### **Oracle® Database**

High Availability Best Practices 10*g* Release 2 (10.2) **B25159-01** 

July 2006



Oracle Database High Availability Best Practices, 10g Release 2 (10.2)

B25159-01

Copyright © 2006, Oracle. All rights reserved.

Contributing Authors: Andrew Babb, Tammy Bednar, Immanuel Chan, Timothy Chien, Craig B. Foch, Michael Nowak, Viv Schupmann, Michael Todd Smith, Vinay Srihari, Lawrence To, Randy Urbano, Douglas Utzig, James Viscusi

Contributors: Larry Carpenter, Joseph Meeks, Ashish Ray (coauthors of MAA white papers)

Contributor: Valarie Moore (graphic artist)

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

# Contents

eface		ix
Audie	nce	ix
Docur	nentation Accessibility	ix
Relate	d Documents	x
Conve	entions	х
Intro	duction to High-Availability Best Practices	
1.1	Oracle Database High-Availability Architecture	. 1-1
1.2	Oracle Database High-Availability Best Practices	. 1-1
1.3	Oracle Maximum Availability Architecture	. 1-2
1.4	Operational Best Practices	. 1-2
Confi	guring for High-Availability	
2.1	Configuring Storage	. 2-1
2.1.1	Evaluate Database Performance Requirements and Storage Performance Capabili <b>2-2</b>	ties
2.1.2	Use Automatic Storage Management (ASM) to Manage Database Files	. 2-2
2.1.3	Use a Simple Disk and Disk Group Configuration	. 2-3
2.1.4	Use Disk Multipathing Software to Protect from Path Failure	. 2-5
2.1.5	Use Redundancy to Protect from Disk Failure	. 2-5
2.1.6	Consider HARD-Compliant Storage	. 2-7
2.2	Configuring Oracle Database 10g	. 2-7
2.2.1	Requirements for High Availability	. 2-8
2.2.2	Recommendations for High Availability and Fast Recoverability	. 2-9
2.2.3	Recommendations to Improve Manageability	2-13
2.3	Configuring Oracle Database 10g with RAC	2-16
2.3.1	Connect to Database using Services and Virtual Internet Protocol (VIP) Address	2-16
2.3.2	Use Oracle Clusterware to Manage the Cluster and Application Availability	2-17
2.3.3	Use Client-Side and Server-Side Load Balancing	2-17
2.3.4	Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks	2-18
2.3.5	Regularly Back Up OCR to Tape or Offsite	2-18
2.3.6	Verify That CRS and RAC Use Same Interconnect Network	2-19
2.3.7	Configure All Databases for Maximum Instances in the Cluster	2-19
2.4	Configuring Oracle Database 10g with Data Guard	2-20
2.4.1	Physical or Logical Standby	2-21
	Audie Docur Relate Conve 1.1 1.2 1.3 1.4 <b>Confi</b> 2.1 2.1.2 2.1.3 2.1.4 2.1.2 2.1.3 2.1.4 2.1.5 2.1.6 2.2 2.2.1 2.2.2 2.2.3 2.3.1 2.3.2 2.3.3 2.3.4 2.3.5 2.3.6 2.3.7 2.4	1.2       Oracle Database High-Availability Best Practices         1.3       Oracle Maximum Availability Architecture         1.4       Operational Best Practices         1.4       Operational Best Practices         2.1       Configuring for High-Availability         2.1.1       Evaluate Database Performance Requirements and Storage Performance Capabili 2-2         2.1.2       Use Automatic Storage Management (ASM) to Manage Database Files         2.1.3       Use a Simple Disk and Disk Group Configuration         2.1.4       Use Disk Multipathing Software to Protect from Path Failure         2.1.5       Use Redundancy to Protect from Disk Failure         2.1.6       Configuring Oracle Database 10g         2.2       Requirements for High Availability         2.2.3       Recommendations to Improve Manageability         2.2.4       Recommendations to Improve Manageability         2.2.3       Configuring Oracle Database 10g with RAC         2.3       Connect to Database using Services and Virtual Internet Protocol (VIP) Address         2.3.2       Use Oracle Clusterware to Manage the Cluster and Application Availability         2.3.3       Use Client-Side and Server-Side Load Balancing         2.3.4       Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks         2.3.5       Regularly Back Up OCR to Ta

2.4.2	Data Protection Mode	2-23
2.4.3	Number of Standby Databases	2-24
2.4.4	General Configuration Best Practices for Data Guard	2-25
2.4.5	Redo Transport Services Best Practices	2-29
2.4.6	Log Apply Services Best Practices	2-33
2.4.7	Role Transition Best Practices	2-37
2.4.8	Maintaining a Physical Standby Database as a Clone	2-41
2.4.9	Recommendations on Protecting Data Outside of the Database	2-43
2.4.10	Assessing Data Guard Performance	2-43
2.5	Configuring Backup and Recovery	2-45
2.5.1	Use Oracle Database Features and Products	2-46
2.5.2	Configuration and Administration	2-47
2.5.3	Backup to Disk	2-49
2.5.4	Backup to Tape	2-52
2.5.5	Backup and Recovery Maintenance	2-52
2.6	Configuring Fast Application Failover	2-53
2.6.1	Configuring Clients for Failover	2-54
2.6.2	Client Failover in a RAC Database	2-54
2.6.3	Failover from a RAC Primary Database to a Standby Database	2-55

# 3 Monitoring Using Oracle Grid Control

3.1	Overview of Monitoring and Detection for High Availability	3-1
3.2	Using Oracle Grid Control for System Monitoring	3-1
3.2.1	Set Up Default Notification Rules for Each System	3-3
3.2.2	Use Database Target Views to Monitor Health, Availability, and Performance	3-6
3.2.3	Use Event Notifications to React to Metric Changes	3-8
3.2.4	Use Events to Monitor Data Guard System Availability	3-8
3.3	Managing the High-Availability Environment with Oracle Grid Control	3-9
3.3.1	Check Oracle Grid Control Policy Violations	3-9
3.3.2	Use Oracle Grid Control to Manage Oracle Patches and Maintain System Baselines <b>3-9</b>	
3.3.3	Use Oracle Grid Control to Manage Data Guard Targets	3-10

