)	By default, this metric monitors file systems on the host using the default warning threshold of 20% and the critical threshold of 5%. Thresholds can be set separately by file system.
Archive Area Used (%)	Set this metric to return the percentage of space used on the archive area destinations. Thresholds can be set separately for each Archive Area Destination. If the space used is more than the threshold value given in the threshold arguments, then a warning or critical alert is generated. If monitoring a cluster database that does not use a shared archive area, disable this metric at the cluster database level and enable this metric at the database instance level. If the database is not running in `ARCHIVELOG` mode, this metric fails to register. The default warning threshold is 80%, but consider using 70% full to send a warning or 90% for the critical threshold. If the database is configured to archive to the Fast Recovery Area, this metric is not applicable. Instead, use the Recovery Area Free Space (%) metric to monitor the Fast Recovery Area.

In Enterprise Manager 12*c* the mechanism for monitoring the Database Alert Log is tightly integrated with the Support Workbench, with the benefits of being able to

generate packages for each problem or incident reported and quickly upload them to support.

As part of integrating with the Support Workbench, errors are categorized into different classes and groups, each served by a separate metric. At the highest level of categorization there are two different classes of errors: incidents and operational errors.

- Incidents are errors that are recorded in the database alert log file, which signify that the database being monitored has detected a critical error condition. For example a critical error condition could be a generic internal error or an access violation.
- Operational Errors are errors that are recorded in the database alert log file, which signify that the database being monitored has detected an error that may affect the operation of the database. For example, an operational error could be an indication that the archiver is hung or a media failure.

Enterprise Manager automatically creates critical events for these incidents and operational errors. If these events are insufficient for your monitoring needs, you can supplement these by using one of two categories of metrics to configure warning and critical thresholds for a set of additional metrics that use the old pre-11g approach which monitors the text alert log. To do this, enable the disabled DB Alert Log metrics group metrics beginning with Database Plugin 12.1.0.4, or the disabled Alert Log metrics group metrics with earlier releases.

When using the DB Alert Log or Alert Log metrics, administrators can maintain the Alert Log Filter Expression in order to prevent certain errors that the administrator determines should be ignored from raising Metric Alert events in Enterprise Manager. To do this, edit the Alert Log Filter Expression, which is accessible by navigating to the Database target menu->Monitoring->Metrics and Collection Settings, and then selecting the edit icon to the right of Generic Alert Log Error under either DB Alert Log or Alert Log, depending upon which approach is being used.

Note: For more information about Alert Log Monitoring in Enterprise Manager 12c, see "Database Alert log monitoring in 12c explained" in My Oracle Support Note 1538482.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1538482.1

For details about the changes to alert log monitoring in Database Plugin 12.1.0.4, see " Changes to Alert Log Monitoring in Database Plugin 12.1.0.4" in My Oracle Support Note (Doc ID) 1587020.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1587020.1

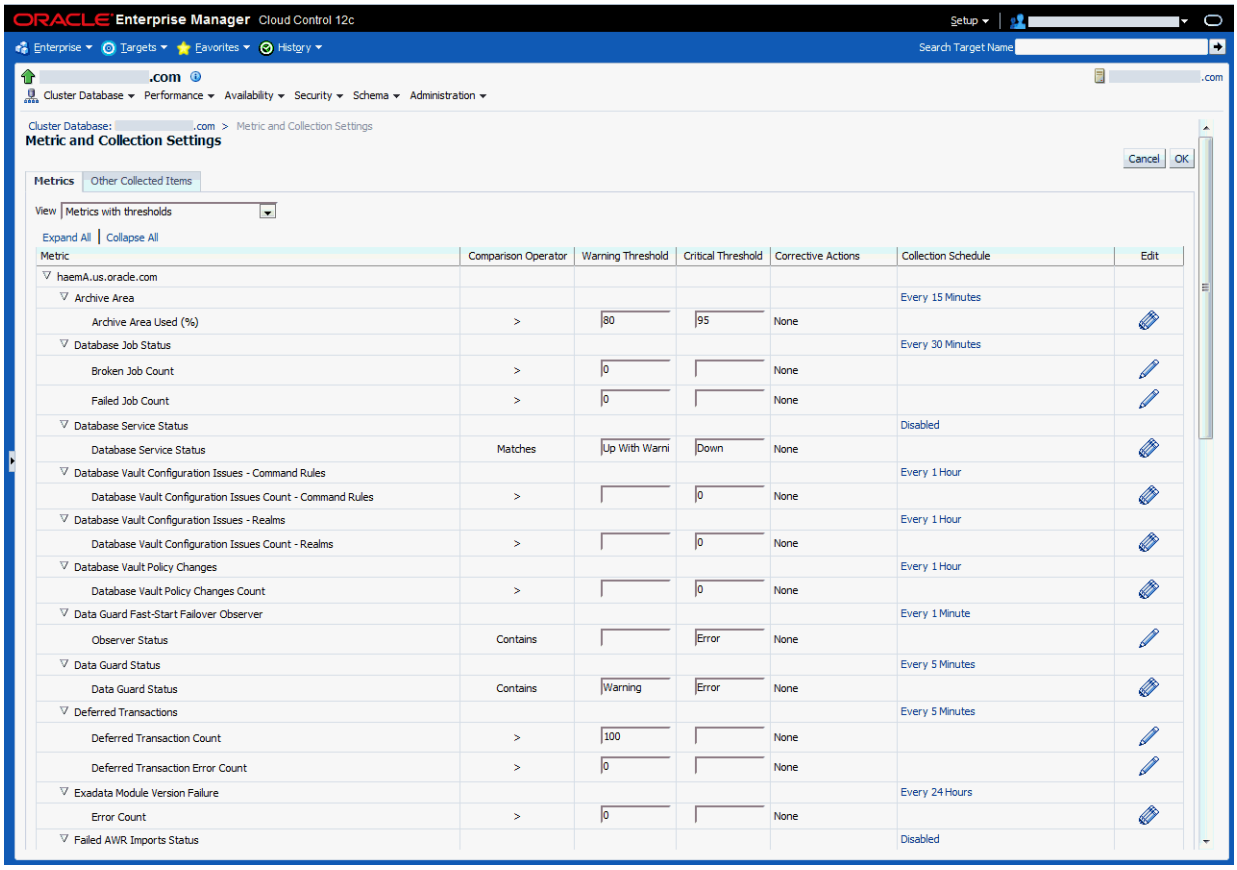
Monitor the system to ensure that the processing capacity is not exceeded. The warning and critical thresholds for these metrics should be modified based on the usage pattern of the system, following the recommendations in Table 11–2.

Table 11–2 Recommendations for Monitoring Processing Capacity

Metric	Recommendation
Process Limit Usage (%)	Set thresholds for this metric to warn if the number of current processes approaches the value of the `PROCESSES` initialization parameter.
Session Limit Usage (%)	Set thresholds for this metric to warn if the instance is approaching the maximum number of concurrent connections allowed by the database.

Figure 11–3 shows the Metric and Collection Settings page for setting and editing metrics. The documentation library contains complete reference information for every metric. To access reference information for a specific metric, use the documentation library search feature.

Figure 11–3 Metric and Collection Settings Page



See Also:

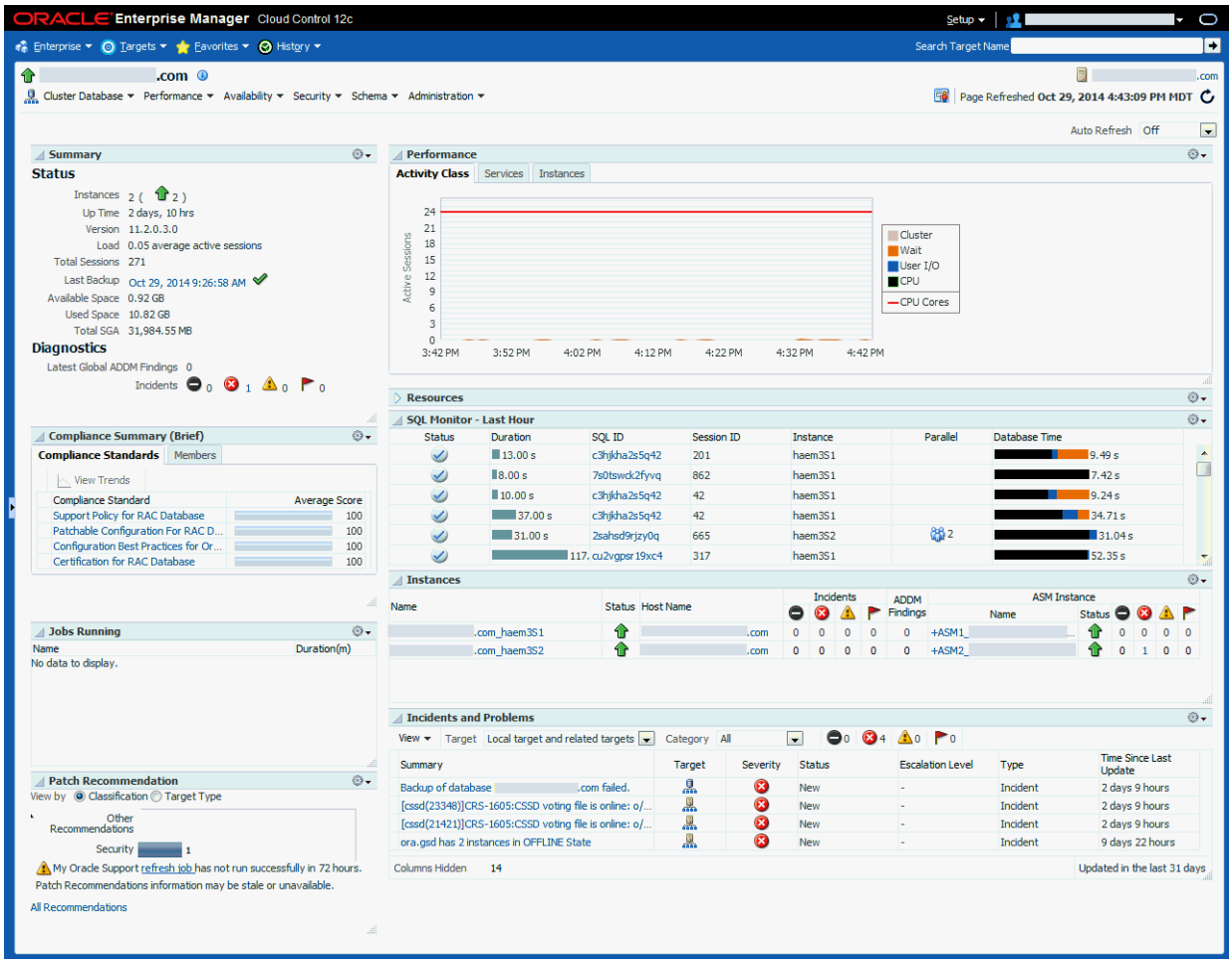
- *Oracle Database 2 Day + Performance Tuning Guide* for information about setting metric thresholds
- *Oracle Database Administrator's Guide* for information on Viewing Problems with the Support Workbench
- *Oracle Enterprise Manager Oracle Database Plug-in Metric Reference Manual* for information about available metrics

Use Database Target Views to Monitor Health, Availability, and Performance

The Database target Home page in Figure 11–4 shows system performance, space usage, and information important to availability such as the date, time, and status of the last backup with a link to a backup report.

You can see the most recent incidents and problems for the target under the Incidents and Problems table, as shown in Figure 11–4. You can access further information about Incidents and Problems by clicking the links in the Summary column.

Figure 11–4 Database Home Page



Performance Analysis and Performance Baseline

Many of the metrics for Database targets in Enterprise Manager pertain to performance. A system that is not meeting performance service-level agreements is not meeting High Availability system requirements. While performance problems seldom cause a major system outage, they can still cause an outage to a subset of customers. Outages of this type are commonly referred to as **application service brownouts**. The primary cause of brownouts is the intermittent or partial failure of one or more infrastructure components. IT managers must be aware of how the infrastructure components are performing (their response time, latency, and availability), and how they are affecting the quality of application service delivered to the end user.

A performance baseline, derived from normal operations that meet the service-level agreement should determine what constitutes a performance metric alert. Baseline

data should be collected from the first day that an application is in production and should include the following:

- Application statistics (transaction volumes, response time, web service times)
- Database statistics (transaction rate, redo rate, hit ratios, top 5 wait events, top 5 SQL transactions)
- Operating system statistics (CPU, memory, I/O, network)

You can use Enterprise Manager to capture a baseline snapshot of database performance and create an Automatic Workload Repository (AWR) baseline. Oracle recommends increasing the AWR retention period if practical in your environment; 30 days is a good starting point. Enterprise Manager compares these values against system performance and displays the result on the database Target page. Enterprise Manager can also send alerts if the values deviate too far from the established baseline. See "Use Automatic Performance Tuning Features" on page 4-13 for more information about Automatic Workload Repository.

There is no formula that can be used to set specific thresholds for a given customer system. Thresholds should be determined using values that are some percentage above the normal operating value of the metric, accounting for the variability of the value. Depending upon the variability, one starting point could be to take the baseline value, multiply by 1.15 to set the warning threshold and multiply by 1.25 to set the critical threshold. As each system and value is different, the key is to understand the system and the performance data associated with a baseline during normal operations, set candidate values, and then monitor and adjust as necessary. System baselines and thresholds need to be reevaluated periodically as the behavior of the system can change over time for a variety of reasons including changes in usage patterns, system volume, and hardware and software updates.

Set thresholds as appropriate for the metrics listed in Table 11–3 for all database targets and incorporate into incident rules to provide notifications as needed.

Table 11–3 Recommendations for Performance Related Metrics

Metric	Level	Recommendation
I/O Requests (per second)	Instance	This metric represents the total rate of I/O read and write requests for the database. It sends an alert when the number of operations exceeds a user-defined threshold. Use this metric with operating system-level metrics that are also available with Enterprise Manager. Set this metric based on the total I/O throughput available to the system, the number of I/O channels available, network bandwidth (in a SAN environment), the effects of the disk cache if you are using a storage array device, and the maximum I/O rate and number of spindles available to the database.
Database CPU Time (%)	Instance	This metric represents the percentage of database call time that is spent on the CPU. It can be used to detect a change in the operation of a system, for example, a drop in Database CPU time from 50% to 25%. The `Consecutive Number of Occurrences Preceding Notification` column indicates the consecutive number of times the comparison against thresholds should hold `TRUE` before an alert is generated. This usage might be normal at peak periods, but it might also be an indication of a runaway process or of a potential resource shortage.
Wait Time (%)	Instance	Excessive idle time indicates that a bottleneck for one or more resources is occurring. Set this instance-level metric based on the system wait time when the application is performing as expected.

Table 11–3 (Cont.) Recommendations for Performance Related Metrics

Metric	Level	Recommendation
Network Bytes (per second)	Instance	This metric reports network traffic that Oracle generates. This metric can indicate a potential network bottleneck. Set this metric based on actual usage during peak periods.
Pages Paged-in (per second)	Host	For UNIX-based systems, represents the number of pages paged in (read from disk to resolve fault memory references) per second. This metric checks the number of pages paged in for the CPU(s) specified by the Host CPU(s) parameter, such as `cpu_stat0` or `*` (for all CPUs on the system). For Microsoft Windows, this metric is the rate at which pages are read from disk to resolve hard page faults. Hard page faults occur when a process refers to a page in virtual memory that is not in its working set or elsewhere in physical memory, and must be retrieved from disk. When a page is faulted, the system tries to read multiple contiguous pages into memory to maximize the benefit of the read operation.
Run Queue Length	Host	For UNIX-based systems, the Run Queue Length metrics represent the average number of processes in memory and subject to be run in the last interval (1 minute average, 5 minute average, and 15 minute average). It is recommended to alert when Run Queue Length = # of CPUs. (An alternative way to do this is to monitor the Load Average metric and compare it to Maximum CPU.) This metric is not available on Microsoft Windows.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about performance monitoring
- *Oracle Database 2 Day DBA* for more information about monitoring and tuning using Enterprise Manager
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring and tuning using Enterprise Manager

Use Metrics to Monitor Data Guard System Availability

Set thresholds for Enterprise Manager metrics to monitor the availability of Data Guard configurations and incorporate into incident rules to provide notifications. As discussed regarding performance related metrics in the previous section, set the thresholds where applicable using values that are some percentage above normal operating values after monitoring the system and understanding what is normal in your environment. Consider your business Recovery Time Objective (RTO), Recovery Point Objective (RPO), and requirements of your Service Level Agreement(s) (SLA) as part of the thresholds. Reevaluate the thresholds over time to ensure they remain appropriate. Table 11–4 shows metrics that are available for monitoring Data Guard databases.

Table 11–4 Recommendations for Setting Data Guard Metrics

Metric	Recommendation
Data Guard Status	Notifies you about system problems in a Data Guard configuration.
Apply Lag	Displays (in seconds) how far the standby is behind the primary database. This metric generates an alert on the standby database if it falls behind more than the user-specified threshold (if any). Set warning and critical thresholds based upon RTO and SLA considerations.
Estimated Failover Time	Displays the approximate number of seconds required to failover to this standby database. Set warning and critical thresholds based upon RTO and SLA considerations.
Redo Apply Rate	Displays the Redo Apply rate in KB/second on this standby database. Set warning and critical thresholds to identify deviation from normal operations.
Transport Lag	Displays the approximate number of seconds of redo that is not yet available on this standby database. The lag may be because the redo data has not yet been transported or there may be a gap. This metric generates an alert on the standby database if it falls behind more than the user-specified threshold (if any). Set warning and critical thresholds based upon RPO considerations.

Managing the High Availability Environment with Enterprise Manager

Use Enterprise Manager as a proactive part of administering any system and for problem notification and analysis, with the following recommendations:

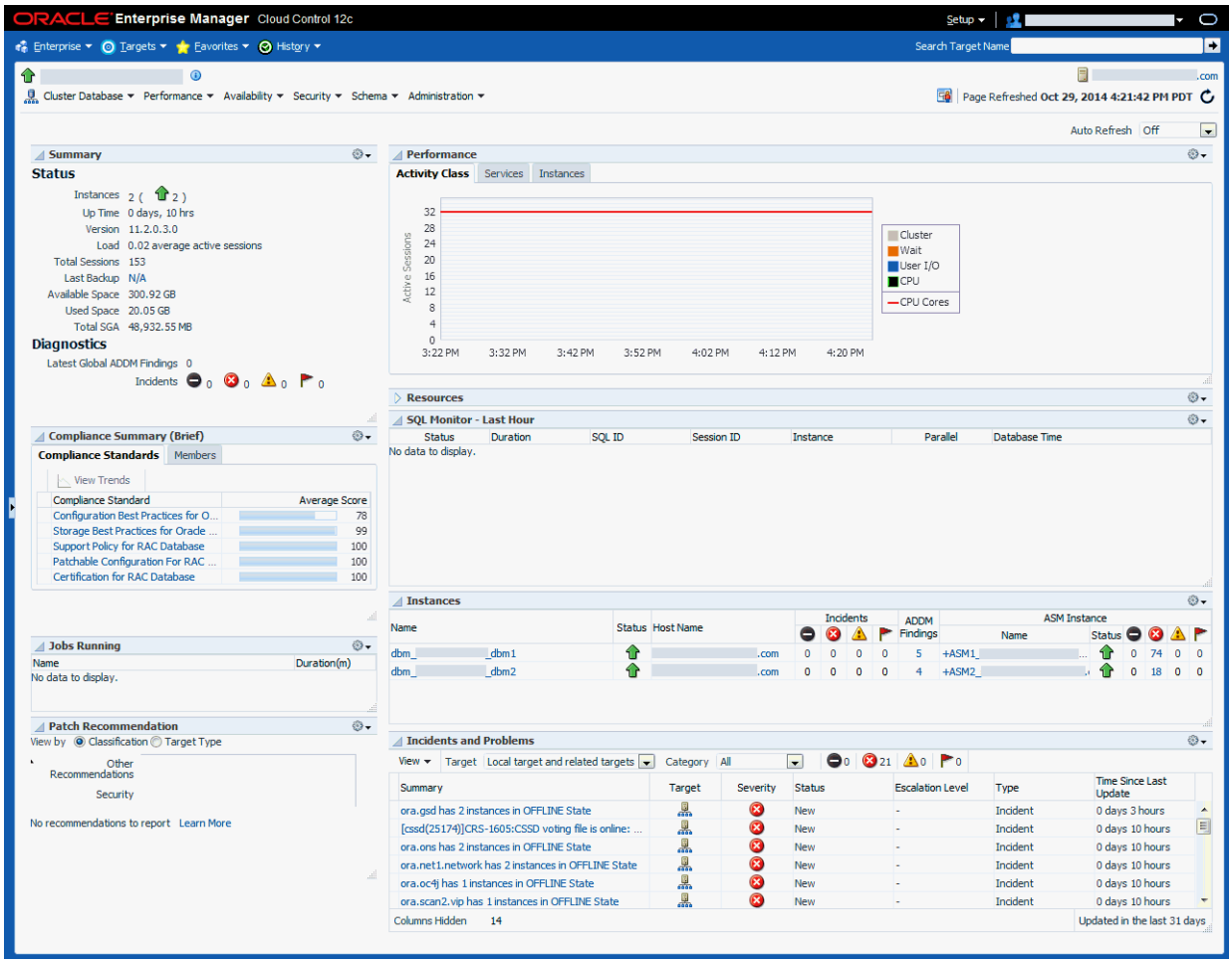
- Check Enterprise Manager Compliance Results
- Use Enterprise Manager to Manage Oracle Patches and Maintain System Baselines
- Manage Database Availability with the High Availability Console
- Configure High Availability Solutions with MAA Advisor

Check Enterprise Manager Compliance Results

Enterprise Manager includes a compliance management framework that provides automatic tracking and reporting on how well managed targets conform to standards. These standards can include industry, Oracle, and internal standards. Enterprise Manager comes with a pre-installed set of compliance standards for Oracle hardware and software including Database, Exadata Database Machine, Fusion Middleware, and more. These standards include recommendations of best practices for all databases. User-defined compliance frameworks and compliance standards can be created, either by creating like an existing framework or standard or by creating a brand new framework or standard.

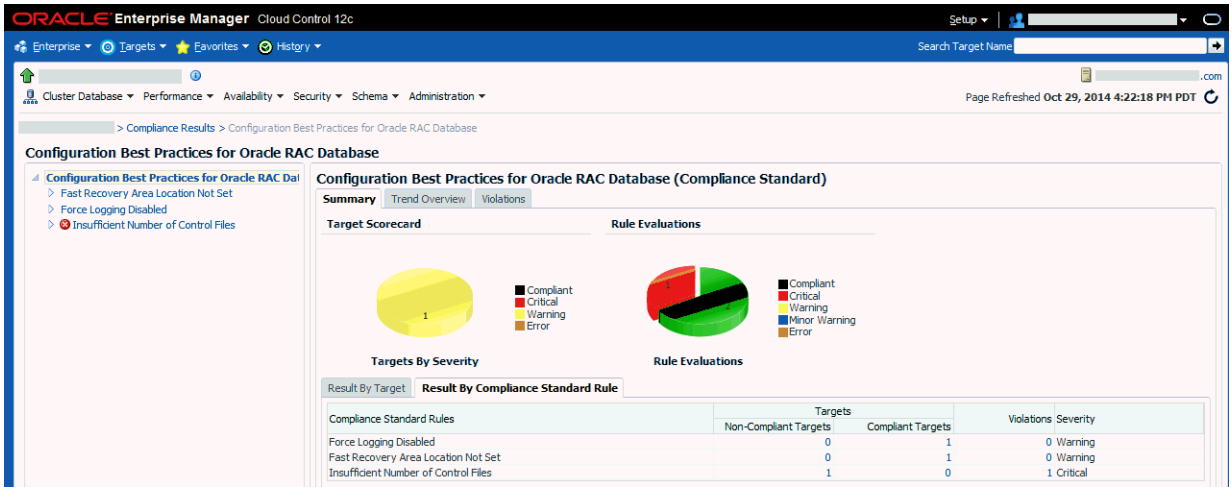
To enable standards, navigate to the Compliance Library and associate the desired standards with the appropriate targets. In order to make use of some standards, such as security standards, templates must be applied to the targets first in order to collect the required information. For more details on which Oracle database compliance standards require which Oracle provided monitoring templates, see Oracle Enterprise Manager Cloud Control Oracle Database Compliance Standards. Compliance results are displayed on the targets' home pages in the **Compliance Summary** area, as shown in Figure 11–5.

Figure 11–5 Database Home Page with Compliance Summary



To see more details on Compliance Standard results, select a link in the **Compliance Summary** area. Figure 11–6 shows the Compliance Results page for the target and the selected Compliance Standard.

Figure 11–6 Database Target Compliance Results Page



To see the results of all Compliance Standards, select **Compliance** then Results from the **Enterprise** menu, as shown in Figure 11–7.

Figure 11–7 Compliance Results Page

See Also:

- *Oracle Enterprise Manager Lifecycle Management Administrator's Guide* for more details on Managing Compliance
- *Oracle Enterprise Manager Cloud Control Oracle Database Compliance Standards* for more details on Compliance Standards

Use Enterprise Manager to Manage Oracle Patches and Maintain System Baselines

Oracle recommends that production environments stay current by implementing recommended patches within six months of their release. Enterprise Manager 12c provides a patch management solution that maximizes ease of use and minimizes patching related downtime, helping administrators meet those recommendations. For the most current details on patch list recommendations for your environment, see My Oracle Support (MOS).

An online mode integrates the patch workflow with MOS, providing a consistent interface that enables Oracle patch recommendations, manual patch searches, access to MOS resources such as knowledge articles and service requests, and automatic resolution of patch conflicts using merge patches. An offline mode supports environments where Enterprise Manager cannot connect to MOS, using patches manually uploaded to the Software Library. Administrators can access MOS on systems where internet access is available, download patches to their local host, and upload the patches to the Software Library either via Enterprise Manager or through the use of the Enterprise Manager Command Line Interface (emcli).

Patch plans provide end-to-end orchestration of the patching workflow, enabling administrators to prepare, validate, and apply a list of patches as a group to one or more targets. Patch plans support one-off patches, including interim patches, diagnostic patches, Patch Set Updates (PSU), and Critical Patch Updates (CPU). Patch plans also support patch sets for some targets.

Patch plans can be tested and saved as patch templates, making it easier for multiple administrators to deploy patches consistently across multiple environments by creating their patch plans from the patch template instead of having to create patch plans from scratch.

Enterprise Manager's patch management solution supports both in-place patching, where the Oracle home is directly patched, and out-of-place patching, where the

existing Oracle home is cloned and the cloned home is patched. Use out-of-place patching where supported to minimize downtime.

For some targets, including Oracle RAC, Oracle Grid Infrastructure, and Oracle Data Guard, the solution also supports patching in rolling mode, where nodes are shut down, patched, and restarted one by one, and parallel mode, where all of the nodes are shut down and the patch is applied on all nodes at the same time. Use rolling mode patching where supported to minimize downtime.

You can examine patch levels for one system and compare them between systems in either a one-to-one or one-to-many relationship. In this case, a system can be identified as a baseline and used to demonstrate maintenance requirements in other systems. This can be done for operating system patches and database patches.

See Also:

- *Oracle Enterprise Manager Lifecycle Management Administrator's Guide* for information about Patching Software Deployments
- *Oracle Enterprise Manager Lifecycle Management Administrator's Guide* for information about Managing Configuration Information
- "Eliminating or Reducing Downtime for Scheduled Outages"

Manage Database Availability with the High Availability Console

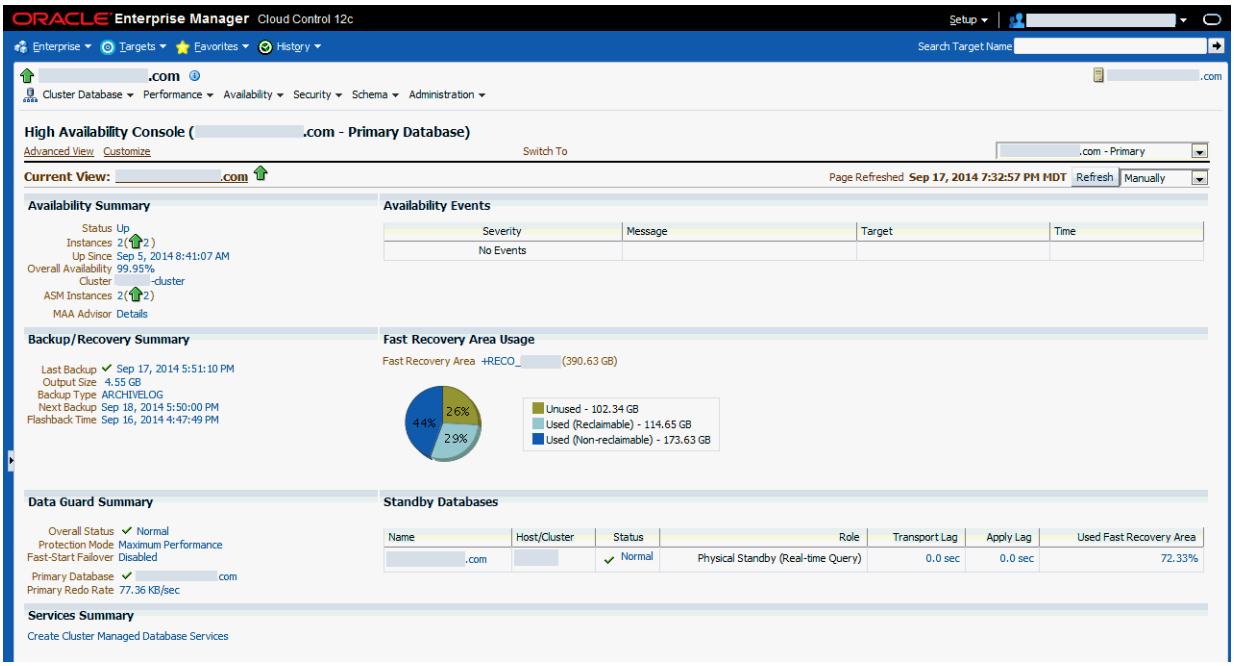
The High Availability (HA) Console is a one stop, dashboard-style page for monitoring the availability of each database. You can use it on any database and if a database is part of a Data Guard configuration, the HA Console allows you to switch your view from the primary database to any of the standby databases.