# 4 Managing Outages

4.1	Outage Overview	4-1
4.1.1	Unscheduled Outages	4-1
4.1.2	Scheduled Outages	4-5
4.2	Recovering from Unscheduled Outages	4-9
4.2.1	Complete Site Failover	4-10
4.2.2	Database Failover with a Standby Database	4-13
4.2.3	Database Switchover with a Standby Database	4-19
4.2.4	RAC Recovery for Unscheduled Outages	4-23
4.2.5	Application Failover	4-25
4.2.6	ASM Recovery After Disk and Storage Failures	4-25
4.2.7	Recovering from Data Corruption (Data Failures)	4-34
4.2.8	Recovering from Human Error	4-37
4.3	Restoring Fault Tolerance	4-44

4.3.1	Restoring Failed Nodes or Instances in a RAC Cluster	4-45
4.3.2	Restoring a Standby Database After a Failover	4-50
4.3.3	Restoring ASM Disk Groups after a Failure	4-52
4.3.4	Restoring Fault Tolerance After Planned Downtime on Secondary Site or Clusterv Outage <b>4-53</b>	vide
4.3.5	Restoring Fault Tolerance After a Standby Database Data Failure	4-54
4.3.6	Restoring Fault Tolerance After the Production Database Was Opened Resetlogs	4-55
4.3.7	Restoring Fault Tolerance After Dual Failures	4-57
4.4	Eliminating or Reducing Downtime for Scheduled Outages	4-57
4.4.1	Storage Maintenance	4-57
4.4.2	RAC Database Patches	4-58
4.4.3	Database Upgrades	4-61
4.4.4	Database Platform or Location Migration	4-63
4.4.5	Online Database and Application Upgrades	4-66
4.4.6	Database Object Reorganization	4-68
4.4.7	System Maintenance	4-70

# 5 Migrating to an MAA Environment

5.1	Overview of Migrating to MAA	5-1
5.2	Migrating to RAC from a Single Instance	5-2
5.3	Adding a Data Guard Configuration to a RAC Primary	5-2

# A Database SPFILE and Oracle Net Configuration File Samples

A.1	SPFILE Samples	A-2
A.2	Oracle Net Configuration Files	A-6
A.2.1	SQLNET.ORA Example for All Hosts Using Dynamic Instance Registration	A-6
A.2.2	LISTENER.ORA Example for All Hosts Using Dynamic Instance Registration	A-7
A.2.3	TNSNAMES.ORA Example for All Hosts Using Dynamic Instance Registration	A-7

# Glossary

Index

# List of Figures

Allocating Entire Disks	2-4
Partitioning Each Disk	2-4
LGWR ASYNC Archival with Network Server (LNSn) Processes	2-31
Oracle Grid Control Home Page	. 3-2
Setting Notification Rules for Availability	. 3-4
Setting Notification Rules for Metrics	. 3-6
Overview of System Performance	. 3-7
Network Routes After Site Failover	4-12
Data Guard Overview Page Showing ORA-16625 Error	4-16
Failover Confirmation Page	4-16
Failover Progress Page	4-17
Data Guard Overview Page After a Failover Completes	4-18
Switchover Operation Confirmation	4-21
Processing Page During Switchover	4-21
New Primary Database After Switchover	4-22
Enterprise Manager Reports Disk Failures	4-28
Enterprise Manager Reports ASM Disk Groups Status	4-29
Enterprise Manager Reports Pending REBAL Operation	4-29
	4-48
	4-49
Nonpartitioned RAC Instances	4-50
Fast-Start Failover and the Observer Are Successfully Enabled	4-52
Reinstating the Former Primary Database After a Fast-Start Failover	4-52
Online Database Upgrade with Oracle Streams	4-67
Database Object Reorganization Using Oracle Enterprise Manager	4-69
	Failover Confirmation Page Failover Progress Page Data Guard Overview Page After a Failover Completes Switchover Operation Confirmation Processing Page During Switchover New Primary Database After Switchover Enterprise Manager Reports Disk Failures Enterprise Manager Reports ASM Disk Groups Status Enterprise Manager Reports Pending REBAL Operation Partitioned Two-Node RAC Database RAC Instance Failover in a Partitioned Database Nonpartitioned RAC Instances Fast-Start Failover and the Observer Are Successfully Enabled Reinstating the Former Primary Database After a Fast-Start Failover Online Database Upgrade with Oracle Streams

### List of Tables

2–1	Determining the Appropriate Protection Mode	
2–2	Archiving Recommendations	2-27
2–3	Minimum Recommended Settings for FastStartFailoverThreshold	2-40
2–4	Comparison of Backup Options	2-50
2–5	Typical Wait Times for Client Failover	
3–1	Recommendations for Monitoring Space	3-5
3–2	Recommendations for Monitoring the Alert Log	3-5
3–3	Recommendations for Monitoring Processing Capacity	. 3-5
3–4	Recommended Notification Rules for Metrics	
3–5	Recommendations for Setting Data Guard Events	3-9
4–1	Unscheduled Outages	4-2
4–2	Recovery Times and Steps for Unscheduled Outages on the Primary Site	. 4-3
4–3	Recovery Steps for Unscheduled Outages on the Secondary Site	4-5
4–4	Scheduled Outages	
4–5	Recovery Steps for Scheduled Outages on the Primary Site	. 4-7
4–6	Managing Scheduled Outages on the Secondary Site	
4–7	Types of ASM Failures and Recommended Repair	4-26
4–8	Recovery Options for Data Area Disk Group Failure	4-30
4–9	Recovery Options for Flash Recovery Area Disk Group Failure	4-32
4–10	Non Database Object Corruption and Recommended Repair	4-35
4–11	Flashback Solutions for Different Outages	4-38
4–12	Summary of Flashback Features	4-38
4–13	Additional Processing When Restarting or Rejoining a Node or Instance	4-45
4–14	Restoration and Connection Failback	4-47
4–15	SQL Statements for Starting Physical and Logical Standby Databases	4-53
4–16	SQL Statements to Start Redo Apply and SQL Apply	4-53
4–17	Queries to Determine RESETLOGS SCN and Current SCN OPEN RESETLOGS	4-55
4–18	SCN on Standby Database is Behind Resetlogs SCN on the Production Database	4-55
4–19	SCN on the Standby is Ahead of Resetlogs SCN on the Production Database	4-56
4–20	Re-Creating the Production and Standby Databases	4-57
4–21	Platform Migration and Database Upgrade Options	4-61
4–22	Platform and Location Migration Options	4-64
4–23	Some Object Reorganization Capabilities	4-68
5–1	Starting configurations Before Migrating to an MAA Environment	5-1
A–1	Generic SPFILE Parameters for Primary, Physical Standby, and Logical Standby Data A-2	bases
A–2	RAC SPFILE Parameters for Primary, Physical Standby, and Logical Standby Databas A-3	ses
A–3	Data Guard SPFILE Parameters for Primary, Physical Standby, and Logical Standby	
	Databases A-3	
A–4	Data Guard Broker SPFILE Parameters for Primary, Physical Standby, and Logical	
	Standby Databases A-4	
A–5	Data Guard (No Broker) SPFILE Parameters for Primary, Physical Standby, and Logic	cal
	Standby Databases A-4	
A–6	Data Guard SPFILE Parameters for Primary and Physical Standby Database Only	A-4
A–7	Data Guard SPFILE Parameters for Primary and Logical Standby Database Only	
A–8	Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, an	
	Logical Standby Database: Maximum Availability or Maximum Protection Modes A-	
A–9	Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, and Logical Standby Database: Maximum Performance Mode A-6	

# Preface

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

### Audience

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

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

## **Documentation Accessibility**

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

#### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

#### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

#### **TTY Access to Oracle Support Services**

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

### **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 Database Oracle Clusterware and Oracle Real Application Clusters Installation *Guide* for your platform
- Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide
- Oracle Database Backup and Recovery Advanced User's Guide
- Oracle Database Administrator's Guide

Oracle High Availability Best Practice white papers can be downloaded at http://www.oracle.com/technology/deploy/availability/htdocs/maa.
htm

# 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

This chapter describes how using Oracle high-availability best practices can increase availability to the Oracle database as well as the entire technology stack. This chapter contains the following topics:

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

### 1.1 Oracle Database High-Availability Architecture

Choosing and implementing the architecture that best fits the availability requirements of a business can be a daunting task. This architecture must encompass appropriate redundancy, provide adequate protection from all types of outages, ensure consistent high performance and robust security, while being easy to deploy, manage, and scale. Needless to mention, this architecture should be driven by well-understood business requirements. Choosing and implementing a high-availability architecture is covered in *Oracle Database High Availability Overview*.

Before using the best practices presented in this book, your organization should have already chosen a high-availability architecture for your database as described in *Oracle Database High Availability Overview*. If you have not already done so, then refer to that document to learn about the high-availability solutions that Oracle offers for Oracle Database before proceeding with this book.

# 1.2 Oracle Database High-Availability Best Practices

To build, implement and maintain a high-availability architecture, a business needs high-availability best practices that involve both technical and operational aspects of its IT systems and business processes. Such a set of best practices removes the complexity of designing a high-availability architecture, maximizes availability while using minimum system resources, reduces the implementation and maintenance costs of the high-availability systems in place, and makes it easy to duplicate the high-availability architecture in other areas of the business. An enterprise with a well-articulated set of high-availability best practices that encompass high-availability analysis frameworks, business drivers and system capabilities, will enjoy an improved operational resilience and enhanced business agility.

Building, implementing, and maintaining a high-availability architecture for Oracle Database using high-availability best practices is the purpose of this book. By using the Oracle Database high-availability best practices described in this book, you will be able to:

- Reduce the implementation cost of an Oracle Database high-availability system by following detailed guidelines on configuring your database, storage, application failover, backup and recovery as described in Chapter 2, "Configuring for High-Availability"
- Avoid potential downtime by monitoring and maintaining your database using Oracle Grid Control as described in Chapter 3, "Monitoring Using Oracle Grid Control"
- Recover quickly from unscheduled outages caused by computer failure, storage failure, human error, or data corruption as described in Chapter 4, "Managing Outages"
- Eliminate or reduce downtime that might occur due to scheduled maintenance such as database patches or application upgrades as described in Chapter 4, "Managing Outages"

# 1.3 Oracle Maximum Availability Architecture

Oracle Maximum Availability Architecture (MAA) is an Oracle best practices blueprint based on proven Oracle high-availability technologies and recommendations. The high-availability best practices described in this book make up one of several components of MAA. MAA involves high-availability best practices for all Oracle products across the entire technology stack—Oracle Database, Oracle Application Server, Oracle Applications, Oracle Collaboration Suite, and Oracle Grid Control.

Some of the key features of MAA include:

- Considers various business service level agreements (SLA) to make high-availability best practices as widely applicable as possible
- Leverages database grid servers and storage grid with low-cost storage to provide highly resilient, lower cost infrastructure
- Uses results from extensive performance impact studies for different configurations to ensure that the high-availability architecture is optimally configured to perform and scale to business needs
- Gives the ability to control the length of time to recover from an outage and the amount of acceptable data loss from a natural disaster
- Evolves with each Oracle version and is completely independent of hardware and operating system

For more information on MAA and documentation on best practices for all components of MAA, visit the MAA web site at:

http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

# **1.4 Operational Best Practices**

One of the best ways to reduce downtime is incorporating operational best practices. You can often prevent problems and downtime before they occur by rigorously testing changes in your test environment, following stringent change control policies to guard your primary database from harm, and having a well-validated repair strategy for each outage type. A monitoring infrastructure such as Grid Control is essential to quickly detect problems. Having an outage and repair decision tree as well as an automated or automatic repair facility reduces downtime by eliminating or reducing decision and repair times.