Use the HA Console to:

- Display high availability events including events from related targets such as standby databases
- View the high availability summary that includes the status of the database
- View the last backup status
- View the Fast Recovery Area Usage, if configured
- If Oracle Data Guard is configured: View the Data Guard summary, set up Data Guard standby databases for any database target, manage switchover and failover of database targets other than the database that contains the Management Repository, and monitor the health of a Data Guard configuration at a glance
- If Oracle RAC is configured: View the Oracle RAC Services summary including Top Services

Figure 11–8 shows the HA Console. This figure shows summary information, details, and historical statistics for the primary database and shows the standby databases for the primary target, various Data Guard standby performance metrics and settings, and the data protection mode.

Figure 11–8 Monitoring a Primary Database in the High Availability Console



In Figure 11–8, the Availability Summary shows that the primary database is up and its availability is currently 99.95%. The Availability Summary also shows the status of Oracle ASM instances. The Availability Events table shows specific high availability events (alerts). You can click the message to obtain more details (or to suppress the event). To set up, manage, and configure a specific solution area for this database, under Availability Summary, next to **MAA Advisor**, click **Details** to go to the Maximum Availability Architecture (MAA) Advisor page (described in more detail in Section , "Configure High Availability Solutions with MAA Advisor").

The **Backup/Recovery Summary** area displays the Last Backup and Next Backup information, including times for both and status, size, and type of the last backup. The area also includes the Flashback Time to which the database can be reset if flashback database is enabled. The Fast Recovery Area Usage area displays information about the fast recovery area. The chart indicates about 73% of the fast recovery area is currently used. You can click the chart to display the page with the metric details.

The **Data Guard Summary** area shows the primary database is running in Maximum Performance mode and has Fast-Start Failover disabled. You can click the link next to **Protection Mode** to modify the data protection mode. In the Standby Databases table, the physical standby database is caught up with the primary database (Apply/Transport Lag) metrics are showing 0 seconds, and the Used Fast Recovery Area (FRA) is 72.33%. Click on the value next to the Primary Redo Rate to view a chart that shows the redo trend. Note that if Data Guard is not configured, the "Switch To" box in the corner of the console is not displayed.

Figure 11–9 shows information similar to figure Figure 11–8, but for the standby database, which is a physical standby database running real-time query. In the Standby Databases table, the Apply/Transport Lag metrics indicate that the physical standby database is caught up with the primary database, and the Used Fast Recovery Area (FRA) is 72%. Note that if Data Guard is not configured, the "Switch To" box in the corner of the console is not displayed.

Figure 11–9 Monitoring the Standby Database in the High Availability Console

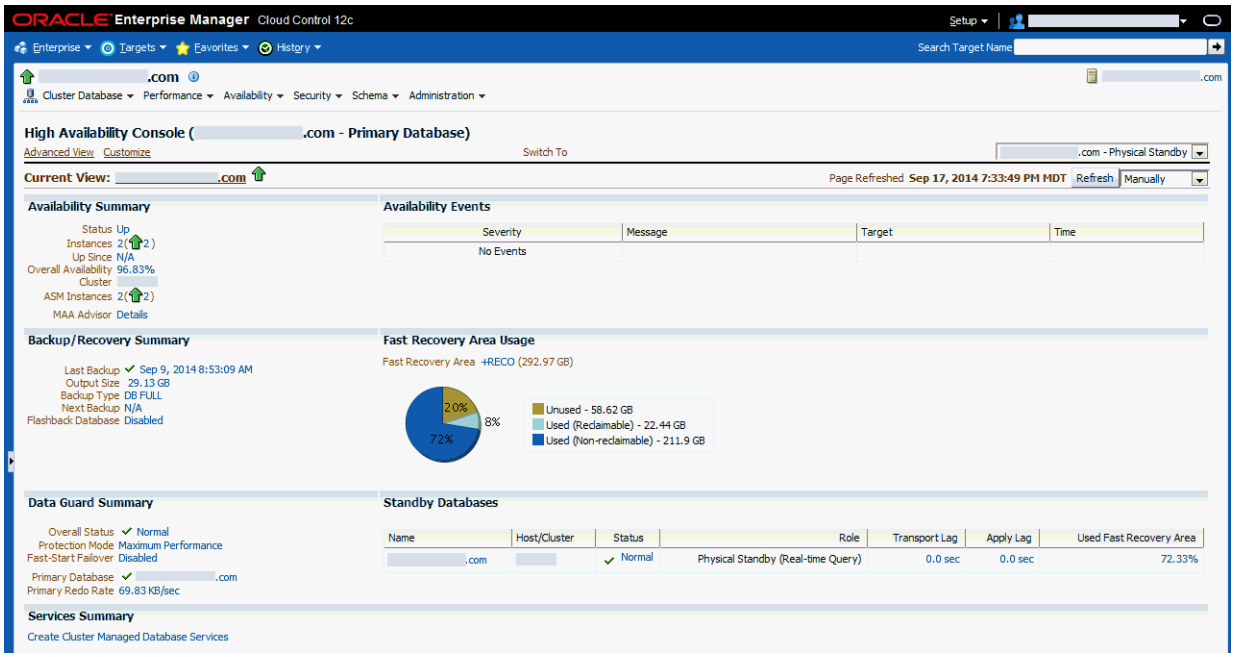
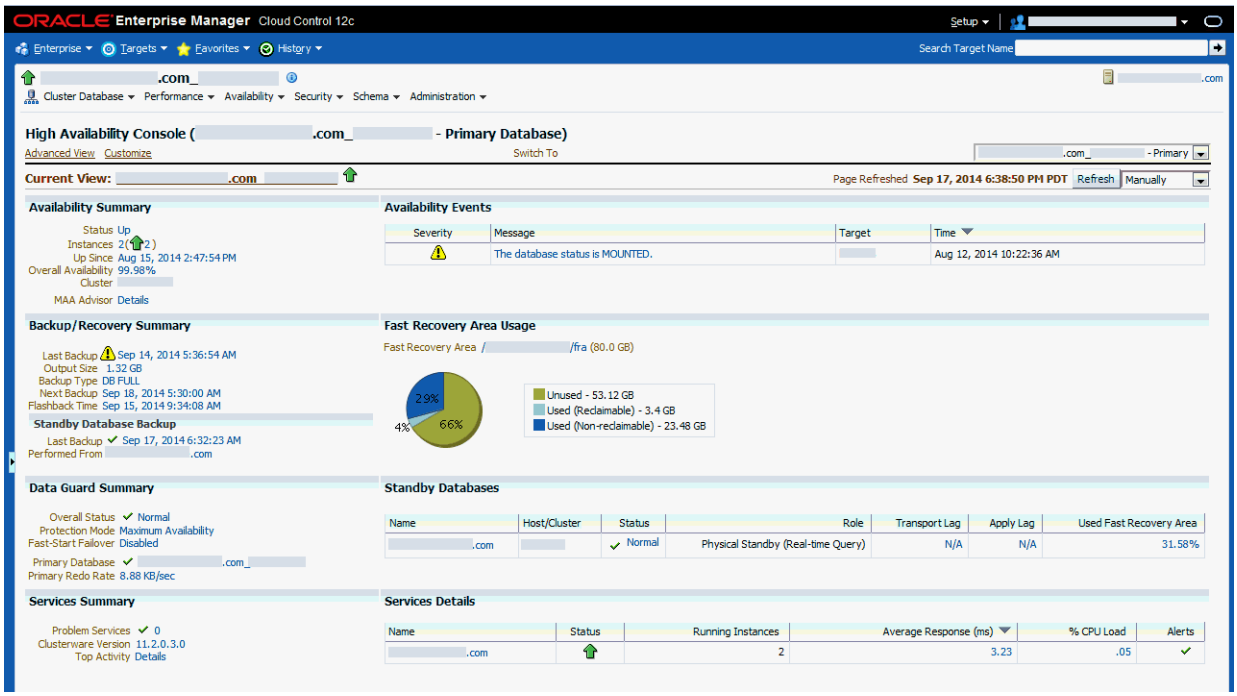


Figure 11–10 shows sample values for **Services Summary** and **Services Details** areas. These areas show summary and detail information about Oracle RAC Services, including links to details on top activity and problem services.

Figure 11–10 Monitoring the Cluster in the High Availability Console Showing Services



See Also: *Oracle Enterprise Manager Cloud Control Introduction* for information about Database Management

Configure High Availability Solutions with MAA Advisor

The goal of the MAA Advisor is to help you implement Oracle's best practices to achieve the optimal high availability architecture.

From the Availability Summary section on the High Availability Console, you can link to the MAA Advisor to:

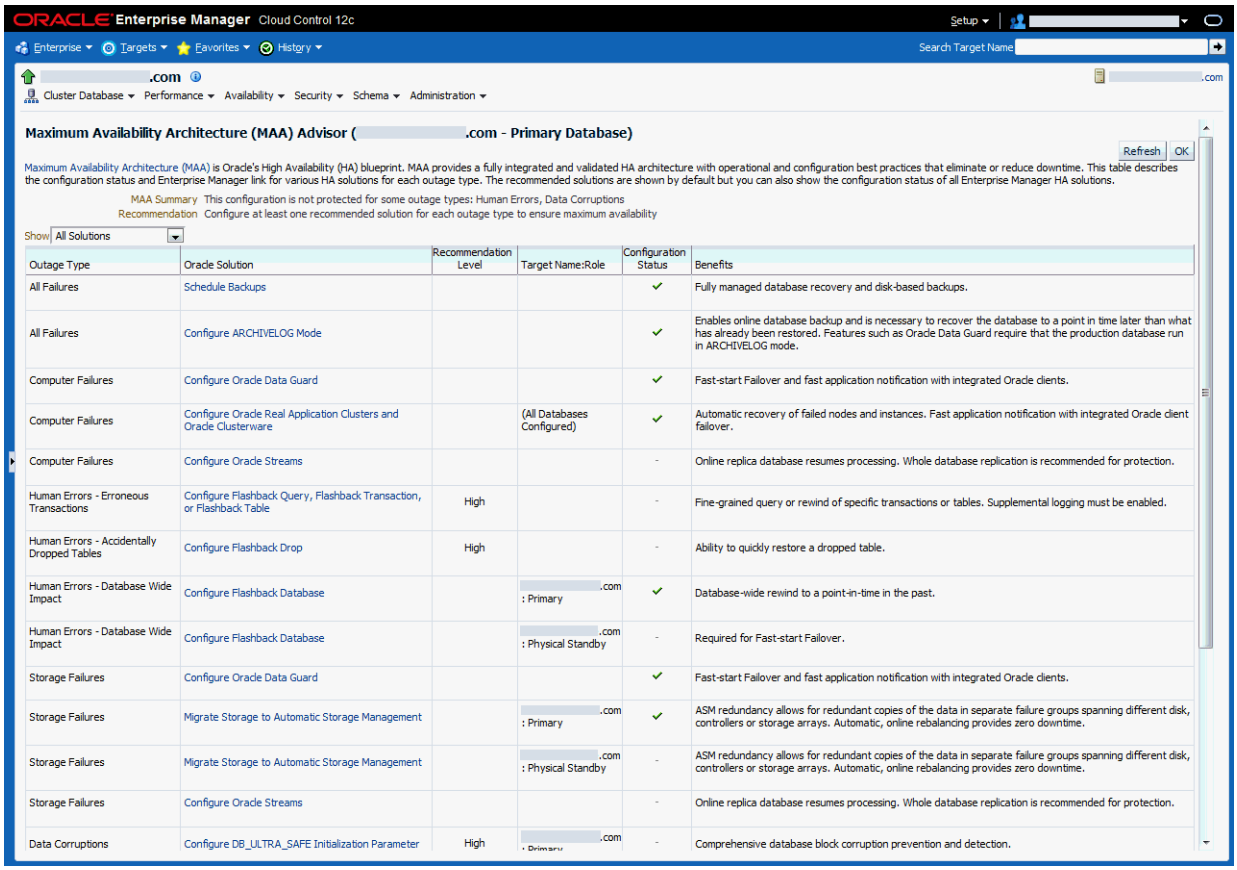
- View recommended Oracle solutions for each outage type (site failures, computer failures, storage failures, human errors, and data corruptions)
- View the configuration status and use the links in the Oracle Solution column to go to the Enterprise Manager page where the solution can be configured.
- Understand the benefits of each solution
- Link to the MAA website for white papers, documentation, and other information

The MAA Advisor page contains a table that lists the outage type, Oracle solutions for each outage, configuration status, and benefits. The MAA Advisor allows you to view High Availability solutions in the following ways:

- **Recommendations Only**—This condensed view shows only the recommended solutions (the default view) for the primary database.
- **All Solutions** —This expanded view shows all configuration recommendations and status for all primary and standby databases in this configuration. It includes an extra column **Target Name:Role** that provides the database name and shows the role (Primary, Physical Standby, or Logical Standby) of the database.

Figure 11–11 shows an example of the MAA Advisor page with the **Show** All Solutions view selected.

Figure 11–11 Maximum Availability Architecture (MAA) Advisor Page in Enterprise Manager



You can click the link in the Oracle Solution column to go to a page where you can set up, manage, and configure the specific solution area. Once a solution has been configured, click **Refresh** to update the configuration status. Once the page is refreshed, click MAA **Advisor Details** on the High Availability Console page to see the updated values.

Using Cluster Health Monitor

The Cluster Health Monitor (CHM) gathers operating system metrics in real time and stores them in its repository for later analysis to determine the root cause of many Oracle Clusterware and Oracle RAC issues with the assistance of Oracle Support. It also works with Oracle Database Quality of Service Management (Oracle Database QoS Management) by providing metrics to detect memory over-commitment on a node. With this information, Oracle Database QoS Management can take action to relieve the stress and preserve existing workloads.

> **See:** *Oracle Clusterware Administration and Deployment Guide* for an Overview of Managing Oracle Clusterware Environments and for more information about Cluster Health Monitor (CHM)

12

Recovering from Unscheduled Outages

This chapter describes the Oracle operational best practices that can tolerate or manage each unscheduled outage type and minimize downtime.

This chapter contains the following topics:

- Overview of Unscheduled Outages
- Recovering from Unscheduled Outages
- Restoring Fault Tolerance

> **See Also:** Chapter 13, "Reducing Downtime for Planned Maintenance" for information about scheduled outages.

Overview of Unscheduled Outages

This section describes unscheduled outages that affect the primary or secondary site components, and describes the recommended methods to repair or minimize the downtime associated with each outage.

Unscheduled outages are unanticipated failures in any part of the technology infrastructure that supports the application, including the following components:

- Hardware
- Software
- Network infrastructure
- Storage infrastructure
- Naming services infrastructure
- Database
- Data center

Your monitoring and high availability infrastructure should provide rapid detection and recovery from downtime.

Managing Unscheduled Outages on the Primary Site Best Practices

Solutions for unscheduled outages are critical for maximum availability of the system. Table 12–1 compares the four MAA reference architectures and summarizes the recovery steps for unscheduled outages on the primary site. The MAA reference architectures are described in *Oracle Database High Availability Overview*. For outages that require multiple recovery steps, the table includes links to the detailed descriptions in Section , "Recovering from Unscheduled Outages".

Table 12–1 Recovery Times and Steps for Unscheduled Outages on the Primary Site

Outage Scope	Bronze: Single Instance HA + Backup	Silver: Oracle RAC	Gold: Oracle RAC and Oracle Active Data Guard[1]	Platinum: Oracle 12c MAA + Platinum-ready Applications
site failure	Hours to days 1. Restore site. 2. Restore from backups. 3. Recover database.	Hours to days 1. Restore site. 2. Restore from backups. 3. Recover database.	Seconds to minutes[2] 1. Section , "Database Failover with a Standby Database" 2. Section , "Complete Site Failover (Failover to Secondary Site)" 3. Section , "Application Failover"	Zero application outage 1. Section , "Database Failover with a Standby Database" 2. Section , "Complete Site Failover (Failover to Secondary Site)" 3. Section , "Application Failover with Application Continuity and Transaction Guard"
clusterwide failure	Not applicable	Hours to days 1. Restore cluster or restore at least one node. 2. Optionally restore from backups if the data is lost or corrupted. 3. Recover database.	Seconds to minutes 1. Section , "Database Failover with a Standby Database" 2. Section , "Application Failover"	Zero application outage 1. Section , "Database Failover with a Standby Database" 2. Section , "Application Failover with Application Continuity and Transaction Guard"
recoverable server failure (node)	Minutes to an hour 1. Restart node and restart database with Oracle Restart. See *Oracle Database Administrator's Guide* 2. Reconnect users.	Seconds[3] 1. Manage automatically as described in Section , "Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)" 2. Section , "Application Failover"	Seconds[3] 1. Manage automatically as described in Section , "Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)" 2. Section , "Application Failover"	Zero application outage 1. Manage automatically as described in Section , "Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)" 2. Section , "Application Failover with Application Continuity and Transaction Guard"

Table 12–1 (Cont.) Recovery Times and Steps for Unscheduled Outages on the Primary Site

Outage Scope	Bronze: Single Instance HA + Backup	Silver: Oracle RAC	Gold: Oracle RAC and Oracle Active Data Guard[1]	Platinum: Oracle 12c MAA + Platinum-ready Applications
database instance failure	Minutes 1. Restart instance with Oracle Restart. 2. Reconnect users.	Seconds[3] 1. Manage automatically as described in Section , "Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)" 2. Section , "Application Failover"	Seconds[3] 1. Manage automatically as described in Section , "Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)" 2. Section , "Application Failover"	Zero application outage 1. Manage automatically with Oracle RAC 2. Section , "Application Failover with Application Continuity and Transaction Guard"
storage failure	No downtime[4] Section , "Oracle ASM Recovery After Disk and Storage Failures"	No downtime[4] Section , "Oracle ASM Recovery After Disk and Storage Failures"	No downtime[4] Section , "Oracle ASM Recovery After Disk and Storage Failures"	No downtime[4] Section , "Oracle ASM Recovery After Disk and Storage Failures"
data corruption	Minutes to hours Section , "Recovering from Data Corruption"	Minutes to hours Section , "Recovering from Data Corruption"	Possible no downtime with Oracle Active Data Guard: Section , "Use Active Data Guard" Seconds to minutes 1. Section , "Database Failover with a Standby Database" 2. Section , "Application Failover"	Possible no downtime with Active Data Guard: Section , "Use Active Data Guard" Zero Application outage to or seconds if data loss failover is required. 1. Section , "Database Failover with a Standby Database" 2. Section , "Application Failover with Application Continuity and Transaction Guard"
human error	< 30 minutes[5] Section , "Recovering from Human Error (Recovery with Flashback)"	< 30 minutes[5] Section , "Recovering from Human Error (Recovery with Flashback)"	<30 minutes[5] Section , "Recovering from Human Error (Recovery with Flashback)"	< 30 minutes[5] Section , "Recovering from Human Error (Recovery with Flashback)"
hang or slow down	customized and configurable[6] Section , "Application Failover"	customized and configurable[6] Section , "Application Failover"	customized and configurable[7] Section , "Application Failover"	customized and configurable[6] and[7] Section , "Application Failover with Application Continuity and Transaction Guard"

[1] While Data Guard physical replication is the most common data protection and availability solution used for Oracle Database, there are cases where active-active logical replication may be preferred, especially when control over the application makes it possible to implement. You may use Oracle GoldenGate in place of Data Guard for these requirements. See the topic, "Oracle Active Data Guard and Oracle GoldenGate" for additional discussion of the trade-offs between physical and logical replication at http://www.oracle.com/technetwork/database/features/availability/dataguardgoldengate-096557.html

[2] Recovery time indicated applies to database and existing connection failover. Network connection changes and other site-specific failover activities may lengthen overall recovery time.

[3] Database is still available, but portion of application connected to failed system is temporarily affected resulting in brownout in some cases of seconds to minutes depending if Oracle RAC or Oracle RAC One is utilized.

[4] Storage failures are prevented by using Oracle ASM with mirroring and its automatic rebalance capability.

[5] Recovery times from human errors depend primarily on detection time. If it takes seconds to detect a malicious DML or DLL transaction, then it typically only requires seconds to flash back the appropriate transactions, if properly rehearsed. Referential or integrity constraints must be considered.

[6] Oracle Enterprise Manager or a customized application heartbeat can be configured to detect application or response time slowdown and react to these SLA breaches. For example, you can configure the Enterprise Manager Beacon to monitor and detect application response times. Then, after a certain threshold expires, Enterprise Manager can alert and possibly restart the database.

[7] Oracle Enterprise Manager or a customized application heartbeat can be configured to detect application or response time slowdown and react to these SLA breaches. For example, you can configure the Enterprise Manager Beacon to monitor and detect application response times. Then, after a certain threshold expires, Enterprise Manager can call the Oracle Data Guard DBMS_DG.INITIATE_FS_FAILOVER PL/SQL procedure to initiate a failover.

See:

- *Oracle Data Guard Broker* for more information about "Application Initiated Fast-Start Failover"
- The topic, "Oracle Active Data Guard and Oracle GoldenGate" for additional discussion of the trade-offs between physical and logical replication at

  http://www.oracle.com/technetwork/database/features/availability/dataguardgoldengate-096557.html

Managing Unscheduled Outages on the Standby Site Best Practices

Outages on the standby site do not impact the availability of the primary database when using Data Guard Maximum Availability (synchronous communication with net_timeout) or Maximum Performance (asynchronous communication).

Note: Outages to a system that uses the Active Data Guard option with the standby database can affect applications that are using the standby database for read activity, but such outages do not impact the availability of the primary database (the availability is based on the mode you specify).

Data Guard Maximum Protection, however, has an impact on availability if the primary database does not receive acknowledgment from a standby database running in SYNC transport mode (net_timeout does not apply to Maximum Protection). For this reason, if you are using Maximum Protection you should follow the MAA best practice of deploying two SYNC standby databases and multiple far sync instances if required. With two standby databases a single standby outage does not impact primary availability or zero data loss protection.

If limited system resources make it impractical to deploy two standby databases, then the availability of the primary database can be restored simply by downgrading the data protection mode to Maximum Availability and restarting the primary database.

Table 12–2 summarizes the recovery steps for unscheduled outages of the standby database on the secondary site. For outages that require multiple recovery steps, the

table includes links to the detailed descriptions in Section , "Recovering from Unscheduled Outages".

Table 12–2 Recovery Steps for Unscheduled Outages on the Secondary Site or Far Sync Instance Site

Outage Type	Recovery Steps for Single-Instance or Oracle RAC Standby Database
Computer failure (instance or node)	1. Restart node and standby instance when they are available. 2. Restart recovery. The broker automatically restarts the log apply services. **Note 1:** If there is only one standby database and if Maximum Protection is configured, then the primary database shuts down to ensure that there is no data divergence with the standby database (no unprotected data). **Note 2:** If this is an Oracle RAC standby database, then there is no affect on primary database availability if you configured the primary database Oracle Net descriptor to use connect-time failover to an available standby instance. If you are using the broker, connect-time failover is configured automatically.
Data corruption	Section , "Restoring Fault Tolerance After a Standby Database Data Failure"
Primary database opens with `RESETLOGS` because of Flashback Database operations or point-in-time media recovery	Section , "Restoring Fault Tolerance After the Primary Database Was Opened Resetlogs"
Far Sync instance or node failure	1. If configured as Oracle RAC Far Sync, then fail over to available instance or node. 2. If Oracle RAC instance is not available and `ALTERNATE` destination is available, then fail over to alternate destination. As the last resort, you can configure `ALTERNATE` as the terminal standby database. If Far Sync configuration best practices have been followed, each of these steps will be performed automatically by the database and no manual intervention is necessary.

See Also:

- Section , "Oracle Data Guard Configuration Best Practices"
- *Oracle Data Guard Concepts and Administration* for information about "Data Guard Protection Modes"
- *Oracle Data Guard Concepts and Administration* for more information about Oracle Active Data Guard option and when Redo Apply can be active while the physical standby database is open
- *Oracle Data Guard Broker* for information about "How the Protection Modes Influence Broker Operations"
- MAA white paper "Oracle Active Data Guard Far Sync Zero Data Loss at Any Distance" at `http://www.oracle.com/technetwork/database/availability/farsync-2267608.pdf`

Recovering from Unscheduled Outages

This section describes best practices for recovering from various types of unscheduled outages.

Complete Site Failover (Failover to Secondary Site)

With complete site failover, the database, the middle-tier application server, and all user connections fail over to a secondary site that is prepared to handle the production load.

When to Use Complete Site Failover

If the standby site meets the prerequisites, then complete site failover is recommended for the following scenarios:

- Primary site disaster, such as natural disasters or malicious attacks
- Primary network-connectivity failures
- Primary site power failures

Best Practices for Complete Site Failover

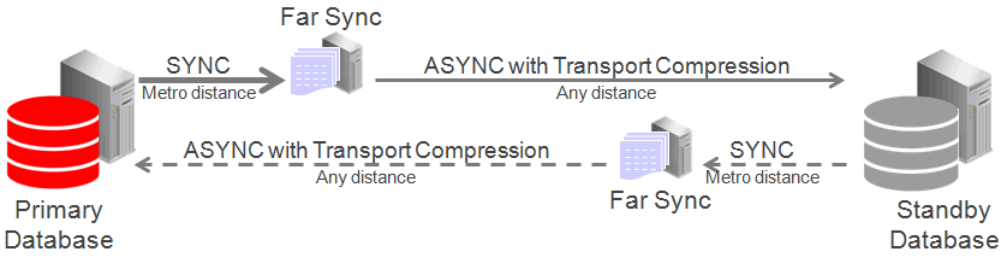
To expedite site failover in minutes:

- Use the Data Guard configuration best practices in Section , "General Data Guard Configuration Best Practices"
- Use Data Guard fast-start failover to automatically fail over to the standby database, with a recovery time objective (RTO) of less than 30 seconds (described in Section , "Fast-Start Failover Best Practices")
- Maintain a running middle-tier application server on the secondary site to avoid the startup time, or redirect existing applications to the new primary database using the Fast Connection Failover best practices described in:
  - Chapter 10, "Client Failover Best Practices for Highly Available Oracle Databases"
  - The MAA white paper: "Client Failover Best Practices for Highly Available Oracle Databases - Oracle Database 12c" from the MAA Best Practices area for Oracle Database at

    `http://www.oracle.com/goto/maa`
- Configure Automatic Domain Name Server (DNS) failover procedure. Automatic DNS failover occurs after a primary site is inaccessible and the wide-area traffic manager at the secondary site returns the virtual IP address of a load balancer at the secondary site and clients are directed automatically on the subsequent reconnect.

The potential for data loss is dependent on the Data Guard protection mode used: Maximum Protection, Maximum Availability, or Maximum Performance. For the Gold and Platinum reference architectures, zero data loss can be achieved over WAN by using Oracle 12*c* Far Sync instance. Figure 12–1 is an example configuration with Far Sync.

Figure 12–1 Example Configuration With Far Sync



Application continuity can enable zero application outage; however, that is only possible if the application tiers are shared across sites.

Complete Site Failover

A wide-area traffic manager on the primary and secondary sites provides the site failover function. The wide-area traffic manager can redirect traffic automatically if the primary site, or a specific application on the primary site, is not accessible. It can also be triggered manually to switch to the secondary site for switchovers. Traffic is directed to the secondary site only when the primary site cannot provide service due to an outage or after a switchover. If the primary site fails, then user traffic is directed to the secondary site automatically.

Figure 12–2 illustrates the possible network routes before site failover:

1. Client requests enter the client tier of the primary site and travel by the WAN traffic manager.
2. Client requests are sent through the firewall into the demilitarized zone (DMZ) to the application server tier.
3. Requests are forwarded through the active load balancer to the application servers.
4. Requests are sent through another firewall and into the database server tier.
5. The application requests, if required, are routed to an Oracle RAC instance.
6. Responses are sent back to the application and clients by a similar path.