The following is a list of key operational practices:

- Document and communicate service level agreements (SLA)
- Create test environments

A good test environment accurately mimics the production system to test changes and prevent problems before they can affect your business.

Establish change control and security procedures

Change control and security procedures maintain the stability of the system and ensure that no changes are incorporated in the primary database unless they have been rigorously evaluated on your test systems.

Set up and follow security best practices

The biggest threat to corporate data comes from employees and contractors with internal access to networks and facilities. Corporate data can be at grave risk if placed on a system or database that does not have proper security measures in place. A well-defined security policy can help protect your systems from unwanted access and protect sensitive corporate information from sabotage. Proper data protection reduces the chance of outages due to security breaches.

#### See Also:

- Oracle Database Concepts for an overview of database security
- Oracle Database Security Guide for security checklists and recommendations
- Leverage Grid Control or another monitoring infrastructure to detect and react to potential failures and problems before they occur
  - Monitor system, network, and database statistics
  - Monitor performance statistics
  - Create performance thresholds as early warning indicators that a system or application has a problem or is underperforming
- Leverage MAA recommended repair strategies and create an outage and repair decision tree for crisis scenarios using the recommended MAA matrix
- Automate and optimize repair practices to minimize downtime by following MAA best practices

**See Also:** Chapter 4, "Managing Outages" for more information on repair strategies and practices

# **Configuring for High-Availability**

This chapter describes Oracle configuration best practices for Oracle Database and related components.

This chapter contains these topics:

- Configuring Storage
- Configuring Oracle Database 10g
- Configuring Oracle Database 10g with RAC
- Configuring Oracle Database 10g with Data Guard
- Configuring Backup and Recovery
- Configuring Fast Application Failover

**See Also:** Appendix A, "Database SPFILE and Oracle Net Configuration File Samples" for complete examples of database parameter settings

### 2.1 Configuring Storage

This section 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 section contains these topics:

- Evaluate Database Performance Requirements and Storage Performance Capabilities
- Use Automatic Storage Management (ASM) to Manage Database Files
- Use a Simple Disk and Disk Group Configuration
- Use Disk Multipathing Software to Protect from Path Failure
- Use Redundancy to Protect from Disk Failure
- Consider HARD-Compliant Storage

# 2.1.1 Evaluate Database Performance Requirements and Storage Performance Capabilities

Characterize your database performance requirements using different application workloads. Extract statistics during your target workloads by getting the beginning and end statistical snapshots. Example target workloads include:

- Average load
- Peak load
- Batch processing

The necessary statistics can be derived from Automatic Workload Repository (AWR) reports or gathered from the GV\$SYSSTAT view. Along with understanding the database performance requirements, the performance capabilities of a storage array must be evaluated.

Low-cost storage arrays, low-cost storage networks, and Oracle Database 10g can in combination create a low-cost storage grid with excellent performance and availability. Low-cost storage is most successfully deployed for databases with certain types of performance and availability requirements. Compared to traditional high-end storage arrays, low-cost storage arrays have excellent data throughput and superior price for each gigabyte. However, low-cost storage arrays do not have better I/O rates for OLTP type applications than traditional storage, although the cost for each I/O per second is comparable. The Oracle Resilient Low-Cost Storage Initiative is designed to help customers reduce IT spending and promote use of low-cost storage arrays in both departmental and enterprise environments.

The Oracle Database flash recovery area is an ideal candidate for low-cost storage. Because the flash recovery area contains recovery related files that are typically accessed with sequential 1MB streams, the performance characteristics of low-cost storage are well suited for the flash recovery area. The flash recovery area can be configured to use low-cost storage while the database area remains on traditional storage.

#### See Also:

- Best Practices for Creating a Low-Cost Storage Grid for Oracle Databases at http://www.oracle.com/technology/deploy/availabil ity/htdocs/maa.htm
- Oracle Resilient Low-Cost Storage Initiative Web site at http://www.oracle.com/technology/deploy/availabil ity/htdocs/lowcoststorage.html

### 2.1.2 Use Automatic Storage Management (ASM) to Manage Database Files

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

ASM should be used to manage all database files. However, ASM can be phased into your environment initially supporting only the flash recovery area. This approach is

particularly well suited for introducing low-cost storage into an existing environment where traditional storage configurations currently exist.

To improve manageability, ASMLib should be used on platforms where it is available. ASMLib is a support library for ASM. ASMLib eliminates the impact when the mappings of disk device names change upon system reboot. Although ASMLib is not required to run ASM, it 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.

#### See Also:

- Chapter 16 "Migrating Databases To and From ASM with Recovery Manager" in the Oracle Database Backup and Recovery Advanced User's Guide
- Oracle Database 10g Release 2 Automatic Storage Management Overview and Technical Best Practices at http://www.oracle.com/technology/products/databas e/asm/pdf/asm\_10gr2\_bptwp\_sept05.pdf
- Oracle ASMLib Web site at http://www.oracle.com/technology/tech/linux/asmli b/index.html
- Oracle Database Administrator's Guide for more information on configuring Automatic Storage Management

#### 2.1.3 Use a Simple Disk and Disk Group Configuration

When using ASM for database storage, you should create two disk groups: one disk group for the database area and another disk group for the flash recovery area:

• The *database area* contains active database files, such as datafiles, control files, online redo log files, Data Guard Broker metadata files, and change tracking files used for RMAN incremental backups. For example:

```
CREATE DISKGROUP DATA DISK
'/devices/lun01','/devices/lun02','/devices/lun03','/devices/lun04';
```

The *flash 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 backup sets, and flashback log files. For example:

```
CREATE DISKGROUP RECO DISK
'/devices/lun05','/devices/lun06','/devices/lun07','/devices/lun08',
'/devices/lun09','/devices/lun10','/devices/lun11','/devices/lun12';
```

To simplify file management, use Oracle managed files to control file naming. Enable Oracle managed files by setting initialization parameters DB\_CREATE\_FILE\_DEST, DB\_RECOVERY\_FILE\_DEST\_AND DB\_RECOVERY\_FILE\_DEST\_SIZE:

**Note:** Using a flash recovery area by setting DB\_RECOVERY\_FILE\_ DEST requires that you also set DB\_RECOVERY\_FILE\_DEST\_SIZE to bound the amount of disk space used by the flash recovery area. DB\_ RECOVERY\_FILE\_DEST and DB\_RECOVERY\_FILE\_DEST\_SIZE are dynamic parameters that allow you to change the destination and size of the flash recovery area. DB\_CREATE\_FILE\_DEST=+DATA DB\_RECOVERY\_FILE\_DEST=+RECO DB\_RECOVERY\_FILE\_DEST\_SIZE=500G

You have two options when partitioning disks for ASM use:

- Allocate entire disks to the database area and flash recovery area disk groups
- Partition each disk into two partitions, one for the database area and another for the flash recovery area

Figure 2–1 Allocating Entire Disks



Figure 2–1 illustrates allocating entire disks. The advantages of this option are:

- It is easier to manage the disk partitions at the operating system level, because each disk is partitioned as just one large partition.
- ASM rebalance operations following a disk failure complete more quickly, because there is only one disk group to rebalance.

The disadvantage of this option is:

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



Figure 2–2 Partitioning Each Disk

The second option is illustrated in Figure 2–2. It requires partitioning each disk into two partitions: a smaller partition on the faster outer portion of each drive for the database area, and a larger partition on the slower inner portion of each drive for the flash recovery area. The ratio for the size of the inner and outer partitions depends on the estimated size of the database area and the flash recovery area.

The advantage of this approach is:

 Higher I/O bandwidth available, because both disk groups are spread over all available spindles. This advantage is considerable for the database area disk group for I/O intensive applications.

The disadvantages are:

- A double disk failure may result in the loss of both disk groups, requiring the use of a standby database or tape backups for recovery.
- An ASM rebalance operation following a disk failure is longer, because both disk groups are affected.
- Higher initial administrative efforts are required to partition each disk properly.

#### See Also:

- Oracle Database Backup and Recovery Basics for information on setting up and sizing the flash recovery area
- Oracle Database Administrator's Guide for information on automatic storage management

#### 2.1.4 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. Disk multipathing software should be used in conjunction with ASM.

When specifying disk names during disk group creation in ASM, the logical device representing the single logical path should be used. 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 ASM, the asm\_diskstring parameter should contain /dev/dm-\* to discover the logical device /dev/dm-0, and that logical device should be used during disk group creation:

asm\_diskstring='/dev/dm-\*'

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

#### 2.1.5 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
- ASM redundancy

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 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 ASM disk group using the EXTERNAL REDUNDANCY clause:

```
CREATE DISKGROUP DATA EXTERNAL REDUNDANCY DISK

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

If the storage array does not offer the desired level of redundancy, or if there is a need to configure redundancy across multiple storage arrays, then use ASM redundancy.

ASM provides redundancy with the use of failure groups, which are defined during disk group creation. ASM redundancy can be either Normal redundancy, where files are two-way mirrored, or high redundancy, where files are three-way mirrored. Once a disk group is created, the redundancy level cannot be changed.

Failure group definition is specific to each storage setup, but these guidelines should be followed:

 If every disk is available through every I/O path, as would be the case if using disk multipathing software, then leave each disk in its own failure group. This is the default 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';
```