Figure 12–2 Network Routes Before Site Failover

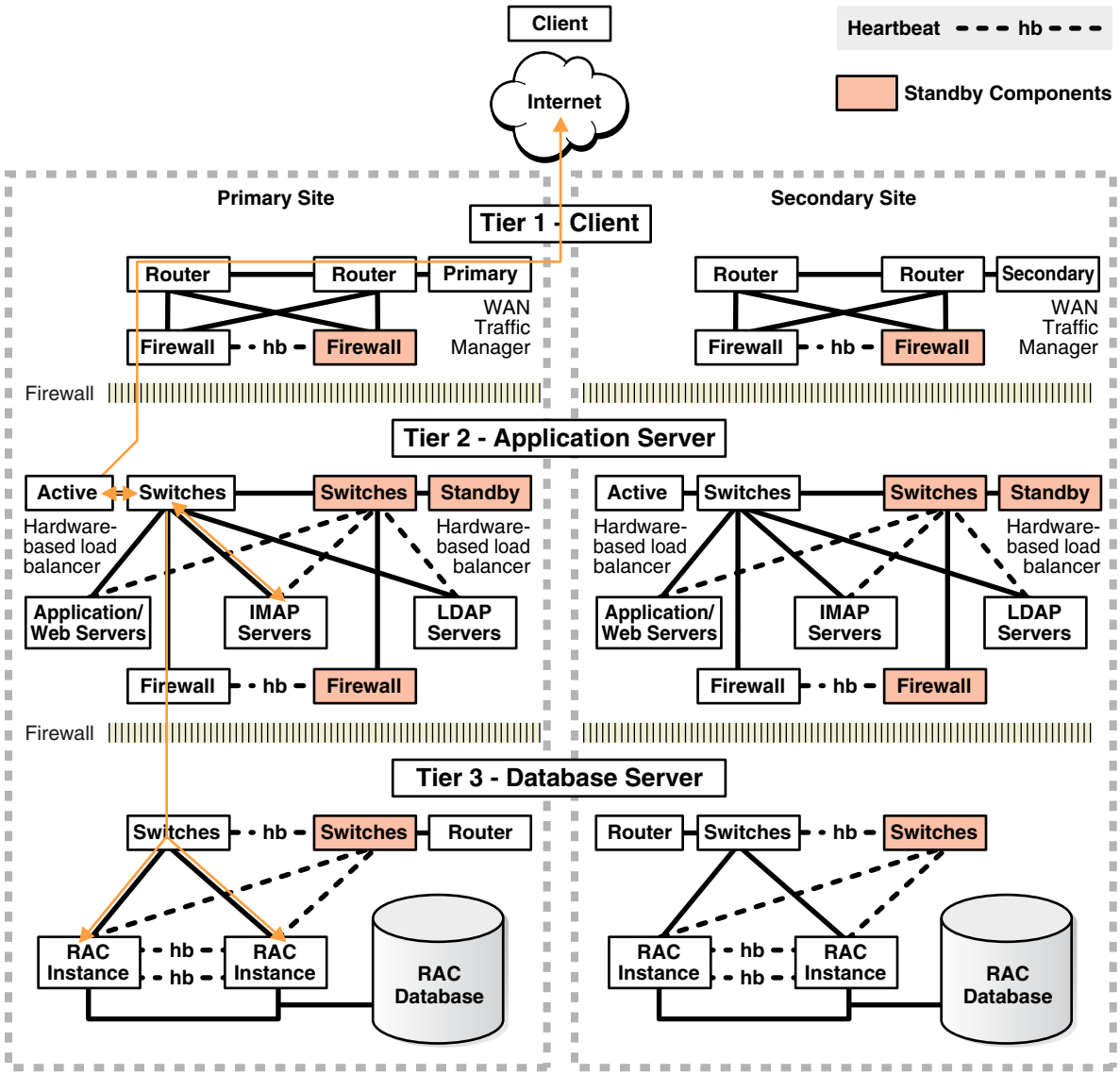
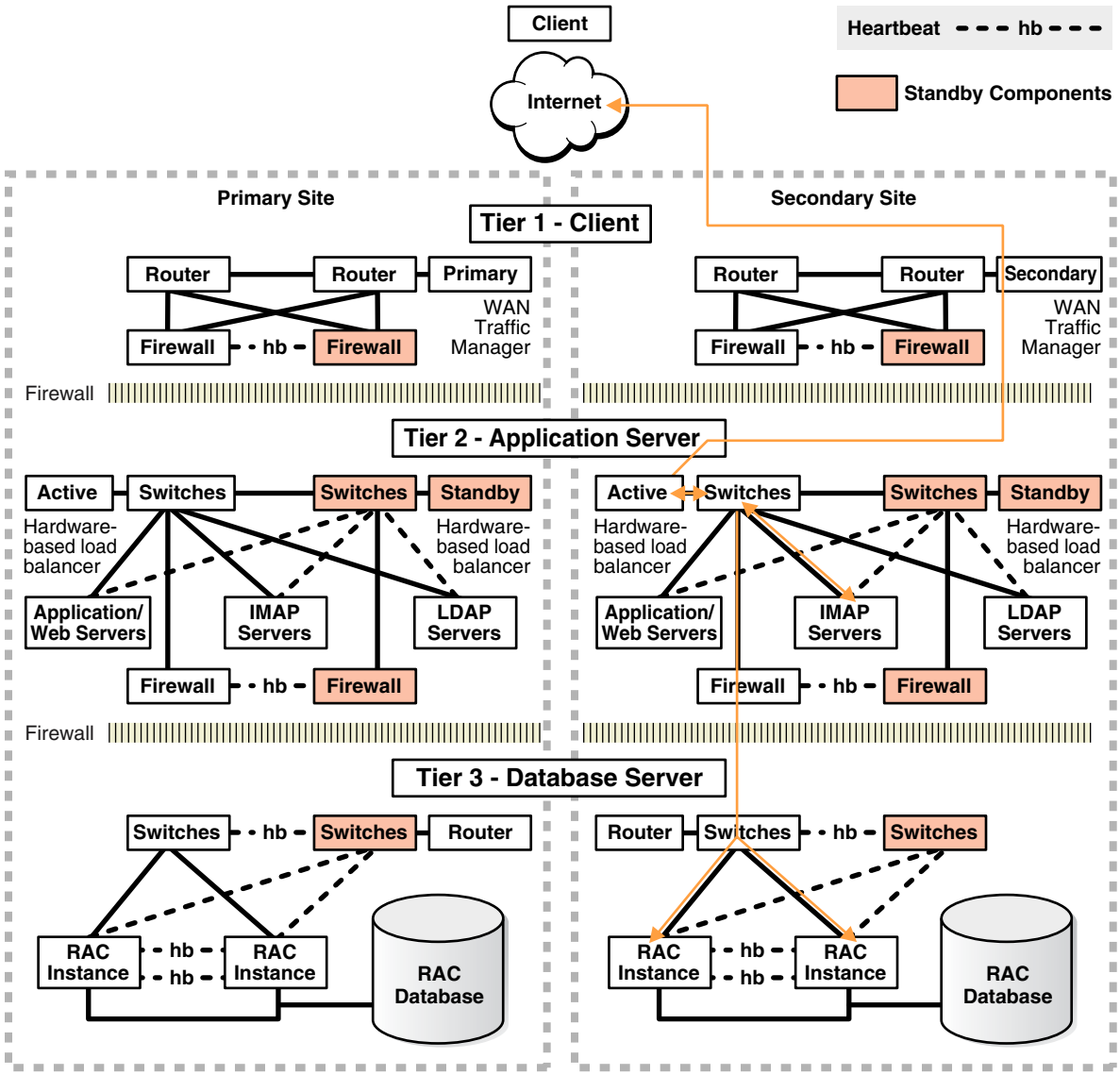


Figure 12–3 illustrates the network routes after site failover. Client or application requests enter the secondary site at the client tier and follow the same path on the secondary site that they followed on the primary site.

Figure 12–3 Network Routes After Site Failover



The following steps describe the effect of a failover or switchover on network traffic:

1. The administrator has failed over or switched over the primary database to the secondary site. This is automatic if you are using Data Guard fast-start failover.
2. The administrator starts the middle-tier application servers on the secondary site, if they are not running. In some cases the same middle-tier application servers can be leveraged if they do not reside in the failed site.
3. The wide-area traffic manager selection of the secondary site can be automatic for an entire site failure. The wide-area traffic manager at the secondary site returns the virtual IP address of a load balancer at the secondary site and clients are directed automatically on the subsequent reconnect. In this scenario, the site failover is accomplished by an automatic domain name system (DNS) failover.

   Alternatively, a DNS administrator can manually change the wide-area traffic manager selection to the secondary site for the entire site or for specific applications. The following is an example of a manual DNS failover:

   a. Change the DNS to point to the secondary site load balancer:

The master (primary) DNS server is updated with the zone information, and the change is announced with the DNS `NOTIFY` announcement.

The slave DNS servers are notified of the zone update with a DNS `NOTIFY` announcement, and the slave DNS servers pull the zone information.

Note: The master and slave servers are authoritative name servers. Therefore, they contain trusted DNS information.

---

b. Clear affected records from caching DNS servers.

A caching DNS server is used primarily for performance and fast response. The caching server obtains information from an authoritative DNS server in response to a host query and then saves (caches) the data locally. On a second or subsequent request for the same data, the caching DNS server responds with its locally stored data (the cache) until the time-to-live (TTL) value of the response expires. At this time, the server refreshes the data from the zone master. If the DNS record is changed on the primary DNS server, then the caching DNS server does not pick up the change for cached records until TTL expires. Flushing the cache forces the caching DNS server to go to an authoritative DNS server again for the updated DNS information.

Flush the cache if the DNS server being used supports such a capability. The following is the flush capability of common DNS BIND versions:

BIND 9.3.0: The command `rndc flushname` *`name`* flushes individual entries from the cache.

BIND 9.2.0 and 9.2.1: The entire cache can be flushed with the command `rndc flush`.

BIND 8 and BIND 9 up to 9.1.3: Restarting the named server clears the cache.

c. Refresh local DNS service caching.

Some operating systems might cache DNS information locally in the local name service cache. If so, this cache must also be cleared so that DNS updates are recognized quickly.

Solaris: `nscd`

Linux: `/etc/init.d/nscd restart`

Microsoft Windows: `ipconfig /flushdns`

d. The secondary site load balancer directs traffic to the secondary site middle-tier application server.

e. The secondary site is ready to take client requests.

Failover also depends on the client's web browser. Most browser applications cache the DNS entry for a period. Consequently, sessions in progress during an outage might not fail over until the cache timeout expires. To resume service to such clients, close the browser and restart it.

Database Failover with a Standby Database

Failover is the operation of transitioning one standby database to the role of primary database. A failover operation is invoked when an unplanned failure occurs on the primary database and there is no possibility of recovering the primary database in a timely fashion.

With Oracle Data Guard, you can automate the failover process using the broker and fast-start failover, or you can perform the failover manually:

- **Fast-start failover** eliminates the uncertainty of a process that requires manual intervention and automatically executes a zero loss or minimum-loss failover (that you configure using the `FastStartFailoverLagLimit` property) within seconds of an outage being detected. See Section , "Fast-Start Failover Best Practices" for configuration best practices.
- **Manual failover** allows for a failover process where decisions are user driven using any of the following methods:
  - Oracle Enterprise Manager
  - The broker command-line interface (DGMGRL)
  - SQL*Plus statements

    See Section , "Best Practices for Performing Manual Failover".

A database failover is accompanied by an application failover and, in some cases, preceded by a site failover. For Platinum-ready applications and for zero data loss failover, applications can achieve zero application outage with Application Failover with Application Continuity and Transaction Guard. After the Data Guard failover, the secondary site hosts the primary database. You must reinstate the original primary database as a new standby database to restore fault tolerance of the configuration. See Section , "Restoring a Standby Database After a Failover."

A failover operation typically occurs in seconds to minutes, and with little or no data loss. With Oracle 12*c* Far Sync capability, zero data loss protection to a standby database is possible regardless of the physical distance between primary and standby databases.

See Also:

- *Oracle Data Guard Concepts and Administration*.for a complete description of failover processing
- The "Data Guard Fast-Start Failover" and "Data Guard Switchover and Failover" MAA best practice white papers available from the MAA Best Practices area for Oracle Database at

  `http://www.oracle.com/goto/maa`

When To Perform a Data Guard Failover

When a primary database failure cannot be repaired in time to meet your Recovery Time Objective (RTO) using local backups or Flashback technology, you should perform a failover using Oracle Data Guard.

You should perform a failover manually due to an unplanned outage such as:

- A site disaster, which results in the primary database becoming unavailable
- Damage resulting from user errors that cannot be repaired in a timely fashion
- Data corruptions that are not automatically resolved by Oracle Active Data Guard, data or media damage, database or cluster failures

A failover requires that you reinstate the initial primary database as a standby database to restore fault tolerance to your environment. You can quickly reinstate the standby database using Flashback Database provided the original primary database has not been damaged. See Section , "Restoring a Standby Database After a Failover."

Best Practices for Implementing Fast-Start Failover

A fast-start failover is completely automated and requires no user intervention.

There are no procedural best practices to consider when performing a fast-start failover. However, it is important to address all of the configuration best practices described in Section , "Fast-Start Failover Best Practices".

See also:

- Section , "Fast-Start Failover Best Practices"
- MAA white paper "Oracle Active Data Guard Far Sync Zero Data Loss at Any Distance" at http://www.oracle.com/technetwork/database/availability/farsync-2267608.pdf

Best Practices for Performing Manual Failover

When performing a manual failover:

- Follow the configuration best practices outlined in Section , "Manual Failover Best Practices."
- Choose from the following methods:
  - **Oracle Enterprise Manager**

    See *Oracle Data Guard Broker* for complete information about how to perform a manual failover using Oracle Enterprise Manager. The procedure is the same for both physical and logical standby databases.
  - **Oracle Data Guard broker command-line interface (DGMGRL)**

    See *Oracle Data Guard Broker* for complete information about how to perform a manual failover using Oracle Enterprise Manager. The procedure is the same for both physical and logical standby databases.
  - **SQL*Plus statements:**
    * *Oracle Data Guard Concepts and Administration* for information about Physical standby database steps for "Performing a Failover to a Physical Standby Database"
    * *Oracle Data Guard Concepts and Administration* for information about Logical standby database steps for "Performing a Failover to a Logical Standby Database"

Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)

Oracle RAC Recovery is performed automatically when there is a node or instance failure. In regular multi instance Oracle RAC environments, surviving instances automatically recover the failed instances and potentially aid in the automatic client failover. Recover times can be bounded by adopting the database and Oracle RAC configuration best practices and can usually lead to instance recovery times of seconds to minutes in very large busy systems, with no data loss. For Oracle RAC One Node configurations recover times are expected to take longer than full Oracle RAC; with Oracle RAC One Node a replacement instance must be started first before it can do the instance recovery.

For instance or node failures with Oracle RAC and Oracle RAC One Node, use the following recovery methods:

- Automatic Instance Recovery for Failed Instances

- Automatic Service Relocation
- Oracle Cluster Registry Recovery

Automatic Instance Recovery for Failed Instances

Instance failure occurs when software or hardware problems cause an instance to shutdown or abort. After instance failure, Oracle automatically uses the online redo log file to perform database recovery.

Instance recovery in Oracle RAC does not include restarting the failed instance or the recovery of applications that were running on the failed instance. Applications will run continuously using service relocation and fast application notification (as described in Section , "Automatic Service Relocation").

When one instance performs recovery for another instance, the recovering instance (and active foreground processes in the case of transaction recovery):

- Reads redo log entries generated by the failed instance and uses that information to ensure that committed transactions are recorded in the database. Thus, data from committed transactions is not lost
- Rolls back uncommitted transactions that were active at the time of the failure and releases resources used by those transactions

When multiple instances fail, if one instance survives Oracle RAC performs instance recovery for any other instances that fail. If all instances of an Oracle RAC database fail, then on subsequent restart of any instance a crash recovery occurs and all committed transactions are recovered. Data Guard is the recommended solution to survive outages when all instances of a cluster fail.

Automatic Service Relocation

Service reliability is achieved by configuring and failing over among the surviving instances. A service will be made available by multiple database instances to provide a service that is needed. If a hardware failure occurs and the failure adversely affects an Oracle RAC database instance, then depending on the configuration, Oracle Clusterware does one the following:

- Oracle Clusterware automatically moves any services on the failed database instance to another available instance, as configured with DBCA or Enterprise Manager. Oracle Clusterware recognizes when a failure affects a service and automatically fails over the service across the surviving instances supporting the service.

  Note: With Oracle RAC One Node the relocation occurs when another instance on a different node is started and enabled for the appropriate services. Thus, Oracle RAC One Node starts a new instance when an instance fails but the new instance is not a "surviving instance."

- A service can be made available on multiple instances by default. In this case, when one of those multiple instances is lost the clients continue to use the available services across the surviving instances, but there are less resources to do the work.

In parallel, Oracle Clusterware attempts to restart and integrate the failed instances and dependent resources back into the system and Cluster Ready Services (CRS) will try to restart the database instance three times. Clients can "subscribe" to node failure

events, in this way clients can be notified of instance problems quickly and new connections can be setup (Oracle Clusterware does not setup the new connections, the clients setup the new connections). Notification of failures using fast application notification (FAN) events occur at various levels within the Oracle Server architecture. When following Oracle client failover best practices (point to chapter 10) in conjunction with Application Continuity applications will in most cases be able to seamless handle outages without seeing any errors.

Oracle Cluster Registry Recovery

Loss of the Oracle Cluster Registry (OCR) file affects the availability of Oracle RAC and Oracle Clusterware. The OCR file can be restored from a backup that is automatically created or from an export file that is manually created by using the `ocrconfig` tool (also use `ocrconfig` to restore the backup). Additionally, Oracle can optionally mirror the OCR so that a single OCR device failure can be tolerated. Ensure the OCR mirror is on a physically separate device and preferably on a separate controller. For more information, see Section , "Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks with Oracle ASM".

If all of the voting disks are corrupted, then you must restore them. To do this you use the `crsctl` command. The steps you use depend on where you store your voting files. If the voting disks are stored in Oracle ASM, then run the commands to migrate the voting disks to the Oracle ASM disk group you specify, with: `crsctl replace votedisk`. If you did not store voting disks in Oracle ASM, then you run the commands to delete and add the voting disks: `crsctl delete css votedisk` and `crsctl add css votedisk`.

See Also:

- Section , "Regularly Back Up OCR to Tape or Offsite"
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about Administering Storage in Real Application Clusters
- *Oracle Clusterware Administration and Deployment Guide* for information about Restoring Oracle Cluster Registry
- *Oracle Clusterware Administration and Deployment Guide* for information about Restoring Voting Disks

Application Failover

With a minimal configuration, applications can receive fast and efficient notification when instances providing services become unavailable. When notified, application reconnects occur transparently to the surviving instances of an Oracle RAC database or to a standby database that has assumed the primary role following a failover.

In an Oracle RAC configuration, services are essential to achieving fast and transparent application failover. Clients are notified of a service relocation through Fast Application Notification (FAN).

In an Oracle Data Guard configuration, you can configure services for client failover across sites. After a site failure in a Data Guard configuration, the new primary database can automatically publish the production service while notifying affected clients, through FAN events, that the services are no longer available on the failed primary database.

For hangs or situations in which the response time is unacceptable, you can configure Oracle Enterprise Manager or a custom application heartbeat to detect application or

response time slowdown and react to these situations. For example, you can configure the Enterprise Manager Beacon to monitor and detect application response times. Then, after a certain time threshold expires, Enterprise Manager can call the Oracle Data Guard `DBMS_DG.INITIATE_FS_FAILOVER` PL/SQL procedure to initiate a database failover immediately followed by an application failover using FAN notifications and service relocation.

FAN notifications and service relocation enable automatic and fast redirection of clients if any failure or planned maintenance results in an Oracle RAC or Oracle Data Guard fail over.

See Also:

- Chapter 10, "Client Failover Best Practices for Highly Available Oracle Databases"
- The MAA white paper "Client Failover for Highly Available Oracle Databases" from the MAA Best Practices area for Oracle Database at

  `http://www.oracle.com/goto/maa`
- *Oracle Data Guard Broker* for more information about Application Initiated Fast-Start Failover and the `DBMS_DG.INITIATE_FS_FAILOVER` PL/SQL procedure
- *Oracle Real Application Clusters Administration and Deployment Guide* "Workload Management with Dynamic Database Services"

Application Failover with Application Continuity and Transaction Guard

Platinum-ready applications can achieve zero application outage during unplanned outages and planned maintenance activities whenever the following prerequisites are met.

- Application has been configured with application continuity and transaction guard AND
- Zero data loss Oracle RAC failover has occurred OR
- Zero data loss Data Guard failover has occurred

Following planned and unplanned outages, Application Continuity attempts to mask the outage by rebuilding the database session, and resubmitting the pending work following recoverable errors that make the database session unavailable. Application Continuity does not resubmit work following call failures due to non-recoverable errors. Submission of invalid data values is an example of a non-recoverable error that would not be available for replay.

When Application Continuity is configured, an end-user request is executed at-most once; replay is started if the time has not exceeded the replay timeout attribute specified for the service. When replaying, Application Continuity appears to the user as a slightly delayed execution. When replay succeeds, this feature masks applications from transient outages (such as session failure, instance or node outages, network failures, and so on) and from planned outages such as repairs, configuration changes, and patch application.

See Also: Section , "Configuring Application Continuity"

Oracle ASM Recovery After Disk and Storage Failures

Table 12–3 summarizes the impacts and recommended repairs for various Oracle ASM failure types.

Table 12–3 Types of Oracle ASM Failures and Recommended Repair

Failure	Description	Impact	Recommended Repair
Oracle ASM instance failure	Oracle ASM instance fails	All database instances accessing Oracle ASM storage from the same node shut down In Flex ASM configuration, all database instances remain available if another ASM instance is available	Automatic Section , "Oracle RAC Recovery for Unscheduled Outages (for Node or Instance Failures)" If Oracle RAC is not used, use Data Guard failover (see Section , "Best Practices for Implementing Fast-Start Failover") If Oracle RAC and Data Guard are not used, fix the underlying problem and then restart Oracle ASM and the database instances
Oracle ASM disk failure	One or more Oracle ASM disks fail, but all disk groups remain online	All data remains accessible. This is possible only with normal or high redundancy disk groups	Oracle ASM automatically rebalances to the remaining disk drives and reestablishes redundancy. There must be enough free disk space in the remaining disk drives to restore the redundancy or the rebalance may fail with an `ORA-15041`. For more information, see Section , "Oracle Storage Grid Best Practices for Planned Maintenance" **Note:** External redundancy disk groups should use mirroring in the storage array to protect from disk failure. Disk failures should not be exposed to Oracle ASM.
Data area disk-group failure	One or more Oracle ASM disks fail, and data area disk group goes offline	Databases accessing the data area disk group shut down	Perform Data Guard failover or local recovery as described in Section , "Data Area Disk Group Failure"
Fast recovery area disk-group failure	One or more Oracle ASM disks fail, and the fast recovery area disk group goes offline	Databases accessing the fast recovery area disk group shut down	Perform local recovery or Data Guard failover as described in Section , "Fast Recovery Area Disk Group Failure"

Oracle ASM Instance Failure

If the Oracle ASM instance fails, then database instances accessing Oracle ASM storage from the same node shut down. The following list describes failover processing:

- If the primary database is an Oracle RAC database, then application failover occurs automatically and clients connected to the database instance reconnect to remaining instances. Thus, the service is provided by other instances in the cluster and processing continues. The recovery time typically occurs in seconds.
- If the primary database is not an Oracle RAC database, then an Oracle ASM instance failure shuts down the entire database.
- If the configuration uses Oracle Data Guard and fast-start failover is enabled, a database failover is triggered automatically and clients automatically reconnect to the new primary database after the failover completes. The recovery time is the amount of time it takes to complete an automatic Data Guard fast-start failover operation. If fast-start failover is not configured, then you must recover from this outage by either restarting the Oracle ASM and database instances manually, or by performing a manual Data Guard failover.

- If the configuration includes neither Oracle RAC nor Data Guard, then you must manually restart the Oracle ASM instance and database instances. The recovery time depends on how long it takes to perform these tasks.

Oracle ASM Disk Failure

If the Oracle ASM disk fails, then failover processing is as follows:

- External redundancy

  If an Oracle ASM disk group is configured as an external redundancy type, then a failure of a single disk is handled by the storage array and should not be seen by the Oracle ASM instance. All Oracle ASM and database operations using the disk group continue normally.

  However, if the failure of an external redundancy disk group is seen by the Oracle ASM instance, then the Oracle ASM instance takes the disk group offline immediately, causing Oracle instances accessing the disk group to crash. If the disk failure is temporary, then you can restart Oracle ASM and the database instances and crash recovery occurs after the disk group is brought back online.

- Normal or a high-redundancy

  If an Oracle ASM disk group is configured as a normal or a high-redundancy type, then disk failure is handled transparently by Oracle ASM and the databases accessing the disk group are not affected.

  An Oracle ASM instance automatically starts an Oracle ASM rebalance operation to distribute the data of one or more failed disks to the remaining, intact disks of the Oracle ASM disk group. While the rebalance operation is in progress, subsequent disk failures may affect disk group availability if the disk contains data that has yet to be remirrored. When the rebalance operation completes successfully, the Oracle ASM disk group is no longer at risk in the event of a subsequent failure. Multiple disk failures are handled similarly, provided the failures affect only one failure group in an Oracle ASM disk group with normal redundancy.

The failure of multiple disks in multiple failure groups where a primary extent and all of its mirrors have been lost causes the disk group to go offline.

When Oracle ASM disks fail, use the following recovery methods:

- Using Enterprise Manager to Repair Oracle ASM Disk Failure
- Using SQL to Add Replacement Disks Back to the Disk Group

Using Enterprise Manager to Repair Oracle ASM Disk Failure

Figure 12–4 shows Enterprise Manager reporting disk failures. Five of 14 alerts are shown. The five alerts shown are Offline messages for Disk RECO2.

Figure 12–4 Enterprise Manager Reports Disk Failures

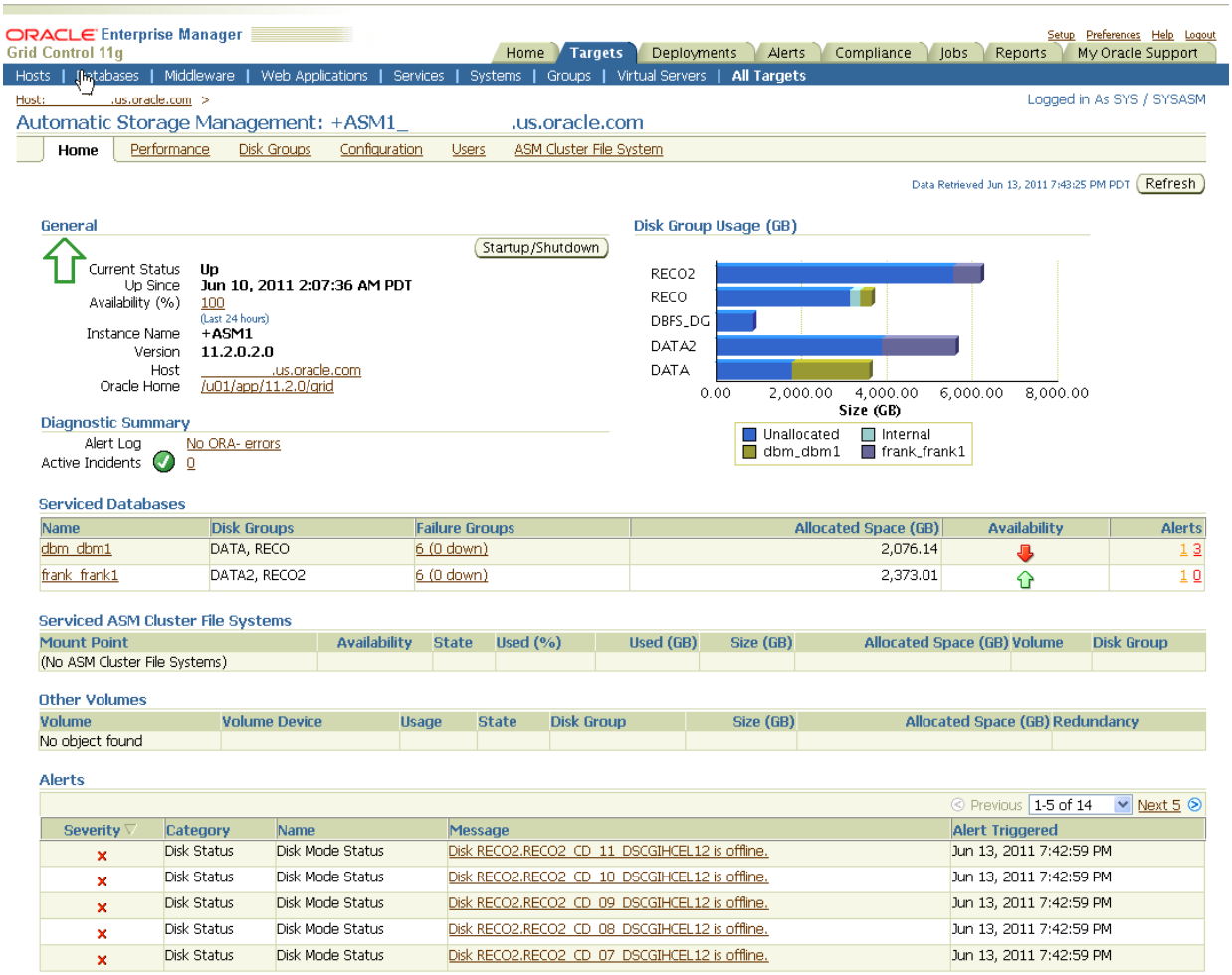


Figure 12–5 shows Enterprise Manager reporting the status of data area disk group DATA, database Data Guard disk group DBFS_DG, and recovery area disk group RECO.