- 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 three examples:
  - For an array with two controllers where each controller sees only half of 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 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';
```

 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 ASM redundancy, enough free space must exist in the disk group so that when a disk fails 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 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 can be safely used in a disk group, taking mirroring into account, and yet be able to restore redundancy after a disk failure is in the column USABLE\_FILE\_MB in the V\$ASM\_DISKGROUP view. USABLE\_FILE\_MB should always be greater than zero. If USABLE\_FILE\_MB falls below zero, then more disks should be added to the disk group.

### 2.1.6 Consider HARD-Compliant Storage

Consider HARD-compliant storage for the greatest protection against data corruption. Data corruption is very rare, but it can have a catastrophic effect on a business when it occurs.

The goal of the Hardware Assisted Resilient Data (HARD) initiative is to eliminate a class of failures that the computer industry has so far been powerless to prevent. RAID has gained a wide following in the storage industry by ensuring the physical protection of data. HARD takes data protection to the next level by going beyond protecting physical data to protecting business data.

The HARD initiative is designed to prevent data corruption before it happens. Under the HARD initiative, Oracle partners with storage vendors to implement Oracle data validation and checking algorithms inside storage devices. This makes it possible to prevent corrupted data from being written to permanent storage.

The classes of data corruption that Oracle addresses with HARD include:

- Writes that physically and logically corrupt Oracle blocks
- Writes of database blocks to incorrect locations
- Writes of partial or incomplete blocks
- Writes by other applications to Oracle data blocks

End-to-end block validation is the technology employed by the operating system or storage subsystem to validate the Oracle Database data block contents. By validating Oracle Database data in the storage devices, data corruption is detected and eliminated before it can be written to permanent storage. This goes beyond the current Oracle Database block validation features that do not detect a stray, lost, or corrupted write until the next physical read.

Storage vendors who are partners with Oracle are given the opportunity to implement validation checks based on a specification. A particular vendor's implementation may offer features specific to its storage technology. Oracle maintains a Web site that shows a comparison of each vendor's solution by product and Oracle version.

**Note:** When using ASM to manage database storage, ASM should always be configured as external redundancy. Additionally, HARD protections should be disabled when doing any rebalance operations, such as adding a new disk, to avoid the risk of HARD inadvertently flagging the movement of data as a bad write.

#### See Also:

http://www.oracle.com/technology/deploy/availability
/htdocs/HARD.html for the most recent information on the HARD
initiative

## 2.2 Configuring Oracle Database 10g

The best practices discussed in this section apply to Oracle Database 10g database architectures in general, including all architectures described in *Oracle Database High Availability Overview*:

- Oracle Database 10g
- Oracle Database 10g with RAC

- Oracle Database 10g with Data Guard
- Oracle Database 10g with RAC and Data Guard (MAA)

These recommendations are identical for both the primary and standby databases when Oracle Data Guard is used. It is necessary to adopt these practices to reduce or avoid outages, reduce risk of corruption, and improve recovery performance.

This section contains the following types of best practices for configuring the database in general:

- Requirements for High Availability
  - Enable ARCHIVELOG Mode
  - Enable Block Checksums
- Recommendations for High Availability and Fast Recoverability
  - Configure the Size of Redo Log Files and Groups Appropriately
  - Use a Flash Recovery Area
  - Enable Flashback Database
  - Use Fast-Start Fault Recovery to Control Instance Recovery Time
  - Enable Database Block Checking
  - Set DISK\_ASYNCH\_IO
- Recommendations to Improve Manageability
  - 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

### 2.2.1 Requirements for High Availability

This section describes the following minimum requirements for configuring Oracle Database for high availability:

- Enable ARCHIVELOG Mode
- Enable Block Checksums

#### 2.2.1.1 Enable ARCHIVELOG Mode

ARCHIVELOG mode enables online database backup and is necessary to recover the database to a point in time later than what has already been restored. Architectures such as Oracle Data Guard and Flashback Database require that the production database run in ARCHIVELOG mode.

**See Also:** Oracle Database Administrator's Guide for more information about using automatic archiving

#### 2.2.1.2 Enable Block Checksums

By default, Oracle always validates the data blocks that it reads from disk. Enabling data and log block checksums by setting DB\_BLOCK\_CHECKSUM to TYPICAL enables Oracle to detect other types of corruption caused by underlying disks, storage systems, or I/O systems. Before a data block is written to disk, a checksum is computed and stored in the block. When the block is subsequently read from disk, the checksum is computed again and compared with the stored checksum. Any difference is treated as a media error, and an ORA-1578 error is signaled. Block checksums are always maintained for the SYSTEM tablespace. If DB\_BLOCK\_CHECKSUM is set to FULL, then in-memory corruption is also detected before being written to disk.

In addition to enabling data block checksums, Oracle also calculates a checksum for every redo log block before writing it to the current log. Redo record corruption is found as soon as the log is archived. Without this option, corruption in a redo log can go unnoticed until the log is applied to a standby database, or until a backup is restored and rolled forward through the log containing the corrupt log block.

RMAN also calculates checksums when taking backups to ensure that all blocks being backed up are validated.

Ordinarily the overhead for TYPICAL is one to two percent and for FULL is four to five percent. The default setting, TYPICAL, provides critical detection of corruption at very low cost and remains a requirement for high availability. Measure the performance impact with your workload on a test system and ensure that the performance impact is acceptable before moving from TYPICAL to FULL on an active database.

### 2.2.2 Recommendations for High Availability and Fast Recoverability

This section describes Oracle Database best practices for reducing recovery time or increasing its availability and redundancy:

- Configure the Size of Redo Log Files and Groups Appropriately
- Use a Flash Recovery Area
- Enable Flashback Database
- Use Fast-Start Fault Recovery to Control Instance Recovery Time
- Enable Database Block Checking
- Set DISK\_ASYNCH\_IO
- Set LOG\_BUFFER to At Least 8 MB
- Use Automatic Shared Memory Management
- Increase PARALLEL\_EXECUTION\_MESSAGE\_SIZE
- Tune PARALLEL\_MIN\_SERVERS

#### 2.2.2.1 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 flash recovery area. This protects against a failure involving the redo log, such as a disk or I/O failure for one of the members, 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.

All online redo log files should be the same size and configured to switch approximately once an hour during normal activity. They should not switch more frequently than every 20 minutes during peak activity.

There should be a minimum of four online log groups to prevent LGWR from waiting for a group to be available following a log switch. A group might be unavailable because a checkpoint has not yet completed or because the group has not yet been archived.

#### See Also:

- Oracle Database Administrator's Guide for more information about managing redo logs
- Oracle Data Guard Concepts and Administration for more information about online, archived, and standby redo log files
- Section 2.4, "Configuring Oracle Database 10g with Data Guard" on page 2-20

#### 2.2.2.2 Use a Flash Recovery Area

The flash recovery area is Oracle managed disk space that provides a centralized disk location for backup and recovery files. The flash recovery area is defined by setting the following database initialization parameters:

DB\_RECOVERY\_FILE\_DEST

This parameter specifies the default location for the flash recovery area.

DB\_RECOVERY\_FILE\_DEST\_SIZE

This parameter specifies (in bytes) the hard limit on the total space to be used by target database recovery files created in the recovery area location.

The flash recovery area should be the primary location for recovery. When the flash recovery area is properly sized, files needed for repair will be 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 Backup and Recovery Basics for detailed information about sizing the flash recovery area and setting the retention period

#### 2.2.2.3 Enable Flashback Database

Flashback Database enables you to rewind the database to a previous point in time without restoring backup copies of the datafiles. Flashback Database is a revolutionary recovery feature that operates on only the changed data. Flashback Database makes the time to correct an error proportional to the time to cause and detect the error, without recovery time being a function of the database size. You can flash back a database from both RMAN and SQL\*Plus with a single command instead of using a complex procedure.

During normal runtime, Flashback Database buffers and writes before images of data blocks into the flashback logs, which reside in the flash recovery area. Ensure there is sufficient I/O bandwidth available to the flash recovery area to maintain flashback write throughput. If flashback writes are slow, as evidenced by the flashback free buffer waits wait event, then database throughput is affected. The amount of disk writes caused by Flashback Database differs depending on the workload and application profile. For a typical OLTP workload that is using a flash recovery area with sufficient disk spindles and I/O throughput, the overhead incurred by Flashback Database is less than two percent.

Flashback Database can flash back a primary or standby database to a point in time prior to a role transition. In addition, a Flashback Database can be performed to a

point in time prior to a resetlogs operation, which allows administrators more flexibility to detect and correct human errors. Flashback Database is required when using fast-start failover so that Data Guard Broker can automatically reinstate the primary database following an automatic failover.

If you have a standby database, then set DB\_FLASHBACK\_RETENTION\_TARGET to the same value for both primary and standby databases.

**See Also:** *Oracle Database Backup and Recovery Basics* for more information on restore points and Flashback Database

#### 2.2.2.4 Use Fast-Start Fault Recovery to Control Instance Recovery Time

The fast-start fault recovery feature reduces the time required to recover from a crash. It also 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. With this 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 recover time objective (RTO), which is the time (in seconds) that it should take to start up the instance and perform cache recovery. Once this parameter is set, the database manages incremental checkpoint writes in an attempt to meet that 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.

**See Also:** Oracle Database Backup and Recovery Advanced User's *Guide* for more information on fast-start fault recovery

#### 2.2.2.5 Enable Database Block Checking

Enable database block checking by setting DB\_BLOCK\_CHECKING to LOW, MEDIUM, or FULL. The block checking performed for each value is as follows:

LOW

Block checking is performed after any in-memory block change.

MEDIUM

All in-memory block change checking is performed as well as semantic block checking for all non index organized-table blocks.

FULL

Block checking is performed for all in-memory block changes as well as semantic block checking for all non index organized-table blocks and index blocks.

When one of these three forms of block checking is enabled, Oracle Database verifies that the block is self-consistent whenever the corresponding types of block change occur. If the block is inconsistent, then it is marked corrupt, an ORA-1578 error is returned, and a trace file is created containing details of the problem. Without block checking enabled, corruption can go undetected until the block is accessed again. Block checking for the SYSTEM tablespace is always enabled, no matter what setting is chosen for DB\_BLOCK\_CHECKING.

Block checking can often prevent memory and data corruption. Turning on this feature typically causes an additional one percent to ten percent overhead, depending on the setting and the workload. Measure the performance impact on a test system using your workload and ensure that it is acceptable before introducing this feature on an active database.

To check for block corruption on a disk that was not preventable by utilizing DB\_ BLOCK\_CHECKING use one of the following:

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

#### See Also:

- Oracle Database Backup and Recovery Reference for more information about the RMAN BACKUP VALIDATE command
- Oracle Database SQL Reference for more information about the SQL ANALYZE TABLE statement
- Oracle Database Utilities for information on DBVERIFY

#### 2.2.2.6 Set DISK\_ASYNCH\_IO

Set  $DISK_ASYNCH_IO=TRUE$  to enable asynchronous disk I/O for optimal I/O performance.

#### 2.2.2.7 Set LOG\_BUFFER to At Least 8 MB

For large production databases, set the LOG\_BUFFER initialization parameter to a minimum of 8 MB. This setting ensures the database allocates maximum memory (typically 16 MB) for writing Flashback Database logs.

#### 2.2.2.8 Use Automatic Shared Memory Management

Memory management has improved significantly with the advent of Automatic Shared Memory Management (ASM). By setting the SGA\_TARGET parameter to a nonzero value, the shared pool, large pool, Java pool, Streams pool, and buffer cache can automatically and dynamically resize, as needed. See the *Oracle Database Administrator's Guide* for more information.

#### 2.2.2.9 Increase PARALLEL\_EXECUTION\_MESSAGE\_SIZE

Increase initialization parameter PARALLEL\_EXECUTION\_MESSAGE\_SIZE from default value of 2048 to 4096. This configuration step accelerates parallel executions, including instance recovery.

#### 2.2.2.10 Tune PARALLEL\_MIN\_SERVERS

Set PARALLEL\_MIN\_SERVERS so that the required number of parallel recovery processes are pre-spawned for fast recovery from an instance or node crash. This works with FAST\_START\_MTTR\_TARGET to bound recovery time.

PARALLEL\_MIN\_SERVERS = CPU\_COUNT + average number of parallel query processes in use for GV\$ queries and parallel execution

#### 2.2.2.11 Disable Parallel 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 will typically occur with a very aggressive setting of FAST\_START\_ MTTR\_TARGET. In this case, set RECOVERY\_PARALLELISM to 1 to disable parallel recovery.

### 2.2.3 Recommendations to Improve Manageability

This section describes best practices for improving Oracle Database manageability:

- 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

### 2.2.3.1 Use Automatic Performance Tuning Features

Effective data collection and analysis is essential for identifying and correcting performance problems. Oracle provides a number of tools that allow a performance engineer to 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

When using AWR, consider the following best practices:

- 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

### 2.2.3.2 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 Data Guard Broker.

#### See Also:

- Oracle Database Administrator's Guide for information on managing initialization parameters with an SPFILE
- Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for information on initialization parameters with Real Application Clusters
- Oracle Data Guard Broker for information on other prerequisites for using Oracle Data Guard Broker
- Appendix A, "Database SPFILE and Oracle Net Configuration File Samples"

#### 2.2.3.3 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

This parameter should be set to AUTO.

UNDO\_RETENTION

This parameter specifies the desired time in seconds to retain undo data. It must be the same on all instances.

UNDO\_TABLESPACE