Figure 12–5 Enterprise Manager Reports Oracle ASM Disk Groups Status

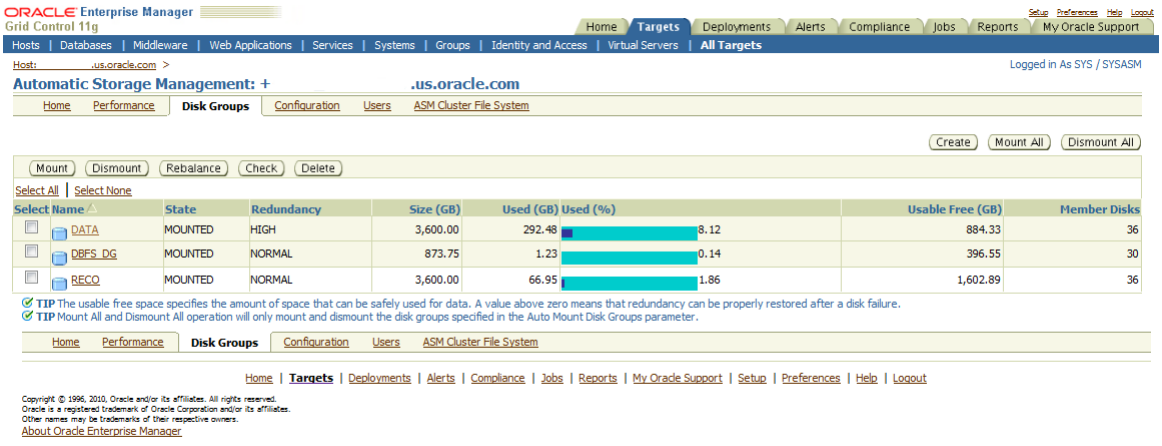
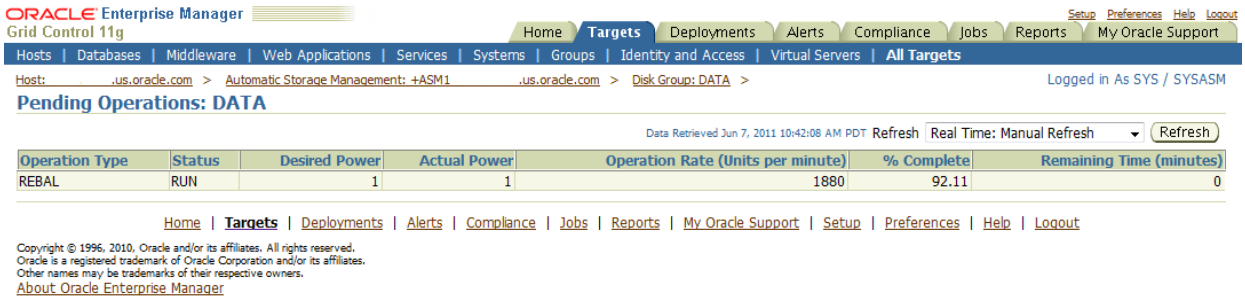


Figure 12–6 shows Enterprise Manager reporting a pending REBAL operation on the DATA disk group. The operation is almost done, as shown in **% Complete**, and the **Remaining Time** is estimated to be 0 minutes.

Figure 12–6 Enterprise Manager Reports Pending REBAL Operation



Using SQL to Add Replacement Disks Back to the Disk Group Perform these steps after one or more failed disks of one specific failure group have been dropped and must be replaced with new disks:

1. Add the one or more replacement disks to the failed disk group with the following SQL command:

```
ALTER DISKGROUP disk_group
   ADD FAILGROUP failure_group
   DISK 'disk1','disk2',...;
```

2. Check the progress of the operation:

```
SELECT * FROM V$ASM_OPERATION;
```

Data Area Disk Group Failure

A data area disk group failure should occur only when there have been multiple failures. For example, if the data area disk group is defined as external redundancy, a single-disk failure should not be exposed to Oracle ASM. However, multiple disk failures in a storage array may be seen by Oracle ASM causing the disk group to go offline. Similarly, multiple disk failures in different failure groups in a normal or high-redundancy disk group may cause the disk group to go offline.

When one or more disks fail in a normal or high redundancy disk group, and the Oracle ASM disk group is accessible, there is no loss of data and no immediate loss of accessibility. An Oracle ASM instance automatically starts an Oracle ASM rebalance operation to distribute the data on the one or more failed disks to the disks that remain intact in the Oracle ASM disk group. When the rebalance operation completes successfully, the Oracle ASM disk group is no longer at risk if a second failure occurs. There must be enough free disk space on the remaining disks in the disk group for the rebalance to complete successfully.

Table 12–4 summarizes the possible solutions for recovering from a data area disk group failure.

Table 12–4 Recovery Options for Data Area Disk Group Failure

Recovery Option	Recovery Time Objective (RTO)	Recovery Point Objective (RPO)
Data Guard failover (see Section , "Fast Recovery Area Disk Group Failure")	Seconds to minutes For Platinum: Zero Application Outage	Varies depending on the data protection level chosen Active Data Guard Far Sync feature can enable more zero data loss configurations
Local Recovery (see "Local Recovery Steps")	Database restore and recovery time	Zero

If Data Guard is being used and fast-start failover is configured, then an automatic failover occurs when the database shuts down due to the data area disk group going offline. If fast-start failover is not configured, then perform a manual failover.

If you decide to perform a Data Guard failover then the **recovery time objective (RTO)** is expressed in terms of minutes or seconds, depending on the presence of the Data Guard observer process and fast-start failover. However, if a manual failover occurs and not all data is available on the standby site, then data loss might result.

After Data Guard failover has completed and the application is available, you must resolve the data area disk group failure. Continue with the following "Local Recovery Steps" procedure to resolve the Oracle ASM disk group failure.

The RTO for local recovery only is based on the time required to:

1. Repair and replace the failed storage components
2. Restore and recover the database

Because the loss affects only the data-area disk group, there is no loss of data. All transactions are recorded in the Oracle redo logs that reside in the fast recovery area, so complete media recovery is possible.

If you are not using Data Guard, then perform the following local recovery steps. The time required to perform local recovery depends on how long it takes to restore and recover the database. There is no data loss when performing local recovery.

Local Recovery Steps

Perform these steps after one or more failed disks have been replaced and access to the storage has been restored:

> **Note:** If you have performed an Oracle Data Guard failover to a new primary database, then you can now use the following procedure to restore and sync the Data Guard setup. Also, see Section , "Restoring a Standby Database After a Failover".

1. Rebuild the Oracle ASM disk group using the new storage location by issuing the following SQL*Plus statement on the Oracle ASM instance:

```
SQL> CREATE DISKGROUP DATA NORMAL REDUNDANCY DISK 'path1','path2',...force;
```

2. Create a dummy pfile containing the DB_NAME, DB_UNIQUE_NAME and COMPATIBLE parameters to be used to start up the database instance, because the spfile used to startup the database resides in the +DATA diskgroup, which is unavailable.

3. Start the database instance NOMOUNT by issuing the following RMAN command:

```
RMAN> STARTUP FORCE NOMOUNT;
```

4. Restore the spfile from the spfile autobackup, and restore the control file from the surviving copy located in the recovery area:

```
RMAN> RESTORE CONTROLFILE FROM 'recovery_area_controlfile';
```

   The absolute path of the controlfile restore location needs to be specified in the restore controlfile command as follows:

```
RMAN> RESTORE CONTROLFILE TO '+RECO/dbm/control01.ora' from autobackup db_
recovery_file_dest='+RECO' db_name='db_name';
```

5. After the spfile is restored to a location in the previous step, the dummy init.ora file created in the first step must be modified to point the restored spfile.

   For example, the pfile would look something like:

```
$ cat initdbm1.ora
SPFILE='+RECO/dbm/spfiledbm.ora' ==> spfile restored in the previous step.
```

6. Restart the database instance with the restored spfile.
7. Modify the control_files parameter in the spfile to point to the restored control files:

```
ALTER SYSTEM SET control_files='<full path and filename of the location of the
restored control files>';
```

8. Start the database instance MOUNT:

```
RMAN> STARTUP FORCE MOUNT;
```

9. Restore the database:

```
RMAN> RESTORE DATABASE
```

10. Recover the database:

```
RMAN> RECOVER DATABASE;
```

11. If you use block change tracking, then disable and re-enable the block change tracking file using SQL*Plus statements:

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

12. Open the database:

```
SQL> ALTER DATABASE OPEN;
```

13. Re-create the log file members on the failed Oracle ASM disk group:

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER 'filename';
SQL> ALTER DATABASE ADD LOGFILE MEMBER 'disk_group' TO GROUP group_no;
```

14. Recreate the temp tablespace as part of cleaning up of the restored database.

```
SQL> DROP TABLESPACE TEMP;
SQL> CREATE TEMPORARY TABLESPACE TEMP TEMPFILE 'disk_group';
```

15. Perform an incremental level 0 backup using the following RMAN command:

```
RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

Fast Recovery Area Disk Group Failure

When the fast recovery-area disk group fails, the database crashes because the control file member usually resides in the fast recovery area and Oracle requires that all control file members are accessible. The fast recovery area can also contain the flashback logs, redo log members, and all backup files.

Because the failure affects only the fast recovery-area disk group, there is no loss of data. No database media recovery is required, because the data files and the online redo log files are still present and available in the data area.

A fast recovery area disk group failure typically occurs only when there have been multiple failures. For example, if the fast recovery-area disk group is defined as external redundancy, a single-disk failure should not be exposed to Oracle ASM. However, multiple disk failures in a storage array may affect Oracle ASM and cause the disk group to go offline. Similarly, multiple disk failures in different failure groups in a normal or high-redundancy disk group may cause the disk group to go offline.

Table 12–5 summarizes possible solutions when the fast recovery-area disk group fails.

Table 12–5 Recovery Options for Fast Recovery Area Disk Group Failure

Recovery Option	Recovery Time Objective (RTO)	Recovery Point Objective (RPO)
Local recovery (see Section , "Local Recovery for Fast Recovery Area Disk Group Failure")	Five minutes or less	Zero
Data Guard failover or switchover (see Section , "Data Guard Role Transition for Fast Recovery Area Disk Group Failure")	Seconds to minutes	Zero

Local Recovery for Fast Recovery Area Disk Group Failure If you decide to perform local recovery then you must perform a fast local restart to start the primary database after removing the controlfile member that is located in the fast recovery area from the init.ora and allocate another disk group as the fast recovery area for archiving.

When the Fast Recovery Area is lost, the database LOG_ARCHIVE_DEST_n location and the database DB_RECOVERY_FILE_DEST must be modified to point a surviving disk group; otherwise, the database hangs when the archive log location is unavailable. Also if guaranteed restore points are in place, the database crashes when the DB_RECOVERY_FILE_DEST becomes unavailable.

For a fast local restart, perform the following steps on the primary database:

1. Change the `CONTROL_FILES` initialization parameter to specify only the members in the Data Area:

   ```
   ALTER SYSTEM SET CONTROL_FILES='+DATA/sales/control1.dbf' SCOPE=spfile;
   ```

2. Change local archive destinations and the fast recovery area to the local redundant, scalable destination:

   ```
   ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='+DATA' SCOPE=spfile;
   ```

3. Start the database with the new settings:

   ```
   STARTUP MOUNT:
   ```

4. Drop the redo log members that were in the lost disk group. For example, issue the following command:

```
ALTER DATABASE DROP LOGFILE MEMBER '+RECO/dbm/onlinelog/group_2.258.750768395';
```

5. If the flashback logs were damaged or lost, it may be necessary to disable and re-enable Flashback Database:

```
ALTER DATABASE FLASHBACK OFF;
ALTER DATABASE FLASHBACK ON;
ALTER DATABASE OPEN;
```

   However, this is a temporary fix until you create a fast recovery area to replace the failed storage components. Oracle recommends using the Local Recovery Steps.

6. Once the failed Fast Recovery Area disk group comes back up or gets recreated, you must set the LOG_ARCHIVE_DEST_1 and the DB_RECOVERY_FILE_DEST location back to the original Fast Recovery Area disk group and transfer the archive logs and the flashback logs back to the Fast Recovery Area disk group.

See Also:

- My Oracle Support note 1339373.1 "Operational Steps for Recovery after Losing a Disk Group in an Exadata Environment" at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1339373.1 for more details and options for avoidance.
- Section , "Data Guard Role Transition for Fast Recovery Area Disk Group Failure Local Recovery Steps"

Data Guard Role Transition for Fast Recovery Area Disk Group Failure If you decide to perform a Data Guard role transition then the recovery time objective (RTO) can be expressed in terms of seconds or minutes, depending on the presence of the Data Guard observer process and fast-start failover.

If the protection level is maximum performance or the standby database is *unsynchronized* with the primary database, then:

1. Temporarily start the primary database by removing the controlfile member and pointing to a temporary fast recovery area (file system) in the SPFILE.
2. Perform a Data Guard switchover to ensure no data loss.
3. After the switchover has completed and the application is available, resolve the fast recovery area disk group failure.
4. Shut down the affected database and continue by using the instructions in the Local Recovery Steps to resolve the Oracle ASM disk group failure. For more information, see "Data Guard Role Transition for Fast Recovery Area Disk Group Failure Local Recovery Steps".

Data Guard Role Transition for Fast Recovery Area Disk Group Failure Local Recovery Steps

Local Recovery Steps

> **Note:** If you performed an Oracle Data Guard failover to a new primary database, then you cannot use this procedure to reintroduce the original primary database as a standby database. This is because Flashback Database log files that are required as part of reintroducing the database have been lost. You must perform a full reinstatement of the standby database.

1. Replace or get access to storage to use for a fast recovery area
2. Rebuild the Oracle ASM disk group using the storage location by issuing the following SQL*Plus statement:

```
SQL> CREATE DISKGROUP RECO NORMAL REDUNDANCY DISK 'path1','path2',...force;
```

3. Start the database instance NOMOUNT using the following RMAN command:

```
RMAN> STARTUP FORCE NOMOUNT;
```

4. Restore the control file from the surviving copy located in the data area:

```
RMAN> RESTORE CONTROLFILE FROM 'data_area_controlfile';
```

5. Start the database instance MOUNT:

```
RMAN> STARTUP FORCE MOUNT;
```

6. If you use Flashback Database, then disable it with the following SQL*Plus statement:

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

7. Open the database and allow instance recovery to complete:

```
SQL> ALTER DATABASE OPEN;
```

8. Issue the following statements only if Flashback Database is required:

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE OPEN;
```

9. Re-create the log file members on the failed Oracle ASM disk group:

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER 'filename';
SQL> ALTER DATABASE ADD LOGFILE MEMBER 'disk_group' TO GROUP group_no;
```

10. Synchronize the control file and the fast recovery area using the following RMAN commands:

```
RMAN> CATALOG RECOVERY AREA;
RMAN> CROSSCHECK ARCHIVELOG ALL;
RMAN> CROSSCHECK BACKUPSET;
RMAN> CROSSCHECK DATAFILECOPY ALL;
RMAN> LIST EXPIRED type;
RMAN> DELETE EXPIRED type;
```

In the example, the *type* variable is a placeholder for both `LIST EXPIRED BACKUP` and `LIST EXPIRED COPY` commands, and also for the `DELETE EXPIRED BACKUP` and `DELETE EXPIRED COPY` commands. You should run all of these commands now.

11. Assuming that data has been lost, perform a backup:

```
RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

Recovering from Data Corruption

A data block is corrupted when it is not in a recognized Oracle Database format, or its contents are not internally consistent. Data block corruption can damage internal Oracle control information or application and user data, leading to crippling loss of critical data and services. The Oracle Database corruption prevention, detection, and repair capabilities are built on internal knowledge of the data and transactions it protects, and on the intelligent integration of its comprehensive high availability solutions. For more information, see Section , "Protect Against Data Corruption".

The recovery process begins when you either suspect or discover a block corruption (for example: `ORA-1578`, `ORA-752`, `ORA-600 [3020]`, and `ORA-753`). Once a corrupt block is found, Oracle provides various techniques for recovering from most block corruptions.

There are various techniques for recovering data blocks, including:

- Data Recovery Advisor is the easiest way to diagnose and repair most problems. For more information, see Section , "Use Data Recovery Advisor".
- Active Data Guard can automatically repair corrupt data blocks in a primary or standby database. For more information, see Section , "Use Active Data Guard".
- RMAN block media recovery can repair individual corrupted blocks by retrieving the blocks from good backups. For more information, see Section , "Use RMAN and Block Media Recovery".
- Data Guard switchover or failover to a standby database. For more information, see Section , "Extracting Data from a Physical Standby Databases".
- Data File Media Recovery with RMAN. For more information, see Section , "Use RMAN and Block Media Recovery"
- When encountering lost write corruptions that result in `ORA-752` or `ORA-600 [3020]`, follow the guidelines in "Resolving ORA-752 or ORA-600 [3020] During Standby Recovery" in My Oracle Support Note 1265884.1 at

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1265884.1`

Furthermore, Data Guard broker has a new `PrimaryLostWriteAction` property that helps automate specific actions whenever standby database detects that a lost write has occurred at the primary database. For example, If fast-start failover is enabled (in either maximum performance or maximum availability mode, and the Data Guard broker `PrimaryLostWriteAction` is set to `FORCEFAILOVER`, then the observer initiates a failover. This option results in a data loss failover but can reduce downtime significantly. The `FORCEFAILOVER` option is available only in Oracle Database 12*c* Release 1 (12.1.0.2) and later.

Whatever method you use to recover corrupted blocks, you first must analyze the type and degree of corruption to perform the recovery. Implementing the optimal techniques to prevent and prepare for data corruptions can save time, effort, and stress when dealing with the possible consequences-lost data and downtime.

MAA best practices provide a step-by-step process for resolving most corruptions and stray or lost writes, including the following:

1. Use Data Recovery Advisor
2. Use Active Data Guard
3. Use RMAN and Block Media Recovery
4. Perform a Data Guard Role Transition
5. Use RMAN and Data File Media Recovery

See Also:

- For more information see the "Preventing, Detecting, and Repairing Block Corruption: Database 11g" MAA white paper from the MAA Best Practices area for Oracle Database at

  `http://www.oracle.com/goto/maa`

- See, "Best Practices for Corruption Detection, Prevention, and Automatic Repair - in a Data Guard Configuration" in My Oracle Support Note 1302539.1 at

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1302539.1`

Use Data Recovery Advisor

Data Recovery Advisor enables you to perform restore operations and recovery procedures or use Flashback Database as follows:

- Perform block media recovery of data files that have corrupted blocks
- Perform point-in-time recovery of the database or selected tablespaces
- Rewind the entire database with Flashback Database
- Completely restore and recover the database from a backup

Data Recovery Advisor has both a command-line and GUI interface. The GUI interface is available when you click Perform Recovery with Oracle Enterprise Manager Database Control (Support Workbench); this allows you use Data Recovery Advisor.

Using the RMAN command-line interface, the Data Recovery Advisor commands include: `LIST FAILURE`, `ADVISE FAILURE`, `REPAIR FAILURE`, and `CHANGE FAILURE`.

If the Data Recover Advisor fixes the problem, then there is no need to continue with any further recovery methods. However, continue to periodically check the alert log for any ORA- errors and any corruption warnings on the primary and standby databases. While the database is operational and corruption is detected, corruption errors are recorded as `ORA-600` or `ORA-01578` in the alert log.

Note: In the current release, Data Recovery Advisor only supports single-instance databases. Oracle RAC databases are not supported. See *Oracle Database Backup and Recovery User's Guide* for more information about Data Recovery Advisor supported database configurations.

See Also:

- Section , "Use Data Recovery Adviser to Detect, Analyze and Repair Data Failures"
- *Oracle Database 2 Day DBA* documentation for details on how to use the GUI interface for Data Recovery Advisor
- *Oracle Database Backup and Recovery User's Guide* for more information about Diagnosing and Repairing Failures with Data Recovery Advisor

Use Active Data Guard

There are two options for using a standby database to repair block corruption on the primary database:

- Oracle Active Data Guard and Automatic Block Repair
- Extracting Data from a Physical Standby Databases

Alternatively, if the corruption is widespread, you may choose to failover or switchover to the standby database while you make repairs to the primary database. For more information, see Section , "Perform a Data Guard Role Transition".

Oracle Active Data Guard and Automatic Block Repair Starting in Oracle Database 11*g* Release 2 (11.2), the primary database automatically attempts to repair a corrupted block in real time by fetching a good version of the same block from a physical standby database. This capability is referred to as Automatic Block Repair; Automatic Block Repair allows corrupt data blocks to be automatically repaired as soon as the corruption is detected. Automatic Block Repair reduces the amount of time that data is inaccessible due to block corruption and reduces block recovery time by using up-to-date good blocks in real-time, as opposed to retrieving blocks from disk or tape backups, or from Flashback logs.

Thus, with Automatic Block Repair you use an Oracle Active Data Guard standby database for automatic repair of data corruptions detected by the primary database. Additionally if the corruption is discovered on an Active Data Guard physical standby database the corruption is automatically repaired with a good block from the Primary. Both of these operations are transparent to the applications.

Note: Automatic Block Repair requires the use of the Oracle Active Data Guard option.

See Also: *Oracle Data Guard Concepts and Administration* for more information about Oracle Active Data Guard option and the Automatic Block Repair feature

Extracting Data from a Physical Standby Databases You can use a Data Guard physical standby database to repair data file wide block corruption on the primary database by replacing the corrupted data files with good copies from the standby database. Once the files are restored on the primary database, data file or tablespace recovery makes the data files consistent with the rest of the database.

See Also: *Oracle Data Guard Concepts and Administration* for information about Recovery from Loss of Data Files on the Primary Database Using Files On a Standby Database

Optionally Failing Over to Standby Database Automatically/Manually due to Corruptions

Wide-spread block corruptions (physical or logical) or lost writes likely indicate a significant hardware issue at the primary site. In this case a failover to a standby database might be the most prudent course of action in order to maintain availability and minimize potential data loss.

Use the Data Guard Broker property PrimaryLostWriteAction to predetermine the action taken in the event a standby database detects a lost write in a fast-start failover configuration.

Use RMAN and Block Media Recovery

Block media recovery recovers one block or a set of data blocks marked "media corrupt" in a data file by using the RMAN `RECOVER BLOCK` command. When a small number of data blocks are marked media corrupt and require media recovery, you can selectively restore and recover damaged blocks rather than whole data files. Block media recovery minimizes redo application time and avoids I/O overhead during recovery. Block media recovery also enables affected data files to remain online during recovery of the corrupt blocks. The corrupt blocks, however, remain unavailable until they are completely recovered.

Use block media recovery when:

- A small number of blocks require media recovery and you know which blocks need recovery.
- Blocks are marked corrupt (you can verify this with the RMAN `VALIDATE CHECK LOGICAL` command).
- The backup file for the corrupted data file is available locally or can be retrieved from a remote location.

> **Note:** Do not use block media recovery to recover from user errors or software bugs that cause logical corruption where the data blocks are intact.

If a significant portion of the data file is corrupt or if the amount of corruption is unknown, then use either RMAN to restore the file from a backup or switch to an on disk image copy, or switchover to your Data Guard standby database.

When corruption is detected, recover the block through the Oracle Enterprise Manager Restore and Recovery Wizard or directly with RMAN. For example, to recover a specific corrupt block using RMAN block media recovery:

```
RMAN> RECOVER BLOCK DATAFILE 7 BLOCK 3;
```

After a corrupt block is repaired, the row identifying this corrupted block is deleted from the `V$DATABASE_BLOCK_CORRUPTION` view.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for information about RMAN's block media recovery

Perform a Data Guard Role Transition

If the primary database corruption is widespread due to a bad controller or other hardware or software problem, then you may want to failover or switchover to the standby database while repairs to the primary database server are made. Use Data Guard switchover or failover for data corruption or data failure when:

- The database is down or when the database is up but the application is unavailable because of data corruption or failure, and the time to restore and recover locally is long or unknown.
- Recovering locally takes longer than the business service-level agreement or RTO. (We can automate this with FSFO and Broker attributes for lost writes and triggering on ORA-1578s.)

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about Data Guard failovers and switchovers

Use RMAN and Data File Media Recovery

When you cannot use any of the following methods to resolve corruptions and stray or lost writes, then use RMAN traditional media recovery:

- Use Data Recovery Advisor
- Use Active Data Guard
- Use RMAN and Block Media Recovery
- Perform a Data Guard Role Transition

> **Note:** If you do not have a Data Guard Physical standby, then you must use traditional media recovery. Using traditional media recovery, a backup copy of one or more files is restored and then data file, tablespace, or database recovery brings the database back to a consistent state.

Data file media recovery affects an entire data file or set of data files for a database by using the RMAN `RECOVER` command. When a large or unknown number of data blocks are marked "media corrupt" and require media recovery, or when an entire file is lost, you must restore and recover the applicable data files.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for information about Advanced User-Managed Recovery Scenarios

Recovering from Human Error (Recovery with Flashback)

Oracle Flashback technology revolutionizes data recovery. Before Flashback technology, it took seconds to damage a database but from hours to days to recover it. With Flashback technology, the time to correct errors can be as short as the time it took to make the error. Fixing human errors that require rewinding the database, table, transaction, or row level changes to a previous point in time is easy and does not require any database or object restoration. Flashback technology provides fine-grained analysis and repair for localized damage such as erroneous row deletion. Flashback technology also enables correction of more widespread damage such as accidentally running the wrong application batch job. Furthermore, Flashback technology is exponentially faster than a database restoration.

Flashback technologies are applicable only to repairing the following human errors:

- Erroneous or malicious update, delete, or insert transactions
- Erroneous or malicious `DROP TABLE` statements
- Erroneous or malicious batch job or wide-spread application errors

Flashback technologies cannot be used for media or data corruption such as block corruption, bad disks, or file deletions. See Section , "Database Failover with a Standby Database" to repair these outages.

> **Note:** For information about Flashback Database configuration best practices, see Section , "Enable Flashback Database"

Table 12–6 summarizes the Flashback solutions for outage varying in scope from destroying a row, such as through a bad update, to destroying a whole database (such as by deleting all the underlying files at the operating system level).

Table 12–6 Flashback Solutions for Different Outages

Outage Scope	Examples of Human Errors	Flashback Solutions	See Also
Row or transaction	Accidental deletion of row Erroneous transaction	Flashback Query Flashback Version Query Flashback Transaction Query Flashback Transaction	**See Also:** Section , "Resolving Row and Transaction Inconsistencies"
Table	Dropped table Erroneous transactions affecting one table or a set of tables	Flashback Drop Flashback Table	**See Also:** Section , "Resolving Table Inconsistencies"
Tablespace or database	Erroneous batch job affecting many tables or an unknown set of tables Series of database-wide malicious transactions	Enable Flashback Database or use multiple Flashback Table commands	**See Also:** Section , "Resolving Database-Wide Inconsistencies"
Single tablespace or a subset of tablespaces	Erroneous transactions affecting a small number of tablespaces	RMAN Tablespace Point-in-Time Recovery (TSPITR)	**See Also:** Section , "Resolving One or More Tablespace Inconsistencies"

Table 12–7 summarizes each Flashback feature.

Table 12–7 Summary of Flashback Features

Flashback Feature	Description	Changes are propagated to ...
Flashback Query	Flashback Query enables you to view data at an earlier point in time. You can use it to view and reconstruct lost data that was deleted or changed by accident. Developers can use this feature to build self-service error correction into their applications, empowering end users to undo and correct their errors.	Physical and logical standby databases
Flashback Version Query	Flashback Version Query uses undo data stored in the database to view the changes to one or more rows along with all the metadata of the changes.	Physical and logical standby databases
Flashback Transaction Query	Flashback Transaction Query enables you to examine changes to the database at the transaction level. As a result, you can diagnose problems, perform analysis, and audit transactions.	Physical and logical standby databases
Flashback Transaction	Flashback Transaction provides a way to roll back one or more transactions and their dependent transactions, while the database remains online.	Physical and logical standby databases

Table 12–7 (Cont.) Summary of Flashback Features

Flashback Feature	Description	Changes are propagated to ...
Flashback Drop	Flashback Drop provides a way to restore accidentally dropped tables.	Physical standby databases
Flashback Table	Flashback Table enables you to quickly recover a table to an earlier point in time without restoring a backup.	Physical and logical standby databases
Flashback Database	Flashback Database enables you to quickly return the database to an earlier point in time by undoing all of the changes that have taken place since that time. This operation is fast because you do not have to restore the backups.	Physical and logical standby databases

Flashback Database uses the Oracle Database flashback logs, while all other features of flashback technology use the Oracle Database unique undo and multiversion read consistency capabilities. For more information, see the configuration best practices for the database, as documented in Section , "Database Configuration High Availability and Fast Recoverability Best Practices" to configure Flashback technologies to ensure that the resources from these solutions are available at a time of failure.

See Also:

- *Oracle Database Administrator's Guide* for information about Recovering Tables Using Oracle Flashback Table
- *Oracle Database Backup and Recovery User's Guide* for information about Using Flashback Database and Restore Points
- *Oracle Database Concepts* for information about Oracle Flashback Technology

In general, the recovery time when using Flashback technologies is equivalent to the time it takes to cause the human error plus the time it takes to detect the human error.

Flashback technologies allow recovery up to the point that the human error occurred.

Use the following recovery methods:

- Resolving Table Inconsistencies
- Resolving Row and Transaction Inconsistencies
- Resolving Database-Wide Inconsistencies
- Resolving One or More Tablespace Inconsistencies

Resolving Table Inconsistencies

Dropping or deleting database objects by accident is a common mistake. Users soon realize their mistake, but by then it is too late and there has been no way to easily recover the dropped tables and its indexes, constraints, and triggers. Objects once dropped were dropped forever. Loss of very important tables or other objects (like indexes, partitions or clusters) required DBAs to perform a point-in-time recovery, which can be time-consuming and lead to loss of recent transactions.

Oracle provides the following statements to help resolve table inconsistencies:

- `Flashback Table` statement to restore a table to a previous point in the database
- `Flashback Drop` statement to recover from an accidental `DROP TABLE` statement

- Flashback Transaction statement to roll back one or more transactions and their dependent transactions, while the database remains online

Flashback Table

Flashback Table provides the ability to quickly recover a table or a set of tables to a specified point in time. In many cases, Flashback Table alleviates the more complicated point-in-time recovery operations. For example:

```
FLASHBACK TABLE orders, order_items
      TO TIMESTAMP
      TO_DATE('28-Jun-11 14.00.00','dd-Mon-yy hh24:mi:ss');
```

This statement rewinds any updates to the ORDERS and ORDER_ITEMS tables that have been done between the current time and a specified timestamp in the past. Flashback Table performs this operation online and in place, and it maintains referential integrity constraints between the tables.

Flashback Drop

Flashback Drop provides a safety net when dropping objects. When a user drops a table, Oracle places it in a recycle bin. Objects in the recycle bin remain there until the user decides to permanently remove them or until space limitations begin to occur on the tablespace containing the table. The recycle bin is a virtual container where all dropped objects reside. Users view the recycle bin and undrop the dropped table and its dependent objects. For example, the employees table and all its dependent objects would be undropped by the following statement:

```
FLASHBACK TABLE employees TO BEFORE DROP;
```

Flashback Transaction

Oracle Flashback Transaction increases availability during logical recovery by easily and quickly backing out a specific transaction or set of transactions and their dependent transactions, while the database remains online.

Use the DBMS_FLASHBACK.TRANSACTION_BACKOUT() PL/SQL procedure to roll back a transaction and its dependent transactions. This procedure uses undo data to create and execute the compensating transactions that return the affected data to its pre-transaction state.

See Also:

- *Oracle Database Advanced Application Developer's Guide* for information about Using Flashback Transaction
- DBMS_FLASHBACK.TRANSACTION_BACKOUT() in *Oracle Database PL/SQL Packages and Types Reference*

Resolving Row and Transaction Inconsistencies

Resolving row and transaction inconsistencies might require a combination of Flashback Query, Flashback Version Query, Flashback Transaction Query, and the compensating SQL statements constructed from undo statements to rectify the problem. This section describes a general approach using a human resources example to resolve row and transaction inconsistencies caused by erroneous or malicious user errors.

Flashback Query

Flashback Query enables an administrator or user to query any data from some earlier point in time. Use this feature to view and reconstruct data that might have been deleted or changed by accident.

Developers can use Flashback Query to build self-service error correction into their applications, empowering end users to undo and correct their errors without delay, and freeing database administrators from having to perform this task. Flashback Query is easy to manage because the database automatically keeps the necessary information to reconstruct data for a configurable time into the past.

The following partial statement displays rows from the `EMPLOYEES` table starting from 2:00 p.m. on June 28, 2011.

```
SELECT * FROM EMPLOYEES
       AS OF TIMESTAMP
       TO_DATE('28-Jun-11 14:00','DD-Mon-YY HH24:MI')
WHERE ...
```

Flashback Version Query

Flashback Version Query provides a way to view changes made to the database at the row level. Flashback Version Query is an extension to SQL and enables the retrieval of all the different versions of a row across a specified time interval. For example:

```
SELECT * FROM EMPLOYEES
       VERSIONS BETWEEN TIMESTAMP
       TO_DATE('28-Jun-11 14:00','dd-Mon-YY hh24:mi') AND
       TO_DATE('28-Jun-11 15:00','dd-Mon-YY hh24:mi')
WHERE ...
```

This statement displays each version of the row, each entry changed by a different transaction, between 2 and 3 p.m. on June 28, 2011. A database administrator can use this to pinpoint when and how data is changed and trace it back to the user, application, or transaction. Flashback Version Query enables the database administrator to track down the source of a logical corruption in the database and correct it. It also enables application developers to debug their code.

Flashback Transaction Query

Flashback Transaction Query provides a way to view changes made to the database at the transaction level. Flashback Transaction Query is an extension to SQL that enables you to see all changes made by a transaction. For example:

```
SELECT UNDO_SQL
FROM FLASHBACK_TRANSACTION_QUERY
WHERE XID = '000200030000002D';
```

This statement shows all of the changes that resulted from this transaction. In addition, compensating SQL statements are returned and can be used to undo changes made to all rows by this transaction. Using a precision tool like Flashback Transaction Query, the database administrator and application developer can precisely diagnose and correct logical problems in the database or application.

Consider a human resources (HR) example involving the `SCOTT` schema. The HR manager reports to the database administrator that there is a potential discrepancy in Ward's salary. Sometime before 9:00 a.m., Ward's salary was increased to $1875. The HR manager is uncertain how this occurred and wishes to know when the employee's salary was increased. In addition, he instructed his staff to reset the salary to the previous level of $1250. This was completed around 9:15 a.m.

The following steps show how to approach the problem.

1. Assess the problem.

   Fortunately, the HR manager has provided information about the time when the change occurred. You can query the information as it was at 9:00 a.m. using Flashback Query.

```
SELECT EMPNO, ENAME, SAL
FROM EMP
AS OF TIMESTAMP TO_DATE('28-JUN-11 09:00','dd-Mon-yy hh24:mi')
WHERE ENAME = 'WARD';

     EMPNO ENAME             SAL
---------- ---------- ----------
      7521 WARD             1875
```

   You can confirm that you have the correct employee by the fact that Ward's salary was $1875 at 09:00 a.m. Rather than using Ward's name, you can now use the employee number for subsequent investigation.

2. Query previous rows or versions of the data to acquire transaction information.

   Although it is possible to restrict the row version information to a specific date or SCN range, you might want to query all the row information that is available for the employee WARD using Flashback Version Query.

```
SELECT EMPNO, ENAME, SAL, VERSIONS_STARTTIME, VERSIONS_ENDTIME, VERSIONS_XID
    FROM EMP
    VERSIONS BETWEEN TIMESTAMP MINVALUE AND MAXVALUE
    WHERE EMPNO = 7521
    ORDER BY NVL(VERSIONS_STARTSCN,1);

EMPNO ENAME  SAL  VERSIONS_STARTTIME     VERSIONS_ENDTIME      VERSIONS_XID
----- ------ ---  ---------------------- --------------------- ---------------
 7521 WARD  1250  28-JUN-11 08.48.43 AM  28-JUN-11 08.54.49 AM 0006000800000086
 7521 WARD  1875  28-JUN-11 08.54.49 AM  28-JUN-11 09.10.09 AM 0009000500000089
 7521 WARD  1250  28-JUN-11 09.10.09 AM                        000800050000008B
```

   You can see that WARD's salary was increased from $1250 to $1875 at 08:54:49 the same morning and was subsequently reset to $1250 at approximately 09:10:09.

   Also, you can see that the ID of the erroneous transaction that increased WARD's salary to $1875 was "0009000500000089".

3. Query the erroneous transaction and the scope of its effect.

   With the transaction information (`VERSIONS_XID` pseudocolumn), you can now query the database to determine the scope of the transaction, using Flashback Transaction Query.

```
    SELECT UNDO_SQL
    FROM FLASHBACK_TRANSACTION_QUERY
    WHERE XID = HEXTORAW('0009000500000089');

    UNDO_SQL

--------------------------------------------------------------------------------
   update "SCOTT"."EMP" set "SAL" = '950' where ROWID = 'AAACV4AAFAAAAKtAAL';
   update "SCOTT"."EMP" set "SAL" = '1500' where ROWID = 'AAACV4AAFAAAAKtAAJ';
   update "SCOTT"."EMP" set "SAL" = '2850' where ROWID = 'AAACV4AAFAAAAKtAAF';
   update "SCOTT"."EMP" set "SAL" = '1250' where ROWID = 'AAACV4AAFAAAAKtAAE';
   update "SCOTT"."EMP" set "SAL" = '1600' where ROWID = 'AAACV4AAFAAAAKtAAB';
```

```
6 rows selected.
```

You can see that WARD's salary was not the only change that occurred in the transaction. Now you can send the information that was changed for the other four employees at the same time as employee WARD, back to the HR manager for review.

4. Determine if the corrective statements should be executed.

   If the HR manager decides that the corrective changes suggested by the UNDO_SQL column are correct, then the database administrator can execute the statements individually.

5. Query the FLASHBACK_TRANSACTION_QUERY view for additional transaction information. For example, to determine the user that performed the erroneous update, issue the following query:

```
SELECT LOGON_USER FROM FLASHBACK_TRANSACTION_QUERY
WHERE XID = HEXTORAW('0009000500000089');

LOGON_USER
----------------------------
MSMITH
```

   In this example, the query shows that the user MSMITH was responsible for the erroneous transaction.

Resolving Database-Wide Inconsistencies

To bring an Oracle database to a previous point in time, the traditional method is point-in-time recovery. However, point-in-time recovery can take hours or even days, because it requires the whole database to be restored from backup and recovered to the point in time just before the error was introduced into the database. With the size of databases constantly growing, it takes hours or even days just to restore the whole database.

Flashback Database is a strategy for doing point-in-time recovery. Flashback Database quickly rewinds an Oracle database to a previous time to correct any problems caused by logical data corruption or user error. Flashback logs are used to capture old versions of changed blocks. When recovery must be performed the flashback logs are quickly replayed to restore the database to a point in time before the error and just the changed blocks are restored. Flashback Database is extremely fast and reduces recovery time from hours to minutes. In addition, it is easy to use. A database can be recovered to 2:05 p.m. by issuing a single statement. Before the database can be recovered, all instances of the database must be shut down and one instance subsequently mounted. The following is an example of a FLASHBACK DATABASE statement.

```
FLASHBACK DATABASE TO TIMESTAMP SYSDATE-1;
```

No restoration from tape, no lengthy downtime, and no complicated recovery procedures are required to use it. You can also use Flashback Database and then open the database in read-only mode and examine its contents. If you determine that you flashed back too far or not far enough, then you can reissue the FLASHBACK DATABASE statement or continue recovery to a later time to find the proper point in time before the database was damaged. Flashback Database works with a primary database, a physical standby database, or a logical standby database.

These steps are recommended for using Flashback Database:

1. Determine the time or the SCN to which to flash back the database.
2. Verify that there is sufficient flashback log information.

```
SELECT OLDEST_FLASHBACK_SCN,
  TO_CHAR(OLDEST_FLASHBACK_TIME, 'mon-dd-yyyy HH:MI:SS')
  FROM V$FLASHBACK_DATABASE_LOG;
```

3. Flash back the database to a specific time or SCN. (The database must be mounted to perform a Flashback Database.)

```
FLASHBACK DATABASE TO SCN scn;
```

or

```
FLASHBACK DATABASE TO TIMESTAMP TO_DATE date;
```

4. Open the database in read-only mode to verify that it is in the correct state.

```
ALTER DATABASE OPEN READ ONLY;
```

If more flashback data is required, then issue another `FLASHBACK DATABASE` statement. (The database must be mounted to perform a Flashback Database.)

If you want to move forward in time, then issue a statement similar to the following:

```
RECOVER DATABASE UNTIL [TIME date | CHANGE scn];
```

5. Open the database:

```
ALTER DATABASE OPEN RESETLOGS;
```

Other considerations when using Flashback Database are as follows:

- If there are not sufficient flashback logs to flash back to the target time, then use an alternative:
  - Use Data Guard to recover to the target time if the standby database lags behind the primary database or flash back to the target time if there's sufficient flashback logs on the standby.
  - Restore from backups.
- After flashing back a database, any dependent database such as a standby database must be flashed back. See Section , "Restoring Fault Tolerance".

Flashback Database does not automatically fix a dropped tablespace, you can use Flashback Database to significantly reduce the downtime. You can flash back the primary database to a point before the tablespace was dropped and then restore a backup of the corresponding data files using `SET NEWNAME` from the affected tablespace and recover to a time before the tablespace was dropped.

Resolving One or More Tablespace Inconsistencies

Recovery Manager (RMAN) automatic tablespace point-in-time recovery (TSPITR) enables you to quickly recover one or more tablespaces in a database to an earlier time without affecting the rest of the tablespaces and objects in the database. You can only use TSPITR on tablespaces whose data is completely segregated from the rest of the database. This usually means that TSPITR is something for which you must plan in advance.

RMAN TSPITR is most useful for the following situations:

- To recover a logical database to a point different from the rest of the physical database, when multiple logical databases exist in separate tablespaces of one physical database. For example, you maintain logical databases in the Orders and Personnel tablespaces. An incorrect batch job or DML statement corrupts the data in only one tablespace.
- To recover data lost after DDL operations that change the structure of tables. You cannot use Flashback Table to rewind a table to before the point of a structural change such as a truncate table operation.
- To recover a table after it has been dropped with the `PURGE` option.
- To recover from the logical corruption of a table.

You perform TSPITR by using the RMAN `RECOVER TABLESPACE` command.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for detailed information about performing RMAN TSPITR

Recovering Databases in a Distributed Environment

Some applications may update multiple databases and participate in distributed transactions. Global consistency between the participating databases may be expected and crucial to the application.

If one database in a distributed database environment requires recovery to an earlier time, it is often necessary to recover all other databases in the configuration to the same point in time when global data consistency is required by the application.

To achieve coordinated, time-based, distributed database recovery:

1. Recover the database that requires the recovery operation using time-based recovery.

   For example, if a database must be recovered because of a media failure, then recover this database first using time-based recovery. Do not recover the other databases at this point.

2. After you have recovered the database and opened it with the `RESETLOGS` option, search the `alert_SID.log` of the database for the `RESETLOGS` message. Your next step depends on the message that you find in the log file, as described in following table:

If the message returned is ...	Then ...
"`RESETLOGS` after complete recovery through change *nnn*"	Recovery is complete. You have applied all the changes in the database and performed complete recovery. Do not recover any of the other databases in the distributed system because this unnecessarily removes database changes.
"`RESETLOGS` after incomplete recovery `UNTIL CHANGE` *nnn*"	You have successfully performed an incomplete recovery. Record the change number from the message and proceed to the next step.

3. Recover or flash back all other databases in the distributed database system using change-based recovery, specifying the change number (SCN) that you recorded in Step 2.

Note: If a database that is participating in distributed transactions fails, in-doubt distributed transactions may exist in the participating databases. If the failed database recovers completely and communications resume between the databases, then the in-doubt transactions is automatically resolved by the Oracle recoverer process (RECO) process. If you cannot wait until the failed database becomes available, you can also manually commit or rollback in-doubt transactions.

Oracle Site Guard orchestrates and automates the coordinated failover of Oracle Fusion Middleware, Oracle Fusion Applications, and Oracle Databases. It is also extensible to include other data center software components. Oracle Site Guard integrates with underlying replication mechanisms that synchronize primary and standby environments and protect mission critical data. It comes with a built-in support for Oracle Data Guard for Oracle database, and Oracle Sun ZFS. Oracle Site Guard can also support other storage replication technologies.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more information about performing time-based recovery
- *Oracle Database Administrator's Guide* for information about how to handle in-doubt transactions and about recovery from distributed transaction failures
- *Oracle Enterprise Manager Oracle Site Guard Administrator's Guide*
- For an additional methodology for recovering multiple Oracle databases to a consistent state with local and distributed database transactions, see My Oracle Support Note 1096993.1. The participating databases may be involved in distributed or remote transactions or can be completely independent but are required to be "synchronized" for application consistency. Siebel, Peoplesoft, SAP, and other custom applications that include multiple databases are real world examples that may require global consistency across multiple databases. For more information, see "Recovery for Global Consistency in an Oracle Distributed Database Environment ", in My Oracle Support Note 1096993.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1096993.1

Restoring Fault Tolerance

Whenever a component in a high availability architecture fails, the full protection, or fault tolerance, of the architecture is compromised and possible single points of failure exist until the component is repaired. Restoring the high availability architecture to full fault tolerance to reestablish full Oracle RAC, Data Guard, or MAA protection requires repairing the failed component. While full fault tolerance might be sacrificed during planned downtime, the method of repair is well understood because it is planned, the risk is controlled, and it ideally occurs at times best suited for continued application availability. However, for unplanned downtime the risk of exposure to a single point of failure must be clearly understood.

This section provides the following topics that describe the steps needed to restore database fault tolerance:

- For Oracle Database 11*g* with Oracle RAC or Oracle RAC One Node
  - Restoring Failed Nodes or Instances in Oracle RAC and Oracle RAC One Node
- For Oracle Database 11*g* with Data Guard and Oracle Database 11*g* with Oracle RAC and Data Guard - MAA
  - Restoring a Standby Database After a Failover
  - Restoring Oracle ASM Disk Groups after a Failure
  - Restoring Fault Tolerance After Planned Downtime on Secondary Site or Cluster
  - Restoring Fault Tolerance After a Standby Database Data Failure
  - Restoring Fault Tolerance After the Primary Database Was Opened Resetlogs
  - Restoring Fault Tolerance After Dual Failures

Restoring Failed Nodes or Instances in Oracle RAC and Oracle RAC One Node

Ensuring that application services fail over quickly and automatically in an Oracle RAC cluster—or between primary and secondary sites—is important when planning for both scheduled and unscheduled outages. Similarly, using Oracle RAC One Node you must ensure that applications failover to the new instance that starts if an Oracle RAC One Node instance fails. To ensure that the environment is restored to full fault tolerance after any errors or issues are corrected, it is also important to understand the steps and processes for restoring failed instances or nodes within an Oracle RAC cluster or databases between sites.

Adding a failed node back into the cluster or restarting a failed Oracle RAC instance or Oracle RAC One Node instance is easily done after the core problem that caused the specific component to originally fail has been corrected. However, you should also consider:

- When to perform these tasks to incur minimal or no effect on the current running environment
- Failing back or rebalancing existing connections

After the problem that caused the initial node or instance failure has been corrected, a node or instance can be restarted and added back into the Oracle RAC environment at any time. For an Oracle RAC One Node, you can also restart a failed instance and go back to running the instance on the original node. Processing to complete the reconfiguration of a node may require additional system resources.

Table 12–8 summarizes additional processing that may be required when adding a node.

Table 12–8 Additional Processing When Restarting or Rejoining a Node or Instance

Action	Additional Resources
Restarting a node or rejoining a node into a cluster	When using only Oracle Clusterware, there is no impact when a node joins the cluster. When using vendor clusterware, there may be performance degradation while reconfiguration occurs to add a node back into the cluster. The impact on current applications should be evaluated with a full test workload.
Restarting or rejoining of an Oracle RAC instance	When you restart an Oracle RAC instance, there might be some potential performance impact while lock reconfiguration takes place. The impact on current applications is usually minimal, but it should be evaluated with a full test workload.

Use the following recovery methods:

- Recovering Service Availability for Oracle RAC
- Recovering Service Availability for Oracle RAC One Node
- Considerations for Client Connections After Restoring an Oracle RAC Instance

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about restarting an Oracle RAC instance
- Your vendor-specified cluster management documentation for detailed steps on how to start and join a node back into a cluster

Recovering Service Availability for Oracle RAC

Note: These descriptions for recovering service availability are for Oracle RAC and do not apply for Oracle RAC One Node systems.

After a failed node has been brought back into the cluster and its instance has been started, Cluster Ready Services (CRS) automatically manages the virtual IP address used for the node and the services supported by that instance automatically. A particular service might or might not be started for the restored instance. The decision by CRS to start a service on the restored instance depends on how the service is configured and whether the proper number of instances are currently providing access for the service. A service is not relocated back to a preferred instance if the service is still being provided by an available instance to which it was moved by CRS when the initial failure occurred.

CRS restarts services on the restored instance if the number of instances that are providing access to a service across the cluster is less than the number of preferred instances defined for the service. After CRS restarts a service on a restored instance, CRS notifies registered applications of the service change.

For example, suppose the HR service is defined with instances A and B as preferred and instances C and D as available in case of a failure. If instance B fails and CRS starts the HR service on C automatically, then when instance B is restarted, the HR service remains at instance C. CRS does not automatically relocate a service back to a preferred instance.

Suppose a different scenario in which the HR service is defined with instances A, B, C, and D as preferred and no instances defined as available, spreading the service across

all nodes in the cluster. If instance B fails, then the HR service remains available on the remaining three nodes. CRS automatically starts the HR service on instance B when it rejoins the cluster because it is running on fewer instances than configured. CRS notifies the applications that the HR service is again available on instance B.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide*

Recovering Service Availability for Oracle RAC One Node

Oracle RAC One Node databases are administered slightly differently from Oracle RAC or single-instance databases. For administrator-managed Oracle RAC One Node databases, you must monitor the candidate node list and make sure a server is always available for failover, if possible. Candidate servers reside in the Generic server pool and the database and its services will fail over to one of those servers.

For policy-managed Oracle RAC One Node databases, you must ensure that the server pools are configured such that a server will be available for the database to fail over to in case its current node becomes unavailable. Also, for policy-managed Oracle RAC One Node databases, the destination node for online database relocation must be located in the database's server pool.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for information about Administering Oracle RAC One Node

Considerations for Client Connections After Restoring an Oracle RAC Instance

After an Oracle RAC instance has been restored, additional steps might be required, depending on the current resource usage and system performance, the application configuration, and the network load balancing that has been implemented.

Existing connections, that might have failed over or started as a new session, on the surviving Oracle RAC instances are not automatically redistributed or failed back to an instance that has been restarted. Failing back or redistributing users might or might not be necessary, depending on the current resource utilization and the capability of the surviving instances to adequately handle and provide acceptable response times for the workload. If the surviving Oracle RAC instances do not have adequate resources to run a full workload or to provide acceptable response times, then it might be necessary to move (disconnect and reconnect) some existing user connections to the restarted instance.

> **Note:** In Oracle RAC One Node there is only one instance for a database (unless you are migrating). Thus an Oracle RAC One Node configuration does not require you to rethink the strategy for 'rebalancing' the connections as there is only one. Clients using Oracle RAC One Node must be able to work with FAN and other client and service facilities to be informed about the status of services.

Connections are started as they are needed, on the least-used node, assuming connection load balancing has been configured. Therefore, the connections are automatically load-balanced over time.

An application service can be:

- Managed with services running on a subset of Oracle RAC instances
- Nonpartitioned so that all services run equally across all nodes

This is valuable for modularizing application and database form and function while still maintaining a consolidated data set. For cases where an application is partitioned or has a combination of partitioning and nonpartitioning, you should consider the response time and availability aspects for each service.

If redistribution or failback of connections for a particular service is required, then you can rebalance workloads automatically using Oracle Universal Connection Pool (UCP). If you are using UCP, then connections are automatically redistributed to the new node.

Note: Oracle Universal Connection Pool (UCP) provides fast and automatic detection of connection failures and removes terminated connections for any Java application using, Fast Connection Failover, and FAN events

---

For load-balancing application services across multiple Oracle RAC instances, Oracle Net connect-time failover and connection load balancing are recommended. This feature does not require changes or modifications for failover or restoration. It is also possible to use hardware-based load balancers. However, there might be limitations in distinguishing separate application services (which is understood by Oracle Net Services) and restoring an instance or a node. For example, when a node or instance is restored and available to start receiving connections, a manual step might be required to include the restored node or instance in the hardware-based load balancer logic, whereas Oracle Net Services does not require manual reconfiguration.

Table 12–9 summarizes the considerations for new and existing connections after an instance has been restored. The considerations differ depending on whether the application services are partitioned, nonpartitioned, or are a combination of both. The actual redistribution of existing connections might or might not be required depending on the resource utilization and response times.

Table 12–9 Restoration and Connection Failback

Application Services	Failback or Restore Existing Connections	Failback or Restore New Connections
Partitioned	Existing sessions are not automatically relocated back to the restored instance. Use the `SRVCTL` utility to manually start, stop, and relocate services. See *Oracle Real Application Clusters Administration and Deployment Guide* "Administering Services" for more information.	Automatically routes to the restored instance by using the Oracle Net Services configuration.
Nonpartitioned	No action is necessary unless the load must be rebalanced, because restoring the instance means that the load there is low. If the load must be rebalanced, then the same problems are encountered as if application services were partitioned.	Automatically routes to the restored instance (because its load should be lowest) by using the Oracle Net Services configuration

Figure 12–7 shows a two-node partitioned Oracle RAC database. Each instance services a different portion of the application (HR and Sales). Client processes connect to the appropriate instance based on the service they require.

Figure 12–7 Partitioned Two-Node Oracle RAC Database

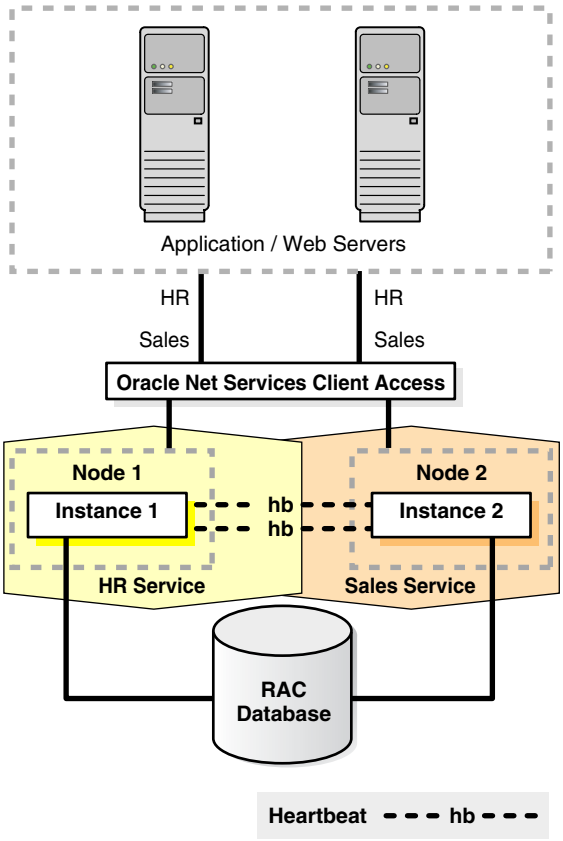
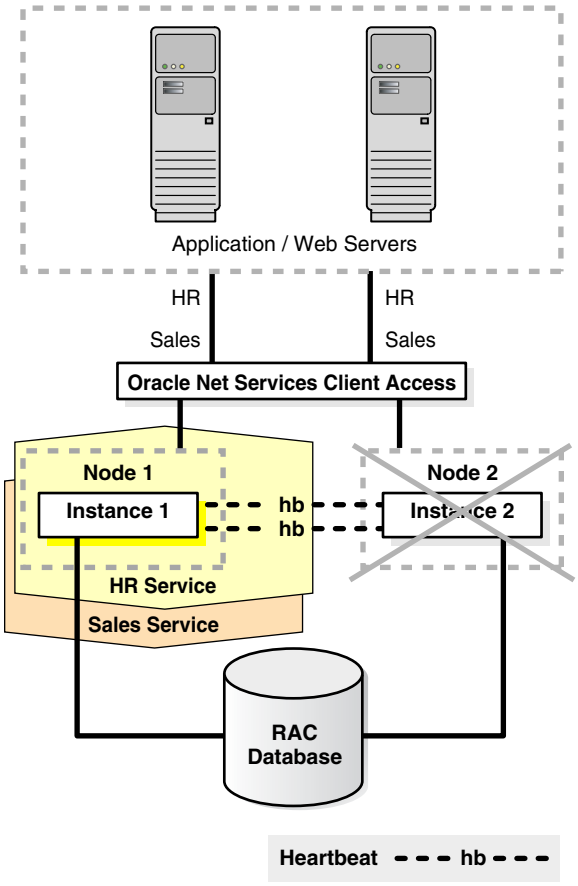


Figure 12–8 shows what happens when one Oracle RAC instance fails.

Figure 12–8 Oracle RAC Instance Failover in a Partitioned Database

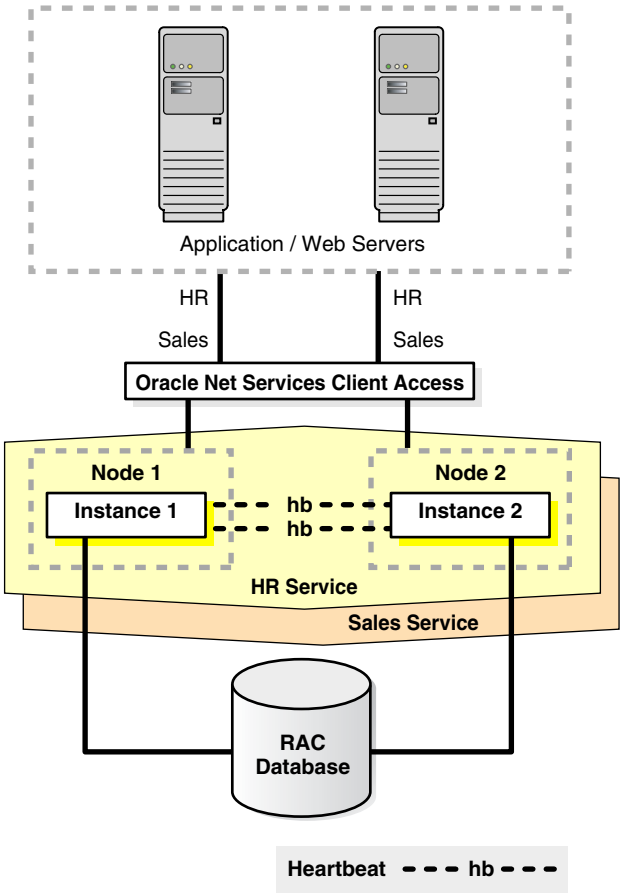


If one Oracle RAC instance fails, then the service and existing client connections can be automatically failed over to another Oracle RAC instance. In this example, the HR and Sales services are both supported by the remaining Oracle RAC instance. In addition, you can route new client connections for the Sales service to the instance now supporting this service.

After the failed instance has been repaired and restored to the state shown in Figure 12–7 and the Sales service is relocated to the restored instance, then you might need to identify and failback any failed-over clients and any new clients that had connected to the Sales service on the failed-over instance. Client connections that started after the instance has been restored should automatically connect back to the original instance. Therefore, over time, as older connections disconnect, and new sessions connect to the Sales service, the client load migrates back to the restored instance. Rebalancing the load immediately after restoration depends on the resource utilization and application response times.

Figure 12–9 shows a nonpartitioned application. Services are evenly distributed across both active instances. Each instance has a mix of client connections for both HR and Sales.

Figure 12–9 Nonpartitioned Oracle RAC Instances



If one Oracle RAC instance fails, then Oracle Clusterware moves the services that were running on the failed instance. If one Oracle RAC instance fails, new client connections are only accepted on the remaining instances that offers that service.

After the failed instance has been repaired and restored to the state shown in Figure 12–9, some clients might have to be moved back to the restored instance. For nonpartitioned applications, identifying appropriate services is not required for rebalancing the client load among all available instances. Also, this is necessary only if a single-instance database is not able to adequately service the requests.

Client connections that started after the instance has been restored should automatically connect back to the restored instance because it has a smaller load. Therefore, over time, as older connections disconnect and new sessions connect to the restored instance, the client load evenly balances again across all available Oracle RAC instances. Rebalancing the load immediately after restoration depends on the resource usage and application response times.

Restoring a Standby Database After a Failover

Following unplanned downtime on a primary database that requires a failover, full fault tolerance is compromised until the standby database is reestablished. Full database protection should be restored as soon as possible. The steps for restoring fault tolerance differ slightly between physical and logical standby databases.

Reinstating databases is automated if you are using Data Guard fast-start failover. After a fast-start failover completes, the observer automatically attempts to *reinstate*

the original primary database as a standby database. **Reinstatement** restores high availability to the broker configuration so that, if the new primary database fails, another fast-start failover can occur. The reinstated database can act as the fast-start failover target for the primary database, making a subsequent fast-start failover possible. The standby database is a viable target of a failover when it begins applying redo data received from the new primary database. If you want to prevent automatic reinstatement (for example, to perform diagnostic or repair work after failover has completed), set the `FastStartFailoverAutoReinstate` configuration property to `FALSE`.

The `FastStartFailoverAutoReinstate` configuration property controls whether the observer should automatically reinstate the original primary after a fast-start failover occurred because a fast-start failover was initiated due to the primary database being isolated for longer than the number of seconds specified by the `FastStartFailoverThreshold` property. In some cases, an automatic reinstatement might not be wanted until further diagnostic or recovery work is done.

To reinstate the original primary database, the database must be started and mounted, but it cannot be opened. The broker reinstates the database as a standby database of the same type (physical or logical) as the original standby database.

If the original primary database cannot be reinstated automatically, you can manually reinstate it using either the DGMGRL `REINSTATE` command or Enterprise Manager. Step-by-step instructions for manual reinstatement are described in *Oracle Data Guard Broker*.

Standby databases do not have to be re-created if you use the Oracle Flashback Database feature. Flashback Database has the following advantages:

- Saves hours of database restoration time
- Reduces overall complexity in restoring fault tolerance
- Reduces the time that the system is vulnerable because the standby database is re-created more quickly

See Also:

- *Oracle Data Guard Concepts and Administration* for information about Flashing Back a Failed Primary Database into a Physical Standby Database
- *Oracle Data Guard Concepts and Administration* for information about Flashing Back a Failed Primary Database into a Logical Standby Database

This section includes the following topics:

- Reinstating the Original Primary Database After a Fast-Start Failover
- Reinstating a Standby Database Using Enterprise Manager After a Failover

Reinstating the Original Primary Database After a Fast-Start Failover

Following a fast-start failover, the observer periodically attempts to reconnect to the original primary database. When the observer regains network access to the original primary database, it initiates a request for the broker to automatically reinstate it as a standby database to the new primary. This quickly restores disaster protection and high availability for the configuration.

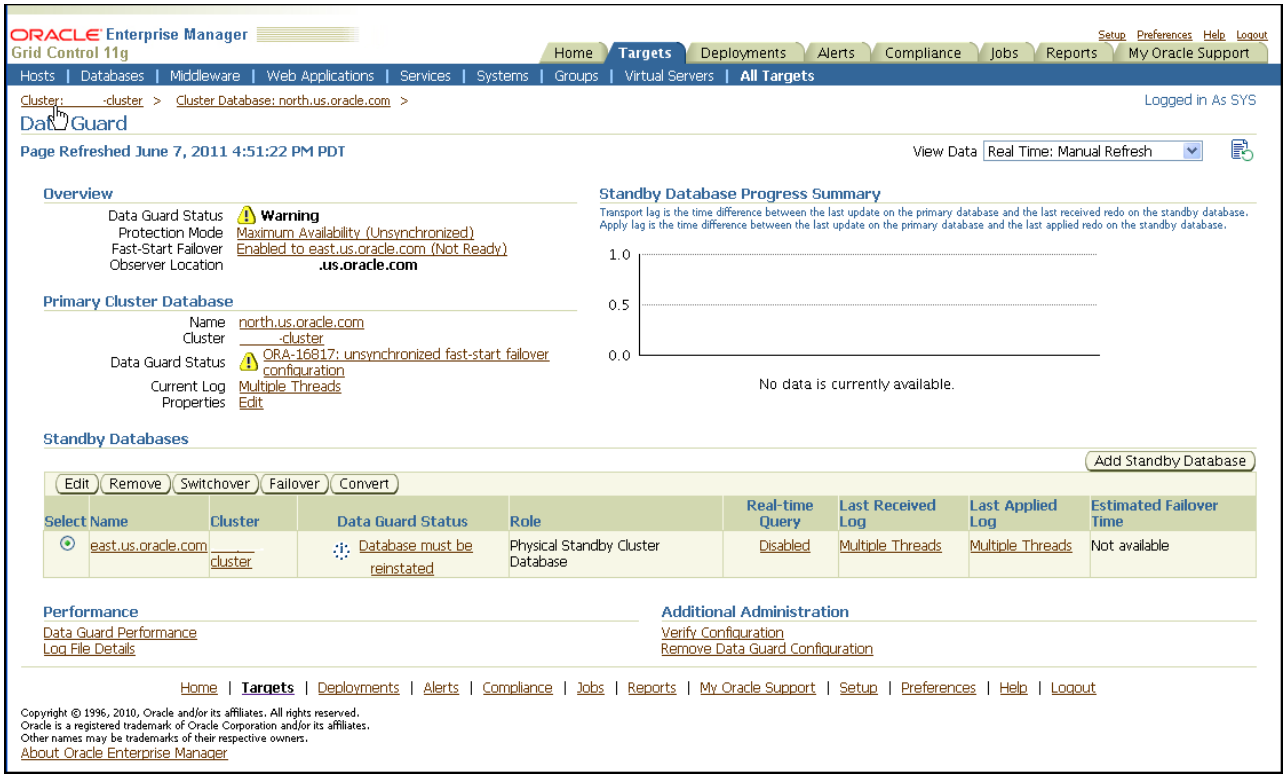
You can enable fast-start failover from any site, including the observer site, in Enterprise Manager while connected to any database in the broker configuration. The

broker simplifies switchovers and failovers by allowing you to invoke them using a single key click in Oracle Enterprise Manager.

Reinstating a Standby Database Using Enterprise Manager After a Failover

Furthermore, you can use Enterprise Manager to reinstate the original primary as the new standby. Figure 12–10 shows an example of the warning message that shows in Enterprise Manager when a reinstatement is needed.

Figure 12–10 Reinstating the Original Primary Database After a Fast-Start Failover



Restoring Oracle ASM Disk Groups after a Failure

Follow the steps in Section , "Data Area Disk Group Failure" or Section , "Fast Recovery Area Disk Group Failure".

Restoring Fault Tolerance After Planned Downtime on Secondary Site or Cluster

After performing the planned maintenance on the secondary site, the standby database and log apply services must be restarted, and then the Data Guard redo transport services automatically catch up the standby database with the primary database. You can use Enterprise Manager and the broker to monitor the Data Guard state.

The following steps are required to restore full fault tolerance after planned downtime on a secondary site or clusterwide outage:

Note: The following steps can be accomplished manually (as described below) or automatically using Enterprise Manager.

---

1. Start the standby database

   You might have to restore the standby database from local backups, local tape backups, or from the primary site backups if the data in the secondary site has been damaged. Re-create the standby database from the new primary database by following the steps for creating a standby database in *Oracle Data Guard Concepts and Administration*.

   After the standby database has been reestablished, start the standby database.

Table 12–10 SQL Statements for Starting Standby Databases

Type of Standby Database	SQL Statement
Physical	`STARTUP MOUNT;`
Logical	`STARTUP;`
Active Data Guard	`STARTUP;`

2. Start Redo Apply (physical standby) or SQL Apply (logical standby):

Table 12–11 SQL Statements to Start Redo Apply and SQL Apply

Type of Standby Database	SQL Statement
Physical (or Active Data Guard)	`RECOVER MANAGED STANDBY DATABASE DISCONNECT;`
Logical	`ALTER DATABASE START LOGICAL STANDBY APPLY;`

3. Verify redo transport services on the primary database

   You might have to reenable the primary database remote archive destination. Query the `V$ARCHIVE_DEST_STATUS` view first to see the current state of the archive destinations:

```
SELECT DEST_ID, DEST_NAME, STATUS, PROTECTION_MODE, DESTINATION, ERROR, SRL
      FROM V$ARCHIVE_DEST_STATUS;
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_n=ENABLE;
ALTER SYSTEM ARCHIVE LOG CURRENT;
```

   Verify redo transport services between the primary and standby databases by checking for errors. Query the `V$ARCHIVE_DEST` and `V$ARCHIVE_DEST_STATUS` views:

```
SELECT STATUS, TARGET, LOG_SEQUENCE, TYPE, PROCESS, REGISTER, ERROR
    FROM V$ARCHIVE_DEST;
SELECT * FROM V$ARCHIVE_DEST_STATUS WHERE STATUS!='INACTIVE';
```

4. Verify that recovery is progressing on standby database

   - For a physical standby database, verify that there are no errors from the managed recovery process and that the recovery has applied the redo from the archived redo log files:

```
SELECT MAX(SEQUENCE#), THREAD# FROM V$LOG_HISTORY GROUP BY THREAD;
SELECT PROCESS, STATUS, THREAD#, SEQUENCE#, CLIENT_PROCESS
    FROM V$MANAGED_STANDBY;
```

   - For a logical standby database, verify that there are no errors from the logical standby process and that the recovery has applied the redo from the archived redo logs:

```
SELECT THREAD#, SEQUENCE# SEQ#
```

```
FROM DBA_LOGSTDBY_LOG LOG, DBA_LOGSTDBY_PROGRESS PROG
WHERE PROG.APPLIED_SCN BETWEEN LOG.FIRST_CHANGE# AND LOG.NEXT_CHANGE#
ORDER BY NEXT_CHANGE#;
```

5. Restore primary database protection mode

   If you had to change the protection mode of the primary database from maximum protection to either maximum availability or maximum performance because of the standby database outage, then change the primary database protection mode back to maximum protection depending on your business requirements.

   ```
   ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE [PROTECTION | AVAILABILITY];
   ```

See Also: *Oracle Data Guard Concepts and Administration*

Restoring Fault Tolerance After a Standby Database Data Failure

Following unplanned downtime on the standby database that requires a full or partial data file restoration (such as data or media failure), full fault tolerance is compromised until the standby database is brought back into service. Full database protection should be restored as soon as possible.

To repair data corruption and data failures on a logical standby database, **you require a backup of the logical standby file and not a backup from the primary database**. Otherwise, you must reinstate or re-create the relevant objects affected by the corruption.

To repair data corruption or data failures on the standby database, you can use the following repair solutions:

- Use RMAN and Block Media Recovery (described in Section , "Use RMAN and Block Media Recovery")
- Use RMAN and Data File Media Recovery (described in Section , "Use RMAN and Data File Media Recovery")
- Re-Create Objects Manually for logical standby databases only (described in Section 14.2.6.7, "Re-Create Objects Manually")

If you had to change the protection mode of the primary database from maximum protection to either maximum availability or maximum performance because of the standby database outage, then change the primary database protection mode back to maximum protection (depending on your business requirements).

```
ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE [PROTECTION | AVAILABILITY];
```

Restoring Fault Tolerance After the Primary Database Was Opened Resetlogs

If the primary database is activated because it was flashed back to correct a logical error or because it was restored and recovered to a point in time, then the corresponding standby database might require additional maintenance. No additional work is required if the primary database completed recovery with no resetlogs.

After opening the primary database with the `RESETLOGS` option, execute the queries shown in Table 12–12.

Table 12–12 Queries to Determine RESETLOGS SCN and Current SCN OPEN RESETLOGS

Database	Query
Primary	`SELECT TO_CHAR(RESETLOGS_CHANGE# - 2) FROM V$DATABASE;`

Table 12–12 (Cont.) Queries to Determine RESETLOGS SCN and Current SCN OPEN RESETLOGS

Database	Query
Physical standby	`SELECT TO_CHAR(CURRENT_SCN) FROM V$DATABASE;`
Logical standby	`SELECT APPLIED_SCN FROM DBA_LOGSTDBY_PROGRESS;`

Table 12–13 shows the actions you take to restore fault tolerance if the standby database is behind the primary database's resetlogs SCN.

Table 12–13 SCN on Standby Database is Behind RESETLOGS SCN on the Primary Database

Database	Action
Physical standby	1. Ensure that the standby database has received an archived redo log file from the primary database. **See Also**: "Verify redo transport services on the primary database" on page 12-48 2. Restart Redo Apply.
Logical standby	Ensure that the standby database has received an archived redo log file from the primary database. **See Also**: "Verify redo transport services on the primary database" on page 12-48

Table 12–14 shows the actions you take to restore fault tolerance if the standby database is ahead of the primary database's resetlogs SCN.

Table 12–14 SCN on the Standby is Ahead of Resetlogs SCN on the Primary Database

Database	Action

Physical standby	1. Ensure that the standby database has received an archived redo log file from the primary database.
	See Also: "Verify redo transport services on the primary database" on page 12-48
	2. Issue the `SHUTDOWN IMMEDIATE` statement, if necessary.
	3. Issue the `STARTUP MOUNT` statement.
	4. Issue the `FLASHBACK DATABASE TO SCN` *`flashback_scn`* statement where *`flashback_scn`* is the SCN returned from the primary database query in Table 12–12. The SCN returned from the primary database query is 2 less than the `RESETLOGS_CHANGE#`.
	Issue the `FLASHBACK DATABASE TO SCN resetlogs_change#_minus_2` statement.
	5. Restart Redo Apply with or without real-time apply:
	With real-time apply:
	`ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT;`
	Without real-time apply:
	`ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;`

Table 12–14 (Cont.) SCN on the Standby is Ahead of Resetlogs SCN on the Primary

Database	Action

Logical standby	1. Determine the SCN at the primary database.
	On the primary database, use the following query to obtain the value of the system change number (SCN) that is 2 SCNs before the RESETLOGS operation occurred on the primary database:
	`SQL> SELECT TO_CHAR(RESETLOGS_CHANGE# - 2) AS FLASHBACK_SCN FROM V$DATABASE;`
	2. Determine the target SCN for flashback operation at the logical standby:
	`SQL> SELECT DBMS_LOGSTDBY.MAP_PRIMARY_SCN (PRIMARY_SCN => FLASHBACK_SCN)`
	`2> AS TARGET_SCN FROM DUAL;`
	3. Flash back the logical standby to the *TARGET_SCN* returned.
	Issue the following SQL statements to flash back the logical standby database to the specified SCN, and open the logical standby database with the RESETLOGS option:
	`SQL> SHUTDOWN;`
	`SQL> STARTUP MOUNT EXCLUSIVE;`
	`SQL> FLASHBACK DATABASE TO SCN TARGET_SCN;`
	`SQL> ALTER DATABASE OPEN RESETLOGS;`
	4. Start SQL Apply:
	`SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;`

Restoring Fault Tolerance After Dual Failures

If a dual failure affecting both the standby and primary databases occurs, then you must re-create the primary database first. Because the sites are identical, the primary database can be created wherever the most recent backup resides.

Table 12–15 summarizes the recovery strategy depending on the type of backups that are available.

Table 12–15 Re-Creating the Primary and Standby Databases

Available Backups	Re-Creating the Primary Database
Backups (including redo) managed by ZDBRA	Restore entire database and associated redo to achieve near zero data loss.
Local backup on primary and standby databases	Restore backup from the primary database. Recover and activate the database as the new primary database.
Local backup only on standby database. Tape backups on standby database.	Restore the local standby backup to the standby database. Recover and activate the database as the new primary database.
Tape backups only	Restore tape backups locally. Recover the database and activate it as the new primary database.

See Also: *Oracle Data Guard Concepts and Administration* for the steps for creating a standby database after the primary database is re-created

13

Reducing Downtime for Planned Maintenance

This chapter describes scheduled outages and the Oracle operational best practices that can tolerate or manage each outage type and minimize downtime.

This chapter contains the following topics:

- Overview of Scheduled Outages
- Eliminating or Reducing Downtime for Scheduled Outages

> **See Also:** Chapter 12, "Recovering from Unscheduled Outages" for information about unscheduled outages

Overview of Scheduled Outages

Scheduled outages are required for regular maintenance of the technology infrastructure that supports the application, including tasks such as:

- Hardware maintenance, repair, and upgrades
- Software upgrades and patching
- Application (programmatic) changes, patches, and upgrades
- Changes to improve performance and manageability of systems

You can implement many of these tasks while maintaining continuous application availability.

The following sections provide best practice recommendations for reducing scheduled outages on the primary and secondary sites:

- Managing Scheduled Outages on the Primary Site
- Managing Scheduled Outages On the Secondary Site

Managing Scheduled Outages on the Primary Site

Table 13–1 shows the preferred solutions for performing scheduled outages on the primary site. The table includes links to detailed descriptions in Section , "Eliminating or Reducing Downtime for Scheduled Outages".

Table 13–1 Solutions for Scheduled Outages on the Primary Site

Planned Maintenance	Description and Examples	Preferred Oracle Solution	Estimated Downtime
Site maintenance	Maintenance performed on the entire site where the current primary database resides is unavailable. Usually known well in advance. ■ Scheduled power outages ■ Site maintenance ■ Regular planned switchovers to test infrastructure	Section , "Site, Hardware, and Software Maintenance Using Data Guard Switchover"	< 5 minutes
Hardware maintenance or system software maintenance that impacts the entire database cluster	Hardware maintenance on a database server cluster. ■ Upgrade of the cluster interconnect ■ Upgrade to the storage tier that requires downtime on the database tier	Section , "Site, Hardware, and Software Maintenance Using Data Guard Switchover"	< 5 minutes
Hardware maintenance or system software maintenance that impacts a subset of the database cluster	Hardware maintenance or system software maintenance on a database server. The scope of the downtime is restricted to a node of the database cluster. ■ Proactive replacement of RAID card battery ■ Addition of memory or CPU to an existing node in the database tier ■ Upgrade of a software component such as the operating system ■ Changes to the configuration parameters for the operating system	Oracle RAC service relocation (see Section , "Dynamic Database Services for System Maintenance")	No downtime

Table 13–1 (Cont.) Solutions for Scheduled Outages on the Primary Site

Planned Maintenance	Description and Examples	Preferred Oracle Solution	Estimated Downtime
Perform patch set, maintenance, or major upgrade to Oracle Grid Infrastructure (includes Oracle Clusterware and Oracle ASM)	Software maintenance of Grid Infrastructure. ▪ Patch set upgrade Grid Infrastructure from 12*c* Release 1 (12.1.0.1) to 12*c* Release 1 (12.1.0.2) Patch Set 1 ▪ Maintenance release upgrade from 11*g* Release 1 to 11*g* Release 2 ▪ Major release upgrade from 11*g* to 12*c*	▪ Section , "Grid Infrastructure Upgrade" *or* ▪ Section , "Site, Hardware, and Software Maintenance Using Data Guard Switchover" *or* ▪ Section , "Data Guard Standby-First Patch Apply" See *Oracle Database 2 Day + Real Application Clusters Guide* and see your platform-specific Oracle Grid Infrastructure Installation Guide for complete details, in the appendix, "How to Upgrade to Oracle Grid Infrastructure"	No downtime (when no role transition) < 5 minutes (if role transition occurs)
Perform patch set, maintenance, or major upgrade to Oracle Database	Software maintenance of Oracle Database. ▪ Patch set upgrade Grid from 11g Release 2 (11.2.0.1) to 11g Release 2 (11.2.0.2) Patch Set 1 ▪ Maintenance release upgrade from 11g Release 1 to 11g Release 2 ▪ Major release upgrade from 10g to 11g	▪ Oracle Database rolling upgrade with Data Guard SQL Apply (see Section , "Upgrading with Data Guard SQL Apply or Transient Logical Standby Database") *or* ▪ Oracle GoldenGate (see Section , "Upgrading with Oracle GoldenGate")	< 5 minutes
Apply Patch Set Update (PSU), Critical Patch Update (CPU), or patch bundle	Software maintenance of Grid Infrastructure or Oracle Database. ▪ Installation of Patch Set Update 11.2.0.2.3 ▪ Installation of 11.2.0.2 Grid Infrastructure Bundle 1 ▪ Installation of Critical Patch Update July 2011 ▪ Installation of Exadata Database Bundle Patch 8	▪ Oracle RAC rolling patch upgrade using OPatch (see Section , "Oracle Database and Grid Infrastructure Patching") *or* ▪ Section , "Data Guard Standby-First Patch Apply"	No downtime No downtime (when no role transition or < 5 minutes if role transition occurs

Table 13–1 (Cont.) Solutions for Scheduled Outages on the Primary Site

Planned Maintenance	Description and Examples	Preferred Oracle Solution	Estimated Downtime
Apply Oracle interim ("one-off") or diagnostic patch	Patch Oracle software to fix a specific customer issue. ■ Installation of patch 10205230	■ Oracle RAC rolling patch upgrade using OPatch (see Section , "Oracle Database and Grid Infrastructure Patching") *or* ■ Section , "Data Guard Standby-First Patch Apply" *or* ■ Section , "Online Patching"	No downtime No downtime (when no role transition or < 5 minutes if role transition occurs No downtime
Database object reorganization or redefinition	Changes to the logical structure or the physical organization of Oracle Database objects, primarily to improve performance or manageability. Changes to the data or schema. Using the Oracle Database online redefinition feature enables objects to be available during the reorganization or redefinition. ■ Moving an object to a different tablespace ■ Converting a table to a partitioned table ■ Add, modify, or drop one or more columns in a table or cluster	Online object reorganization with `DBMS_REDEFINITION` (see Section , "Data Reorganization and Redefinition")	No downtime

Table 13–1 (Cont.) Solutions for Scheduled Outages on the Primary Site

Planned Maintenance	Description and Examples	Preferred Oracle Solution	Estimated Downtime
Database storage maintenance	Maintenance of storage where database files reside. ▪ Converting to Oracle ASM ▪ Adding or removing storage ▪ Patching or upgrading storage firmware or software	Online storage maintenance using Oracle ASM (see Section , "Storage Maintenance")	No downtime
Database platform or location migration	Changing operating system platform of the primary and standby databases. Changing physical location of the primary database ▪ Moving to the Linux operating system ▪ Moving the primary database from one data center to another	Section , "Database Platform or Location Migration"	< 5 minutes to hours (depending on method chosen)
Application changes	May include data changes, schema, and other programmatic changes. ▪ Application upgrades	▪ Section , "Edition-Based Redefinition for Online Application Maintenance and Upgrades" *or* ▪ Section , "Upgrading with Oracle GoldenGate"	< 5 minutes

Managing Scheduled Outages On the Secondary Site

Scheduled outages on the secondary site may impact availability of applications that use Active Data Guard to offload read-intensive work from the primary database. Outages on the secondary site might affect the RTO and RPO if there are concurrent failures on the primary site. Outages on the secondary site can be managed with no effect on primary database availability:

- If maximum protection database mode is configured and there is only one standby database protecting the primary database, then you must downgrade the protection mode before scheduled outages on the standby instance or database so that there is no downtime on the primary database.
- If maximum protection database mode is configured and there are multiple standby databases, there is no need to downgrade the protection mode if at least one standby database that is configured with the `LGWR SYNC AFFIRM` attributes is available, and to which the primary database can transmit redo data.

When scheduling secondary site maintenance, consider that the duration of a site-wide or clusterwide outage adds to the time that the standby database lags behind the primary database, which in turn lengthens the time to restore fault tolerance. See Section , "Determine Protection Mode and Data Guard Transport"for an overview of the Data Guard protection modes.

Table 13–2 describes the steps for performing scheduled outages on the secondary site.

Table 13–2 Managing Scheduled Outages on the Secondary Site

Planned Maintenance	Oracle Database 11*g* with Data Guard	Oracle Database 11*g* - MAA
Site shutdown	Before the outage: Section , "Managing Scheduled Outages On the Secondary Site" After the outage: Section , "Restoring Fault Tolerance After Planned Downtime on Secondary Site or Cluster"	Before the outage: Section , "Managing Scheduled Outages On the Secondary Site" After the outage: Section , "Restoring Fault Tolerance After Planned Downtime on Secondary Site or Cluster"
Hardware or non-Oracle database software maintenance on the node that is running the managed recovery process (MRP)	Before the outage: Section , "Managing Scheduled Outages On the Secondary Site"	Before the outage: Section , "Managing Scheduled Outages On the Secondary Site"
Hardware or non-Oracle database software maintenance on a node that is not running the MRP	Not applicable	No effect because the primary standby node or instance receives redo logs that are applied with the managed recovery process After the outage: Restart node and instance, when available
Hardware or non-Oracle database software maintenance (clusterwide impact)	Not applicable	Before the outage: Section , "Managing Scheduled Outages On the Secondary Site" After the outage: Section , "Restoring Fault Tolerance After Planned Downtime on Secondary Site or Cluster"
Oracle patch and software upgrades	Downtime needed for upgrade, but there is no effect on the primary node unless the configuration is in maximum protection database mode	Downtime needed for upgrade, but there is no effect on the primary node unless the configuration is in maximum protection database mode

Eliminating or Reducing Downtime for Scheduled Outages

The best practices for eliminating or reducing downtime for scheduled outages includes the following sections:

- Site, Hardware, and Software Maintenance Using Data Guard Switchover
- Online Patching
- Data Guard Standby-First Patch Apply
- Oracle Database and Grid Infrastructure Patching
- Storage Maintenance
- Database Upgrades
- Database Platform or Location Migration
- Edition-Based Redefinition for Online Application Maintenance and Upgrades
- Oracle GoldenGate for Online Application Upgrades
- Data Reorganization and Redefinition
- Dynamic Database Services for System Maintenance

Before performing any update to your system, Oracle recommends you perform extensive testing.

Site, Hardware, and Software Maintenance Using Data Guard Switchover

A *switchover* is a planned transition that includes a series of steps to switch database roles between the primary and standby databases. Following a successful switchover operation, the standby database assumes the primary role and the primary database becomes a standby database. Database switchover can be done by Oracle Enterprise Manager, Oracle Data Guard broker, or by issuing SQL*Plus statements. At times the term *switchback* is also used within the scope of database role management. A switchback operation is a subsequent switchover operation to return the standby databases to their original roles.

Switchovers are useful in many situations when performing site maintenance, and hardware or software maintenance such as a grid infrastructure upgrade.

When to Perform a Data Guard Switchover

Switchover can occur whenever a primary database is started, the target standby database is available, and all the archived redo logs are available.

Switchovers are useful in the following situations:

- Scheduled maintenance such as hardware maintenance or firmware patches on the primary host
- Resolution of data failures when the primary database is still opened
- Testing and validating the secondary resources, as a means to test disaster recovery readiness
- When using SQL Apply to perform a rolling upgrade (see Section , "Upgrading with Data Guard SQL Apply or Transient Logical Standby Database")

Switchover is not possible or practical under the following circumstances:

- Archived redo log files that are needed for apply are missing
- A point-in-time recovery is required
- The primary database is not open and cannot be opened

Best Practices for Configuring Data Guard Switchover

Before performing a switchover, employ the Data Guard configuration best practices. For more information, see Section , "Oracle Data Guard Switchovers Best Practices".

How to Perform Data Guard Switchover

You can perform switchovers using Oracle Enterprise Manager. If you are not using Oracle Enterprise Manager, then you can perform switchovers manually using the DGMGRL command-line interface or SQL*Plus statements:

- Using Oracle Enterprise Manager, as described in *Oracle Data Guard Broker*
- Using the DGMGRL command-line interface, as described in *Oracle Data Guard Broker*
- Using SQL*Plus:
  - Role Transitions Involving Physical Standby Databases:

See *Oracle Data Guard Concepts and Administration* for detailed steps on Role Transitions Involving Physical Standby Databases.

– Role Transitions Involving Logical Standby Databases:

See *Oracle Data Guard Concepts and Administration* for detailed step on Role Transitions Involving Logical Standby Databases.

After performing the Data Guard Switchover do the following:

- If the database is moved to the secondary site and the application tier is also moved to the secondary site, perform complete site failover. For more information see Section , "Complete Site Failover (Failover to Secondary Site)."
- If only the database is moved to the secondary site, perform application failover. See Section , "Application Failover" for more information.

Online Patching

Beginning with Oracle Database 11*g* there is support for online patching for qualified interim and diagnostic patches. Online patching provides the ability to patch the processes in an Oracle instance without bringing the instance down. Each process associated with the instance checks for patched code at a safe execution point, and then copies the code into its process space. Thus, the processes being patched may not necessarily pick up the new code at the exact same time.

A key difference between traditional patching and online patching is that traditional patching is implemented at the software level and online patching is implemented at the software or Oracle Database instance level. In other words, instances using an `ORACLE_HOME` that receives a traditional patch always use the patched code whereas instances using an `ORACLE_HOME` that receives an online patch receive the patched code only if the instance is specified when the patch is applied.

Note: For online patching, note the following:

- See the patch README for details on whether a patch supports online installable.

The best practices for online patching:

- During the next scheduled maintenance, when instances can be shutdown, rollback all online patches and apply the patches in an offline manner.
- Patches that are online installable should be installed in an online manner when the patch needs to be applied urgently and downtime cannot be taken to apply the patch. If instance downtime is acceptable, then apply the patch in an offline manner (as described in the patch README).
- Apply the patch to one instance at a time.
- When rolling back online patches, ensure all patched instances are included to avoid the dangerous and confusing situation of having different software across instances using the same `$ORACLE_HOME`.
- Assess memory impact on a test system before deploying to production (for example: using the `pmap` command).
- Never remove the `$ORACLE_HOME/hpatch` directory.

See Also:

- *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX* for information about Patching Oracle Software with OPatch
- For the most up-to-date information about online patching, installation and rollback, see "RDBMS Online Patching - Hot Patching" in My Oracle Support Note 761111.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=761111.1

Data Guard Standby-First Patch Apply

Oracle Data Guard Standby-First Patch Apply provides support for different database home software between a primary database and its physical standby database(s) for the purpose of applying and validating Oracle patches in rolling fashion with minimal risk to the primary database.

Data Guard has long supported running different configuration between primary and standby systems. This has included support for the following:

- Differences in hardware (for example, X3 Exadata Database Machine with X4 Exadata Database Machine)
- Differences in operating system (for example, Oracle Linux 5.7 with Oracle Linux 5.8)
- Differences in database storage (for example, Oracle ASM-based storage with NFS-based storage, or Exadata 11.2 with Exadata 12.1)
- Differences in Oracle Clusterware version and patch level (for example, 11.2.0.3 GIPSU4 with 11.2.0.3 GIPSU5)

However, differences in database home software were limited to rolling upgrade scenarios supported only by logical standby databases. In order to apply a later database home patch (for example, Exadata bundle patch, or database PSU) to a Data Guard environment with physical standby, you had to perform one of the following actions:

- Shutdown both the primary and standby databases and apply the update to both systems before restarting, or
- Convert the physical standby database to a logical standby database and apply the update using the rolling upgrade process, then convert the standby database back to a physical standby (a feature known as transient logical standby).

With Data Guard Standby-First Patch Apply, Oracle supports different database home software between a primary database and its physical standby database(s), in addition to the differences listed above.

Oracle Data Guard Standby-First Patch Apply provides support for different database home software between a primary database and its physical standby database(s) for the purpose of applying and validating Oracle patches and patch bundles in rolling fashion with minimal risk to the primary database. For example, with Data Guard Standby-First Patch Apply you apply a database home patch first to a physical standby database. The standby is used to run read-only workload, or read-write workload if it is a snapshot standby, for testing and evaluation of the patch. After passing evaluation, the patch is then installed on the primary system with greater assurance of the effectiveness and stability of the database home patch.

Oracle Data Guard Standby-First Patch Apply is supported only for certified interim patches and patch bundles (for example, Patch Set Update, or Database Patch for Exadata) for Oracle Database 11.2.0.1 and later, on both Oracle Engineered Systems (for example, Exadata, SuperCluster) and non-Engineered Systems. A patch and patch bundle that is Data Guard Standby-First certified will state the following in the patch README:

Data Guard Standby-First Installable

The following types of patches are candidates to be Data Guard Standby-First certified:

- Database home interim patches
- Exadata bundle patches (for example, monthly and quarterly database patches for Exadata)
- Database patch set updates

> **Note:** Patches and patch bundles that update modules that may potentially disrupt the interoperability between primary and physical standby systems running different database home software will not be certified "Data Guard Standby-First Installable" and will not state so in the patch README.

Oracle patch sets and major release upgrades do not qualify for Data Guard Standby-First Patch Apply. For example, upgrades from 11.2.0.2 to 11.2.0.3 or 11.2 to 12.1 do not qualify. Use the Data Guard transient logical standby rolling upgrade process for database patch sets and major releases.

Other configuration differences between primary and standby systems listed above in the Overview section that have been previously supported continue to be supported.

Data Guard Standby-First Patch Apply has the following advantages:

- Ability to apply software changes to the physical standby database for recovery, backup or query validation prior to role transition, or prior to application on the primary database. This mitigates risk and potential downtime on the primary database.
- Ability to switch over to the targeted database after completing validation with reduced risk and minimum downtime.
- Ability to switch back, also known as fallback, if there are stability or performance regressions.

See Also:

- "Database Rolling Upgrades Made Easy by Using a Data Guard Physical Standby Database" MAA white paper at `http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-upgrades-made-easy-131972.pdf`
- *Oracle Database Upgrade Guide* for information about Considerations for Downgrading and Compatibility and the Oracle Database `COMPATIBLE` parameter
- *Oracle Automatic Storage Management Administrator's Guide* for information about Disk Group Compatibility Attributes
- "Oracle Patch Assurance - Data Guard Standby-First Patch Apply" in My Oracle Support Note 1265700.1 at

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1265700.1`

Oracle Database and Grid Infrastructure Patching

See the following sections:

- Rolling Patch Installation
- Best Practices to Minimize Downtime for All Patch Upgrades
- Best Practices to Minimize Downtime for Database Rolling Patches
- Out-of-place Software Installation and Patching
- Using OPlan for Patching

Rolling Patch Installation

With Oracle RAC, you can apply certain database and grid infrastructure patches to one node or instance at a time, which enables continual application and database availability. Interim ("one-off") patches, Patch Set Updates (PSUs), and Critical Patch Updates (CPUs) to database software are usually applied to implement known fixes for software problems an installation has encountered or to apply diagnostic patches to gather information regarding a problem. Such patch application is often carried out during a scheduled maintenance outage.

Oracle now provides the capability to do rolling patch upgrades with Oracle RAC with little or no database downtime. The tool used to achieve this is the `opatch` command-line utility.

The advantage of an Oracle RAC rolling patch installation is that it enables at least some instances of the Oracle RAC installation to be available during the scheduled outage required for patch upgrades. Only the Oracle RAC instance that is currently being patched must be brought down. The other instances can continue to remain available. Thus, the effect on the application downtime required for such scheduled outages is further minimized. Oracle's `opatch` utility enables the user to apply the patch successively to the different instances of the Oracle RAC installation.

Rolling patch installation is available only for patches that have been certified by Oracle to be eligible for rolling upgrades. The patch README file indicates whether a patch can be applied in an Oracle RAC rolling manner. Typically, patches that can be installed in a rolling manner include:

- Exadata Database Bundle Patches

- Patch Set Update (PSU)
- Critical Patch Update (CPU)
- Interim (“one-off”) patches
- Diagnostic patches

Rolling patch installation is not available for deployments where the Oracle Database software is shared across the different nodes. This is the case where the Oracle home is on Cluster File System (CFS) or on shared volumes provided by file servers or NFS-mounted drives. The feature is only available where each node has its own copy of the Oracle Database software.

> **See also:** Section , "Grid Infrastructure Upgrade"and Section , "Database Upgrades"sections for patch set upgrades and release upgrades.

Best Practices to Minimize Downtime for All Patch Upgrades

Use the following recommended practices for all database patch installations:

- Always confirm with Oracle Support Services that the patch is valid for your problem and for your deployment environment.
- Have a plan for applying the patch and a plan for backing out the patch.
- Apply the patch to your test environment first and verify that it fixes the problem.
- When you plan the elapsed time for applying the patch, include time for starting up and shutting down the other tiers of your technology stack if necessary.
- If the patch is not a candidate for Oracle RAC rolling patch installation and you can incur the downtime for applying the patch, go to Section , "Database Upgrades" on page 13-17 to assess whether other solutions are feasible.

Best Practices to Minimize Downtime for Database Rolling Patches

The following are additional recommended practices for Oracle RAC rolling patch installation.

- If multiple instances share an Oracle home, then all of them are affected by application of a patch. Administrators should verify that this does not cause unintentional side effects. Also, you must shut down all such instances on a node during the patch application. You must take this into account when scheduling a planned outage. As a best practice, only similar applications should share an Oracle home on a node. This provides greater flexibility for patching.
- The Oracle inventory on each node is the repository that keeps a central inventory of all Oracle software installed. The inventory is node-specific. It is shared by all Oracle software installed on the node. It is similar across nodes only if all nodes are the same in terms of the Oracle Database software deployed, the deployment configuration, and patch levels. Because the Oracle inventory greatly aids the patch application and patch management process, it is recommended that its integrity be maintained. Oracle inventory should be backed up after each patch installation to any Oracle software on a specific node. This applies to the Oracle inventory on each node of the cluster.
- Use the Oracle Universal Installer to install all Oracle database software. This creates the relevant repository entries in the Oracle inventory on each node of the cluster. Also, use the Oracle Universal Installer to add nodes to an existing Oracle RAC cluster.

However, if this was not done or is not feasible for some reason, adding information about an existing Oracle database software installation to the Oracle inventory can be done with the `attach` option of the `opatch` utility. Node information can be also added with this option.

- The nature of the Oracle rolling patch upgrade enables it to be applied to only some nodes of the Oracle RAC cluster. So an instance can be operating with the patch applied, while another instance is operating without the patch. This is not possible for nonrolling patch upgrades. Apply nonrolling patch upgrades to all instances before the Oracle RAC deployment is activated. A mixed environment is useful if a patch must be tested before deploying it to all the instances. Applying the patch with the `-local` option is the recommended way to do this.

  In the interest of keeping all instances of the Oracle RAC cluster at the same patch level, it is strongly recommended that after a patch has been validated, it should be applied to all nodes of the Oracle RAC installation. When instances of an Oracle RAC cluster have similar patch software, services can be migrated among instances without running into the problem a patch might have fixed.
- Maintain all patches (including those applied by rolling upgrades) online and do not remove them after they have been applied. Keeping the patches is useful if a patch must be rolled back or applied again.

  Store the patches in a location that is accessible by all nodes of the cluster. Thus all nodes of the cluster are equivalent in their capability to apply or roll back a patch.
- Perform rolling patch upgrades, just like any other patch upgrade, when no other patch upgrade or Oracle installation is being performed on the node. The application of multiple patches is a sequential process, so plan the scheduled outage accordingly.
- If you must apply multiple patches at the same time but only some patches are eligible for rolling upgrade, then apply all of the patches in a nonrolling manner. This reduces the overall time required to accomplish the patching process.
- For patches that are not eligible for rolling upgrade, the next best option for Oracle RAC deployments is the `MINIMIZE_DOWNTIME` option of the `APPLY` command.
- Perform the rolling upgrade when system usage is low to ensure minimal disruption of service for the end users.

> **See Also:** *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX* for more information about the opatch utility

Out-of-place Software Installation and Patching

The following software installations are performed, by default, out-of-place by Oracle Universal Installer (OUI). Software installations performed by OUI are full software installations:

- Major release
- Maintenance release
- Patch set (beginning with 11g Release 2)

The following software installations are performed in-place by the OPatch utility. OPatch installs the software update into an existing `ORACLE_HOME` by overwriting existing software with updated software from the patch being installed:

- Interim patch installation
- Bundle patch installation

- Patch Set Update (PSU) installation
- Critical Patch Update (CPU) installation
- Diagnostic patch installation

Advantages of out-of-place patching

- Applications remain available while software is upgraded in the new ORACLE_HOME.
- The configuration inside the ORACLE_HOME is retained because the cloning procedure involves physically copying the software (examples are files such as LISTENER.ORA, TNSNAMES.ORA, and INITSID.ORA).
- It is easier to rollback or test between the original ORACLE_HOME and the patched ORACLE_HOME.
- When consolidating, you could have multiple versions of ORACLE_HOME, so this option should better support consolidation.

Considerations for using out-of-place patching

- When performing out-of-place patch installation with cloning, you must change any ORACLE_HOME environment variable hard coded in application code and Oracle-specific scripts.
- Out-of-place patching requires more disk space than in-place patching.

Out-of-place patching with OPatch

Traditionally, patches installed with OPatch are done in-place, which means that the new code is applied directly over the old code.

The disadvantages of in-place patching are:

- The application cannot connect to the database while new code is being installed.
- If patch rollback is required, the application cannot connect to the database while old code is being reinstalled.

Note: This downside to an in-place database patch set upgrade does not apply when you use Standby-First Patch apply.

For more information, see Section , "Data Guard Standby-First Patch Apply."

Software installation performed by OPatch to the Oracle Database software home or the Grid Infrastructure software home can be performed out-of-place by using ORACLE_HOME cloning techniques to copy the software to a new home directory before applying a patch to the new ORACLE_HOME with OPatch. The high-level approach to perform out-of-place patching is:

1. Clone the active ORACLE_HOME to a new ORACLE_HOME.
2. Patch the new ORACLE_HOME.
3. Switch to make the new ORACLE_HOME the active software home. This can be done in a rolling manner one node at a time.

See Also: For details about out-of-place patching, see "Minimal downtime patching via cloning 11gR2 ORACLE_HOME directories on Oracle Database Machine" My Oracle Support Note 1136544.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1136544.1

Using OPlan for Patching

OPlan is a utility that facilitates the patch installation process by providing you with step-by-step patching instructions specific to your environment for both in-place and out-of-place patch installation. Currently, OPlan is supported for Exadata Database Bundle Patches. For the latest information, see "Oracle Software Patching with OPLAN" in My Oracle Support Note 1306814.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1306814.1

Grid Infrastructure Upgrade

Beginning with Oracle Database 11*g* Release 2 (11.2), Oracle ASM is installed when you install the Oracle Grid Infrastructure components, and it shares the Oracle home with Oracle Clusterware when installed in a cluster such as with Oracle RAC. When performing maintenance on the Grid Infrastructure, such as patching or upgrade, it will impact both Oracle ASM and Oracle Clusterware.

Grid Infrastructure Rolling Upgrade

Grid Infrastructure upgrade means taking the Oracle Clusterware and Oracle ASM to a later major version, maintenance version, or patch set. Grid Infrastructure upgrade is performed in a rolling manner.

See Also: *Oracle Database 2 Day + Real Application Clusters Guide*

Oracle Grid Infrastructure Installation Guide for your platform for complete details, in the Appendix, "How to Upgrade to Oracle Grid Infrastructure"

Storage Maintenance

Use the following procedure when adding or upgrading storage on the system. The procedures in the following sections assume that you are adding storage to an Oracle ASM disk group.

- Migrating to Oracle ASM Storage
- Adding and Removing Storage

Migrating to Oracle ASM Storage

If you have an existing Oracle database that stores database files on non-ASM storage, you can migrate some or all of these database files to Oracle ASM. Migrating a database to ASM storage requires moving data files, redo log files, and control files to ASM.

Data Files

You can use the `ALTER DATABASE MOVE DATAFILE` SQL statement to relocate online data files. This statement enables you to relocate a data file to ASM storage while the database is open and users are accessing the data file.

Redo Log Files

Use the `ALTER DATABASE DROP LOGFILE GROUP` SQL statement to drop inactive redo log files from non-ASM storage, and the `ALTER DATABASE ADD LOGFILE` SQL statement to add redo log files to ASM storage.

Control Files

Migrate control files to ASM storage by shutting down the database, copying the existing control files into ASM using Recovery Manager (RMAN) or ASMCMD, editing database initialization parameter `CONTROL_FILES` to refer to the new location, then starting the database.

See Also: *Oracle Database Administrator's Guide*

Adding and Removing Storage

Disks can be added to and removed from Oracle ASM with no downtime. When disks are added or removed, Oracle ASM automatically starts a rebalance operation to evenly spread the disk group contents over all drives in the disk group.

The best practices for adding or removing storage include:

- Make sure your host operating system and storage hardware can support adding and removing storage with no downtime before using Oracle ASM to do so.
- Use a single `ALTER DISKGROUP` command when adding or removing multiple disk drives (this way there is only one rebalance operation where, with separate drops and adds there are two or more rebalance operations. For more information, see Section , "Use a Single Command to Add or Remove Storage").

  For example, if the storage maintenance is to add drives and remove existing drives, use a single `ALTER DISKGROUP` command with the `DROP DISK` clause to remove the existing drives and the `ADD DISK` clause to add the drives:

  ```
  ALTER DISKGROUP data
        DROP DISK diska5
        ADD FAILGROUP failgrp1 DISK '/devices/diska9' NAME diska9;
  ```

- When dropping disks from a disk group, specify the `WAIT` option in the `REBALANCE` clause so the `ALTER DISKGROUP` statement does not return until the contents of the drives being dropped have been moved to other drives. After the statement completes, the drives can be safely removed from the system. For example:

  ```
  ALTER DISKGROUP data
  DROP DISK diska5
  ADD FAILGROUP failgrp1 DISK '/devices/diska9' NAME diska9
  REBALANCE WAIT;
  ```

- When dropping disks in a normal or high redundancy disk group, ensure there is enough free disk space in the disk group to reconstruct full redundancy.
- Monitor the progress of rebalance operations using Enterprise Manager or by querying `V$ASM_OPERATION`.
- For long-running rebalance operations that occur during periods of low database activity, increase the rebalance power limit to reduce the rebalance time.

See Also: *Oracle Automatic Storage Management Administrator's Guide*

Database Upgrades

Database upgrade means taking the database to a later major release, maintenance release, or patch set. The following Oracle features are available to perform database upgrades:

- Upgrading with Database Upgrade Assistant (DBUA)
- Upgrading with Data Guard SQL Apply or Transient Logical Standby Database
- Upgrading with Oracle GoldenGate
- Upgrading by Transporting Data

The method you choose to perform database upgrades can vary depending on the following considerations:

- Downtime required to complete the upgrade
- Setup time and effort required before the downtime
- Temporary additional resources necessary (for example, disk space or CPU)
- Complexity of the steps allowed to complete the upgrade

Table 13–3 lists the methods that you can use for database upgrades, and recommends what method to use for particular cases.

Table 13–3 Database Upgrade Options

Upgrade Method	Use This Method When...
Upgrading with Database Upgrade Assistant (DBUA)	Recommended method when the maintenance window is sufficient or when data type constraints prohibit the use of the other methods in this table.
Upgrading with Data Guard SQL Apply or Transient Logical Standby Database	DBUA cannot finish within the maintenance window and the database is not a candidate for Oracle RAC rolling patch upgrade. Use a transient logical standby when the configuration has only a physical standby database.
Upgrading with Oracle GoldenGate	Oracle GoldenGate is already used for complete database replication or when the database version predates Oracle 10*g* (the minimum version for Oracle Data Guard database rolling upgrades), or when additional flexibility for replicating back to the previous version is required (fast fall back option) or where zero downtime upgrades using multi-master replication is required.
Upgrading by Transporting Data	The database is using data types unsupported by Data Guard SQL Apply or Oracle GoldenGate, and the user schemas are simple.

Regardless of the upgrade method you use, you should follow the guidelines and recommendations provided in the *Oracle Database Upgrade Guide* and its companion document, "Oracle 11gR2 Upgrade Companion" in My Oracle Support Note 785351.1 at

`https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=785351.1`

See Also:

- "Oracle Support Lifecycle Advisors" in My Oracle Support Note 250.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=250.1

- Oracle Technology Network Oracle Database Upgrade web page at

  http://www.oracle.com/technetwork/database/upgrade/index.html

Upgrading with Database Upgrade Assistant (DBUA)

Database Upgrade Assistant (DBUA) is used to upgrade a database in place from an earlier software version.

When deciding if DBUA is the proper tool to use when performing a database upgrade with minimal downtime, consider the following:

- DBUA upgrades the database dictionary and all components. For example: Java, XDB, and so on, that have been installed while the database is unavailable for normal user activity.
- Downtime required for a database upgrade when using DBUA is determined by the time needed to:
  - Upgrade all database dictionary objects to the new version. Upgrade is now performed in parallel using the Parallel Upgrade Utility (catctl.pl).
  - Recompile all invalid PL/SQL modules after the upgrade is complete. Recompilation can be performed in parallel to reduce upgrade time.
  - Restart the database
  - Reconnect the clients to the upgraded database
- To reduce the amount of downtime required for a database upgrade when using DBUA:
  - Use Upgrade Parallelism and parallelism to recompile invalid PL/SQL modules.
  - Remove any database options that are not being used.

    DBUA upgrades all of the installed database options, whether they are required by an application. By reducing the number of options that must be upgraded, you can reduce the overall upgrade time.
  - Update data dictionary statistics immediately before the upgrade.

Use DBUA for a database upgrade when the time to perform the upgrade with this method fits within the maintenance window.

See Also:

- *Oracle Database Upgrade Guide* for more information about DBUA and upgrading your Oracle Database software
- "Oracle 11gR2 Upgrade Companion" in My Oracle Support Note 785351.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=785351.1
- To upgrade from Oracle9*i* to Oracle Database 11*g*, see "Oracle 11gR1 Upgrade Companion" in My Oracle Support Note 601807.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=601807.1

Upgrading with Data Guard SQL Apply or Transient Logical Standby Database

Use Data Guard SQL Apply or a transient logical standby database to upgrade a database with minimal downtime using a process called a *rolling upgrade*. Data Guard currently supports homogeneous environments where the primary and standby databases run on the same platform.

See Also: For exceptions that are specific to heterogeneous environments and for other late-breaking information about rolling upgrades with SQL Apply, see "Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration" in My Oracle Support Note 413484.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=413484.1

SQL Apply Rolling Upgrades Use Data Guard SQL Apply for rolling database upgrade when a conventional upgrade cannot complete the upgrade within the maintenance window and the application does not use user-defined types. Oracle Data Guard using SQL Apply is the recommended solution for performing patch set and database upgrades with minimal downtime.

Note the following points when deciding if Data Guard SQL Apply is the appropriate method for minimizing downtime during a database upgrade:

- SQL Apply has some data type restrictions (see *Oracle Data Guard Concepts and Administration* for a list of the restrictions). If there are data type restrictions, consider implementing Extended Datatype Support (EDS). If the source database is using data types not natively supported by SQL Apply, you can use Extended Datatype Support (EDS) to accommodate several more advanced data types.
- You can perform a SQL Apply rolling upgrade for any upgrade, including a major release upgrade if the source release is Oracle Database 10*g* release 1 (10.1.0.3) or higher. Before you begin, review the detailed steps for a SQL Apply rolling upgrade and verify the supported data types in *Oracle Data Guard Concepts and Administration*.
- If the source database is using a software version not supported by SQL Apply rolling upgrade (earlier than Oracle Database release 10.1.0.3) or using EDS cannot sufficiently resolve SQL Apply data type conflicts, then consider using Database Upgrade Assistant (DBUA), transportable tablespace, or Oracle GoldenGate.

- Downtime required for a database upgrade (rolling upgrade) when using Data Guard SQL Apply is determined by the time needed to:
  - Perform a Data Guard switchover
  - Reconnect the clients to the new database

Note that for databases originating with the first patch set of Oracle Database 12*c* Release 1 (12.1), the preferred method for performing a rolling upgrade with an existing physical standby database is to use the DBMS_ROLLING PL/SQL package. Refer to *Oracle Data Guard Concepts and Administration*.

See Also:

- *Oracle Data Guard Concepts and Administration* for information about using SQL Apply to upgrade the Oracle Database
- The MAA white paper "Database Rolling Upgrade Using Data Guard SQL Apply Oracle Database 11g and 10gR2" at http://www.oracle.com/goto/maa
- For SQL Apply, EDS support starts from Oracle Database Release 10.2.0.4. The ESD implementation is different from 10.2.0.4 to 11.1 and in Oracle Database Release 11.2, for more information:

  From 10.2.0.4 to 11.1 you must build ESD following the examples specified in detail in "Extended Datatype Support (EDS) for SQL Apply" in My Oracle Support Note 559353.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=559353.1

  In Oracle Database Release 11.2 EDS-related procedures are part of the `DBMS_LOGSTDBY` package; for more information see "SQL Apply Extended Datatype Support - 11.2" in My Oracle Support Note 949516.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=949516.1

Transient Logical Standby Database Rolling Upgrade You can use a **transient logical standby database** to perform a rolling database upgrade using your current physical standby database by temporarily converting it to a logical standby database. Use a transient logical standby when your configuration only has a physical standby database. Performing a rolling upgrade using a transient logical standby is similar to the standard SQL Apply rolling upgrade with the following differences:

- A guaranteed restore point is created on the primary database to flash the database back to a physical standby database after the switchover.
- The conversion of a physical standby database to a logical standby database uses the `KEEP IDENTITY` clause to retain the same `DB_NAME` and `DBID` as that of its primary database.
- The `ALTER DATABASE CONVERT TO PHYSICAL STANDBY` statement converts the original primary database from a logical standby to a physical standby database.
- The original primary database is actually upgraded through Redo Apply after it is converted from the transient logical standby database role to a physical standby database.

Note that for databases originating with the first patch set of Oracle Database 12*c* Release 1 (12.1), the preferred method for performing a rolling upgrade with an

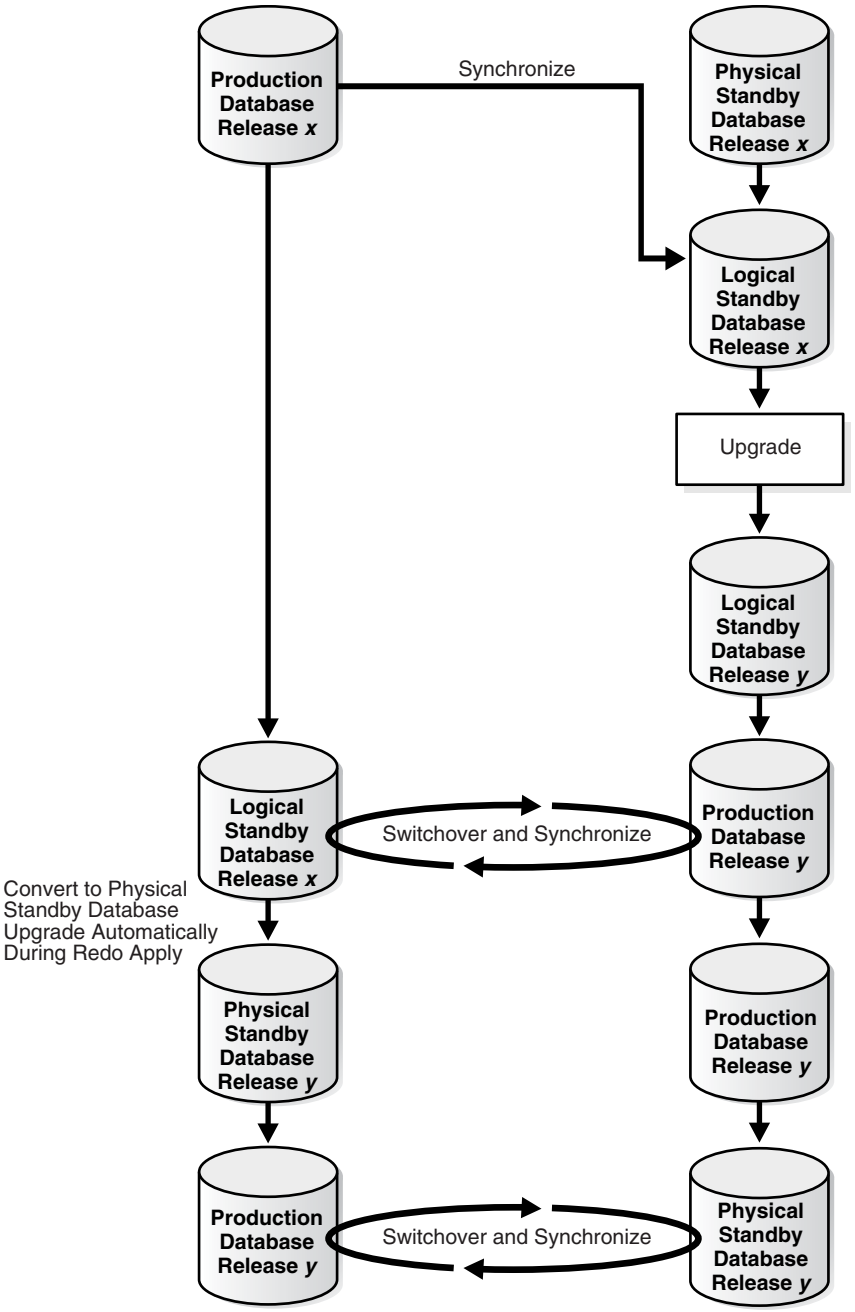
existing physical standby database is to use the DBMS_ROLLING PL/SQL package. Refer to *Oracle Data Guard Concepts and Administration*.

Figure 13–1 shows the flow of processing that occurs when you perform a rolling upgrade with a transient logical standby database.

Note: To simplify the operation shown in Figure 13–1, a Bourne shell script is available that automates the database rolling upgrade procedure (starting with Oracle Database 11*g* Release 1). The database rolling upgrade is performed using an existing Data Guard physical standby database and the transient logical standby rolling upgrade process. The Bourne shell script, named `physru`, is available for download with details in "Oracle 11g Data Guard: Database Rolling Upgrade Shell Script" in My Oracle Support Note 949322.1 at

`https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=949322.1`

Figure 13–1 Using a Transient Logical Standby Database for Database Rolling Upgrade



See Also:

- *Oracle Data Guard Concepts and Administration* for more information about "Performing a Rolling Upgrade With an Existing Physical Standby Database"
- The MAA white paper, "Database Rolling Upgrades Made Easy by Using a Data Guard Physical Standby Database", which describes the process of automating many of the tasks associated with a database rolling upgrade, available from the MAA Best Practices area for Oracle Database at

  http://www.oracle.com/goto/maa

Upgrading with Oracle GoldenGate

Consider using Oracle GoldenGate as an alternative to Data Guard database rolling upgrades for upgrading the database software from one version to another with minimal downtime, for requirements that Oracle Data Guard is not designed to address. Oracle GoldenGate offers the following advantages over Oracle Data Guard for this purpose:

- Oracle GoldenGate can upgrade an Oracle Database in rolling fashion from an Oracle Database release before Oracle Database 10*g* (Data Guard Database Rolling Upgrades are supported beginning with Oracle Database 10*g*).
- Oracle GoldenGate can be configured for one-way replication from a later Oracle Database release to a previous Oracle Database release to enable a fast fall-back option (Oracle Data Guard can only replicate from a earlier database release to a later release). This is useful in cases where you want to operate at the new release for a period and have the option to quickly revert to the previous release should unanticipated issues arise days after production cut-over. By configuring one-way replication from the new release to the previous release, production can be switched to the prior release quickly, without losing data or incurring the time of a downgrade, while the problems are resolved.
- Oracle GoldenGate can be configured for multi-master replication between different Oracle Database releases to facilitate a zero downtime upgrade (Oracle Data Guard is a one-way replication solution). When the new Oracle release is deployed and ready for user connections, new user connections can be directed to the new release while existing user connections at the old release continue to process transactions. As existing user connections terminate, utilization of the Oracle Database operating at the previous release diminishes naturally without users perceiving any downtime. Multi-master replication keeps both databases synchronized during this transitional phase. Once all users have migrated to the new release, simpler one-way replication can maintain synchronization of the previous database release to provide a fast fall-back option as described in the previous bullet item. Note that multi-master replication is not suitable for all applications - conflict detection and resolution is required.
- If you cannot use the procedure described in Section , "Upgrading with Data Guard SQL Apply or Transient Logical Standby Database" to upgrade your database and you require zero-to-minimum downtime while performing the database or application upgrade, then configure Oracle GoldenGate to perform a database upgrade with little or no downtime. For more information, see the White Paper, "Zero-Downtime Database Upgrades Using Oracle GoldenGate" at

  `http://www.oracle.com/technetwork/middleware/goldengate/overview/ggzerodowntimedatabaseupgrades-174928.pdf`

  > **See Also:** *Oracle GoldenGate For Windows and UNIX Administrator's Guide* for more information about database upgrades using Oracle GoldenGate

Upgrading by Transporting Data

The downtime required for a database upgrade when using transporting data is determined by the time needed to:

- Place the source database user tablespaces in read-only mode.
- Perform Data Pump import of the transportable metadata and non-transportable data.

- Make available the source database data files to the upgraded target database. Ideally data files are used in their current location.

Using transporting data to perform a database upgrade is recommended when:

- You can use the data files in their current location to avoid copying data files as part of the transport process. If the target database is on a different machine, this requires that the storage is accessible to both the source and target systems.
- DBUA cannot complete within the maintenance window.
- Oracle GoldenGate or Data Guard SQL Apply cannot be used due to data type restrictions.

Upgrading Using Full Transportable Export/Import You can use full transportable export/import to upgrade a database from an Oracle Database 11g Release 2 (11.2.0.3) or later to Oracle Database 12c. To do so, install Oracle Database 12c and create an empty database. Next, use full transportable export/import to transport the Oracle Database 11g Release 2 (11.2.0.3) database into the Oracle Database 12c database.

Refer to *Oracle Database Administrator's Guide*.

Upgrading Using Transportable Tablespaces If the source database is Oracle Database 11g Release 2 (11.2.0.2) or earlier, then use transportable tablespaces to accomplish a database upgrade by transporting all user data files into a pre-created, prepared target database. Note that

The SYSTEM tablespace cannot be moved with transportable tablespaces. The target database SYSTEM tablespace contents, including user definitions and objects necessary for the application, must be built manually. Use Data Pump to move the contents of the SYSTEM tablespace.

See Also:

- *Oracle Database Administrator's Guide* for an Introduction to Transportable Tablespaces
- The MAA white paper "Database Upgrade Using Transportable Tablespaces" available at

  http://www.oracle.com/goto/maa

Database Platform or Location Migration

When you perform a database migration, the primary goal is to move your data out of an existing source system and into a new Oracle Database running the latest release. Moving your data is accomplished with tools such as Data Pump, Transportable Tablespaces, Oracle Data Guard, and Oracle GoldenGate. However, during a migration you should address two equally important items that should be goals for any migration plan:

- Simplify: during a migration, simplify your implementation. Most database environments that have evolved through different versions and different DBAs contain old information (and the current DBA might question why something is used in the system). The purpose of simplifying is to make administration easier and more reliable; this simplification leads to a more highly available system.
- Optimize: during a migration you can optimize your implementation. In many cases the migration involves an updated database version so you have new features available. While performing a migration you should consider adopting new features and practices.

Add the following steps to your migration planning to simplify and optimize:

- Consider Your Options and Your Migration Strategy
- Plan Your Migration
- Oracle Features for Platform Migration and Upgrades

Consider Your Options and Your Migration Strategy

When developing your migration strategy the first step will be to learn about your new target environment and determine how your data is going to physically move from your source system to the target system. At the heart of the target environment is the Oracle Database. As with any Oracle database upgrade or migration, you should follow the guidelines and recommendations provided in the *Oracle Database Upgrade Guide* and its Upgrade Companion document in My Oracle Support. See Oracle Upgrade Companions (Doc ID 1905086.1.)

- Update init.ora during migration.

  Take your existing init.ora file and remove parameters you consider no longer important. For changes that take parameters away from their default setting, justify the changes. For example, you might be able to remove underscore parameters that are set to work around issues found in previous releases (for example to handle an optimizer problem you resolved in a previous release).

- Update SQL during migration.

  Remove SQL hints added in a previous Oracle Database version that were put in place to force the optimizer to generate the desired plan. The optimizer generally creates a good execution plan without the need for hints when provided good statistics.

- Simplify or change schema objects during migration.

  You should consider if there are changes to the schema layout that you can make during a migration. For example, consider the following:

  – Changes in the partitioning scheme for large tables

  – Adoption of newly available compression capabilities, such as Hybrid Columnar Compression (HCC) if migrating to Oracle Exadata Database Machine (see *Oracle Database Concepts* for more information)

  – Adoption of Transparent Data Encryption (TDE), especially if migrating to a system that provides cryptographic hardware acceleration

  Also, determine if there are objects that should not be migrated, such as excessive use of indexes. If you are going to have altered or fewer schema objects in the database you must consider whether it is better to migrate the database in its current form, then perform the changes after migration, or be more selective during the migration.

- Remove unused tablespaces and data files during migration.

  You should consider if you can remove unused or unnecessary tablespaces and data files during a migration. Using fewer tablespaces and data files leads to better manageability and performance.

Plan Your Migration

As you plan the migration consider the following points:

- Consider upgrading the source database to match the target version as this may improve the migration (in some cases significantly). For example, the parallel capabilities of Data Pump are continually enhanced with each new Oracle Database release, so a database export from the source system could be improved and completed faster if the source database is upgraded to match the target version.
- Consider dropping schema objects that are not needed in the source database before the migration. This can reduce the amount of data that has to be migrated.
- Determine and consider the business needs and downtime requirements. Review the Oracle features for platform migration in Section , "Oracle Features for Platform Migration and Upgrades," for the factors that influence the amount of downtime required.
- Consider whether there is a requirement or an opportunity to perform the migration in stages. For example, if there is a large amount of read only data in the source database, it might be migrated well before the live data migration to reduce downtime.
- Any platform migration exercise should include a significant amount of testing.

Oracle Features for Platform Migration and Upgrades

The following Oracle features are available to perform platform migrations and upgrades:

- Physical Standby Databases for Platform Migration
- Transportable Database for Platform Migration
- Oracle GoldenGate for Platform Migration
- Oracle Data Pump for Platform Migration
- Transportable Tablespaces for Platform Migration
- Data Guard Redo Apply (Physical Standby Database) for Location Migration

The method you choose to perform these database maintenance tasks depends on the following considerations:

- Downtime required to complete the maintenance operations
- Setup time and effort required before the downtime
- Amount of temporary additional resources necessary, such as disk space or CPU
- Complexity of the steps allowed to complete maintenance operations

Table 13–4 summarizes the methods you can use for platform migrations and database upgrades, and recommends which method to use for each operation.

Table 13–4 Platform and Location Migration Options

Operation	Recommended Method	Alternate Methods
Platform migration to same endian platform	Physical Standby Databases for Platform Migration	1. Use Transportable Database for Platform Migration when a cross-platform physical standby database is not available for the platform combination to be migrated. 2. Use Oracle GoldenGate for Platform Migration transportable database cannot finish within the maintenance window.

Table 13–4 (Cont.) Platform and Location Migration Options

Operation	Recommended Method	Alternate Methods
Platform migration to different endian platform	Oracle Data Pump for Platform Migration	1. Use Oracle GoldenGate for Platform Migration when Data Pump cannot finish within the maintenance window. 2. Use Transportable Tablespaces for Platform Migration when the database is using data types unsupported by Oracle GoldenGate.
Location Migration Only	Data Guard Redo Apply (Physical Standby Database) for Location Migration	None.

Note: Query the `V$TRANSPORTABLE_PLATFORM` view to determine the endian format of all platforms. Query the `V$DATABASE` view to determine the platform ID and platform name of the current system.

Physical Standby Databases for Platform Migration

The recommended approach for platform migration is to create a physical standby and perform a switchover. Physical standby databases support certain heterogeneous platform combinations. For an up-to-date list of platform combinations, see "Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration" in My Oracle Support Note 413484.1 at

`https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=413484.1`

Oracle Data Guard and physical standby databases are the recommended solution for performing system and cluster upgrades that are not upgradeable using Oracle RAC rolling upgrades. For example, Data Guard is also recommended for:

- System upgrades that cannot be upgraded using Oracle RAC rolling upgrades due to system restrictions.
- Migrations to Oracle ASM, to Oracle RAC from a nonclustered environment, to 64-bit systems, to a different platform with the same endian format or to a different platform with the same processor architecture, or to Windows from Linux or to Linux from Windows.
- When you have a primary database with 32-bit Oracle binaries on Linux 32-bit, and a physical standby database with 64-bit Oracle binaries on Linux 64-bit. Such configurations must follow additional procedures during Data Guard role transitions (switchover and failover) as described in Support Note 414043.1.

See Also:

- See "Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration" in My Oracle Support Note 413484.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=413484.1

- See "Role Transitions for Data Guard Configurations Using Mixed Oracle Binaries" in My Oracle Support Note 414043.1 at

  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=414043.1

Transportable Database for Platform Migration

Transportable database is the recommended solution for migrating an entire database to another platform that has the same endian format, but only when a cross-platform physical standby database is not available for the source/target platform combination to be migrated.

Consider the following points when deciding if transportable database is the appropriate method to use when moving a database to another platform:

- Transportable database supports moving databases between platforms with the same endian format.
- Downtime required for a platform migration when using transportable database is determined by the time needed to:
  - Place the source database in read-only mode.
  - Convert data files. Only files that contain undo segments, or files that contain automatic segment-space management (ASSM) segment headers if converting from or to HP Tru64, require conversion.
  - Transfer all data files from the source system to the target system.

    You can significantly reduce the amount of downtime by using a storage infrastructure that can make the data files available to the target system without physically moving the files.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more information about cross platform use of transportable database
- The MAA white paper "Platform Migration Using Transportable Database" at

  http://www.oracle.com/goto/maa

Oracle GoldenGate for Platform Migration

You can use Oracle GoldenGate to move a database from one platform to another with minimal downtime. Consider using Oracle GoldenGate if transportable database cannot perform the migration quickly enough, when the application does not use user-defined types, and you can perform any extra administrative effort required to perform the migration.

Note the following points when deciding if Oracle GoldenGate is an appropriate method for performing a platform migration:

- Oracle GoldenGate does not support user-defined types, such as object types, REF values, varrays, and nested tables.
- Extra administrative effort may be required to set up and maintain the Oracle GoldenGate environment.
- Downtime required for a platform migration when using Oracle GoldenGate is determined by the time needed to apply the remaining transactions in the queue and to reconnect clients to the new database.

See: *Oracle GoldenGate For Windows and UNIX Administrator's Guide*

Oracle Data Pump for Platform Migration

Oracle Data Pump technology enables very high-speed movement of data and metadata from one database to another, across different platforms and different database versions.

Note the following when deciding if Data Pump is an appropriate method for a platform migration:

- Downtime required for a platform migration when using Data Pump is determined by the time needed to perform a full database export, transfer the export dump files to the target system, then perform a full database import.
- Downtime may be reduced by performing the export to storage that is shared between the source and target systems, thus eliminating the need to transfer the export dump files.
- Data Pump supports the ability to load the target database directly from the source database over database links, known as network import. In some cases a network import may be faster than the multi-step approach of export database, transfer dump files, and import database.

Use Data Pump when moving a database to a platform with different endian format when the network import time is acceptable.

See Also:

- *Oracle Database Utilities* for more information about Oracle Data Pump
- *Oracle Database Upgrade Guide* for more information about upgrading your Oracle Database software

Transportable Tablespaces for Platform Migration

Transportable tablespaces accomplish a platform migration by transporting all user data files into a pre-created, prepared target database. Use transportable tablespaces when the database is using data types unsupported by Oracle GoldenGate and the user schemas are simple.

Note the following points when deciding if transportable tablespaces is the appropriate method for performing a platform migration:

- The `SYSTEM` tablespace cannot be moved with transportable tablespaces. The target database `SYSTEM` tablespace contents, including user definitions and objects necessary for the clients, must be built manually. Use Data Pump to move the necessary contents of the `SYSTEM` tablespace.
- Downtime required for a platform migration or database upgrade when using transportable tablespaces is determined by the time needed to:

- Place the source database tablespaces in read-only mode.
- Perform a network import of the transportable metadata.
- Transfer all data files from the source system to the target system.

  This time can be reduced significantly by using a storage infrastructure that can make the data files available to the target system without the physically moving the files.
- Convert all data files to the new platform format using RMAN.

Use transportable tablespaces to migrate to a platform when Oracle Data Pump cannot complete within the maintenance window, and Oracle GoldenGate or Data Guard SQL Apply cannot be used due to data type restrictions.

> **See Also:** *Oracle Database Administrator's Guide* for more information about transportable tablespaces

Data Guard Redo Apply (Physical Standby Database) for Location Migration

You can use Data Guard Redo Apply to change the location of a database to a remote site with minimal downtime by setting up a temporary standby database at a remote location and performing a switchover operation.

The downtime required for a location migration when using Data Guard Redo Apply is determined by the time required to perform a switchover operation.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about Redo Apply and physical standby databases

Edition-Based Redefinition for Online Application Maintenance and Upgrades

Edition-based redefinition enables you to upgrade a database component of an application while it is in use, thereby minimizing or eliminating down time. This is accomplished by changing (redefining) database objects in a private environment known as an edition.

To upgrade an application while it is in use, you copy the database objects that comprise the application and redefine the copied objects in isolation. Your changes do not affect users of the application—they continue to run the unchanged application. When you are sure that your changes are correct, you make the upgraded application available to all users.

In favorable cases, rollover is possible. You can use the pre-upgrade and the post-upgrade editions concurrently so that sessions that were started before the post-upgrade edition was published can continue to use the pre-upgrade edition until they are terminated naturally while new sessions use the post-upgrade edition. In less favorable cases, all pre-upgrade sessions must be terminated before new sessions can be allowed to use the post-upgrade edition. In such cases, the application suffers a small amount of downtime.

> **See:**
>
> - *Oracle Database Advanced Application Developer's Guide* for information about Edition-Based Redefinition
> - *Oracle Database Administrator's Guide* for information about Managing Editions

Oracle GoldenGate for Online Application Upgrades

An application upgrade may include a database upgrade plus any required application code and schema changes. If you require zero-to-minimum downtime while performing the database or application upgrade, then use Oracle GoldenGate to perform a database upgrade with little or no downtime. Oracle GoldenGate provides continuous system availability and eliminates planned outages to allow uninterrupted business operations.

> **See Also:** *Oracle GoldenGate For Windows and UNIX Administrator's Guide*

Data Reorganization and Redefinition

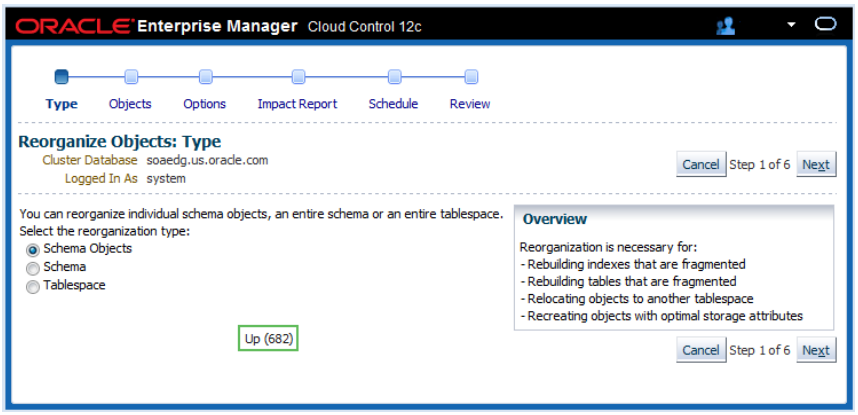
Many scheduled outages related to the data server involve some reorganization of the database objects. The Online Reorganization and Redefinition feature of Oracle Database enables data reorganization to be performed even while the underlying data is being modified. This feature enhances availability and manageability by allowing users full access to the database during a data reorganization operation.

In highly available systems, it is occasionally necessary to redefine large tables that are constantly accessed to improve the performance of queries or DML. Using Online Reorganization and Redefinition, administrators have the flexibility to modify table physical attributes and transform both data and table structure at the same time users have full access to the database. This capability improves data availability, query performance, response time, and disk space usage, all of which are important in a mission-critical environment. Plus, Online Reorganization and Redefinition can make the application upgrade process easier, safer and faster.

The recommended best practice is to reorganize tables using the `DBMS_REDEFINITION` PL/SQL package, because it provides a significant increase in availability compared to traditional methods of redefining tables that require tables to be taken offline. Whether you call `DBMS_REDEFINITION` manually at the command line or using Oracle Enterprise Manager Reorganize Objects wizard, the entire reorganization process occurs while users have full access to the table, thus ensuring system availability.

Figure 13–2 shows a page in the Oracle Enterprise Manager Reorganize Objects wizard that you can use as an alternative to calling the `DBMS_REDEFINITION` package at the SQL*Plus command line. After you answer a few questions in the wizard, it generates a script and performs the reorganization.

Figure 13–2 Database Object Reorganization Using Oracle Enterprise Manager



Consider the following when performing data reorganization:

- Minimize concurrent activity on the table during an online operation.

  During an online operation, Oracle recommends users minimize activities on the base table. Database activities should affect less than 10% of the table while an online operation is in progress. Also the database administrator can use the Database Resource Manager to minimize the affect of the data reorganization to users by allocating enough resources to the users.

- Oracle does not recommend running online operations at peak times or running a batch job that modifies a large amount of data during an online data reorganization.

- Rebuild indexes online versus dropping an index and then re-creating an index online.

  Rebuilding an index online requires additional disk space for the new index during the operation, whereas dropping an index and then re-creating an index does not require additional disk space.

- Coalesce an index online versus rebuilding an index online.

  Online index coalesce is an in-place data reorganization operation, hence does not require additional disk space like index rebuild does. Index rebuild requires temporary disk space equal to the size of the index plus sort space during the operation. Index coalesce does not reduce the height of the B-tree. It only tries to reduce the number of leaf blocks. The coalesce operation does not free up space for users but does improve index scan performance.

  If a user must move an index to a new tablespace, use online index rebuild.

- Perform online maintenance of local and global indexes.

  Oracle Database 11*g* supports both local and global partitioned indexes with online operations. When tables and indexes are partitioned, this allows administrators to perform maintenance on these objects, one partition at a time, while the other partitions remain online.

See Also:

- *Oracle Database Administrator's Guide* for more information about redefining tables online
- The Online Reorganization link from the Oracle Database High Availability page at

  `http://www.oracle.com/technetwork/database/features/availability/index.html`

Dynamic Database Services for System Maintenance

For a scheduled outage that requires an instance, node, or other component to be isolated, Oracle RAC provides the ability to relocate, disable, and enable services.

Clients will not be impacted by node or instance maintenance if they are using FAN-aware connection pools such as UCP, ODP.NET, and OCI Session Pools. The essential steps needed to successfully manage client services for node or instance maintenance are:

1. Check current status of the services and related instances to ensure services can be moved successfully.
2. Stop services normally (not using FORCE option) on the node to be maintained.

3. Disable the service so clusterware does not bring up the service until the proper time should an unplanned outage occur.
4. Disconnect long-running sessions after the current transaction completes.
5. Repeat Steps 2 - 4 for all services affected by the maintenance.
6. Shutdown the database instance.
7. Perform desired maintenance.
8. Start up instance(s) on the node.
9. Enable services that were previously disabled.
10. Start services that were stopped.
11. Observe that connections are appearing on the service again.
12. Repeat this procedure for each node or instance that needs to be shut down for maintenance.

Relocation migrates a service to another instance. Services and instances can be selectively disabled while repair, change, or upgrade is performed on hardware or system software and reenabled after the maintenance is complete. This ensures that the service or instance is not started during the maintenance outage. The service and instance is disabled at the beginning of the planned outage. It is then enabled after the maintenance outage.

When using Oracle RAC, Oracle Clusterware daemons start automatically at the time the node is started. When performing maintenance that requires one or more system restarts or requires that all non-operating system processes be shut down, use the `crsctl` command to stop and disable the startup of the Oracle Clusterware daemons. After maintenance is complete, enable and start the Oracle Clusterware daemons with `crsctl` commands.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for information about Dynamic Database Services
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about using the `crsctl` command
- The MAA white paper: "Optimizing Availability During Planned Maintenance Using Oracle Clusterware and Oracle RAC" at

  `http://www.oracle.com/goto/maa`
- My Oracle Support note 1593712.1 "Graceful Application Switchover in RAC with No Application Interruption"

Glossary

Oracle Active Data Guard option

A physical standby database can be open for read-only access while Redo Apply is active if a license for the Oracle Active Data Guard option has been purchased. This capability, known as **real-time query**, also provides the ability to have block-change tracking on the standby database, thus allowing incremental backups to be performed on the standby.

clusterwide failure

The whole cluster hosting the Oracle RAC database is unavailable or fails. This includes failures of nodes in the cluster, and any other components that result in the cluster being unavailable and the Oracle database and instances on the site being unavailable.

computer failure

An outage that occurs when the system running the database becomes unavailable because it has crashed or is no longer accessible.

data corruption

A corrupt block is a block that has been changed so that it differs from what Oracle Database expects to find. Block corruptions fall under two categories: physical and logical block corruptions.

hang or slow down

Hang or slow down occurs when the database or the application cannot process transactions because of a resource or lock contention. Perceived hang can be caused by lack of system resources.

human error

An outage that occurs when unintentional or malicious actions are committed that cause data in the database to become logically corrupt or unusable. The service level impact of a human error outage can vary significantly depending on the amount and critical nature of the affected data.

logical unit numbers (LUNs)

Three-bit identifiers used on a SCSI bus to distinguish between up to eight devices (logical units) with the same SCSI ID.

lost write

A lost write is another form of **data corruption** that can occur when an I/O subsystem acknowledges the completion of the block write, while in fact the write did not occur in the persistent storage. No error is reported by the I/O subsystem back to Oracle.

MAA environment

The Maximum Availability architecture provides the most comprehensive set of solutions for both unplanned and because it inherits the capabilities and advantages of both Oracle Database 11*g* with Oracle RAC and Oracle Database 11*g* with Data Guard.

MAA involves high availability best practices for all Oracle products across the entire technology stack—Oracle Database, Oracle WebLogic Server, Oracle Applications, Oracle Collaboration Suite, and Enterprise Manager.

network server processes

The Data Guard network server processes, also referred to as LNS*n* processes, on the primary database perform a network send to the RFS process on the standby database. There is one network server process for each destination.

real-time query

If a license for the **Oracle Active Data Guard option** has been purchased, you can open a physical standby database while Redo Apply continues to apply redo data received from the primary database.

recovery point objective (RPO)

The maximum amount of data an IT-based business process may lose before causing harm to the organization. RPO indicates the data-loss tolerance of a business process or an organization in general. This data loss is often measured in terms of time, for example, five hours or two days worth of data loss.

recovery time objective (RTO)

The maximum amount of time that an IT-based business process can be down before the organization suffers significant material losses. RTO indicates the downtime tolerance of a business process or an organization in general.

site failure

An outage that occurs when an event causes all or a significant portion of an application to stop processing or slow to an unusable service level. A site failure may affect all processing at a data center, or a subset of applications supported by a data center.

snapshot standby database

An updatable standby database that you create from a physical standby database. A snapshot standby database receives and archives redo data received from the primary database, but the snapshot standby database does not apply redo data from the primary database while the standby database is open for read/write I/O. Thus, the snapshot standby typically diverges from the primary database over time. Moreover, local updates to the snapshot standby database cause additional divergence. However, a snapshot standby protects the primary database because the snapshot standby can be converted back into a physical standby database.

storage failure

An outage that occurs when the storage holding some or all of the database contents becomes unavailable because it has shut down or is no longer accessible.

transient logical standby database

A transient logical standby database is a physical standby database that has been temporarily converted into a logical standby database to perform a rolling database upgrade.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z