This parameter should 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. By default, Oracle Database automatically tunes undo retention by collecting database use statistics and estimating undo capacity needs. You can affect this automatic tuning by setting the UNDO\_RETENTION initialization parameter. It is only necessary to set this initialization parameter in the following cases:

- The undo tablespace has the AUTOEXTEND option enabled.
- You want to have undo retention for LOBs.
- You want a retention guarantee.

**Note:** By default, undo data can be overwritten by ongoing transactions, even if the UNDO\_RETENTION setting specifies that the undo data should be maintained. To guarantee that unexpired undo data is not overwritten, retention guarantee must be enabled for the undo tablespace.

With the retention guarantee option, the undo guarantee is preserved even if there is need for DML activity (DDL statements are still allowed). If the tablespace is configured with less space than the transaction throughput requires, then the following four things will occur in this sequence:

- **1.** If you have an autoextensible file, then it will automatically grow to accommodate the retained undo data.
- 2. A warning alert is issued at 85 percent full.
- **3.** A critical alert is issued at 97 percent full.
- 4. Transactions receive an out-of-space error.

**See Also:** Oracle Database Administrator's Guide for more information about the UNDO\_RETENTION setting and the size of the undo tablespace

#### 2.2.3.4 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 datafile 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 on locally managed tablespaces

#### 2.2.3.5 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 new tablespaces created using default attributes.

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

#### 2.2.3.6 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 altogether. This is a more efficient way of handling temporary segments, from the perspective of both system resource usage and database performance.

A default temporary tablespace should be specified for the entire database to prevent accidental exclusion of the temporary tablespace clause. This can be done at database creation time by using the DEFAULT TEMPORARY TABLESPACE clause of the CREATE DATABASE statement, or after database creation by 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 on managing tablespaces

#### 2.2.3.7 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 need to be restarted. When the space problem is resolved, the suspended operation is automatically resumed.

To use resumable space allocation, set the RESUMABLE\_TIMEOUT initialization parameter to the number of seconds of the retry time. You must also at the session level issue the ALTER SESSION ENABLE RESUMABLE statement.

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

#### 2.2.3.8 Use Database Resource Manager

The Database 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 Database 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 Database 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. The Database Resource Manager is enabled by default.

**See Also:** *Oracle Database Administrator's Guide* for more information on Database Resource Manager

# 2.3 Configuring Oracle Database 10g with RAC

The best practices discussed in this section apply to Oracle Database 10*g* with RAC. These best practices build on the Oracle Database 10*g* configuration best practices described in Section 2.2, "Configuring Oracle Database 10*g*" on page 2-7. These best practices are identical for the primary and standby databases if they are used with Data Guard in Oracle Database 10*g* with RAC and Data Guard - MAA. Some of these best practices might reduce performance levels, but are necessary to reduce or avoid outages. The minimal performance impact is outweighed by the reduced risk of corruption or the performance improvement for recovery.

This section includes the following topics:

- Connect to Database using Services and Virtual Internet Protocol (VIP) Address
- Use Oracle Clusterware to Manage the Cluster and Application Availability
- Use Client-Side and Server-Side Load Balancing
- Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks
- Regularly Back Up OCR to Tape or Offsite
- Verify That CRS and RAC Use Same Interconnect Network
- Configure All Databases for Maximum Instances in the Cluster

### 2.3.1 Connect to Database using Services and Virtual Internet Protocol (VIP) Address

With Oracle Database 10g, application workloads can be defined as services so that they can be individually managed and controlled. 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 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 10g, rules can be defined to automatically allocate processing resources to services. Oracle RAC 10g 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, these rules could be modified at the end of a quarter to ensure that there are enough processing resources to complete critical financial functions on time. Rules can also be defined so that when instances running critical services fail, the workload will be automatically shifted to instances running less critical workloads. Services can be created and administered

with Enterprise Manager, Database Configuration Assistant (DBCA), and the DBMS\_ SERVICE PL/SQL package.

Application connections to the database should be made through a Virtual Internet Protocol (VIP) address to a service defined as part of the workload management facility allowing 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 on which the VIP address can accept connections. Clients that attempt to connect to the VIP address receive a rapid connection refused error instead of waiting for TCP connect timeout messages, thereby reducing the time wasted during the initial connection attempt to a failed node. VIP addresses are configured using the Virtual Internet Protocol Configuration Assistant (VIPCA).

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for more information on workload management

#### 2.3.2 Use Oracle Clusterware to Manage the Cluster and Application Availability

Oracle Clusterware is the only clusterware needed for most platforms on which RAC operates. You can also use clusterware from other vendors if the clusterware is certified for RAC. However, adding unnecessary layers of software for functionality that is already provided by Oracle Clusterware adds complexity and cost and can reduce system availability, especially for planned maintenance.

Oracle Clusterware includes a high-availability framework that provides an infrastructure to manage any application. Oracle Clusterware ensures the applications it manages start when the system starts. Oracle Clusterware also monitors the applications to make sure they are always available. For example, if a process fails, then Oracle Clusterware attempts to restart the process based on scripts that you customize. 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.

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for more information on managing application availability with Oracle Clusterware

### 2.3.3 Use Client-Side and Server-Side Load Balancing

Client-side load balancing evenly spreads new connection requests across all listeners. It is defined in your client connection definition by setting the parameter LOAD\_ BALANCE to ON. (The default is ON for description lists). When this parameter is set to ON, Oracle Database randomly selects an address in the address list and connects to that node's listener. This provides a balancing of client connections across the available listeners in the cluster. When the listener receives the connection request, it connects the user to an instance that it knows provides the requested service. To see what services a listener supports, run the LSNRCTL services command.

Server-side load balancing uses the current workload being run on the available instances for the database service requested during a connection request and directs the connection request to the least loaded instance. Server-side connection load balancing requires each instance to register with all available listeners, which is accomplished by setting LOCAL\_LISTENER and REMOTE\_LISTENER parameters for each instance. This is done by default when creating a database with DBCA.

Connection load balancing can be further enhanced by using the load balancing advisory and defining the connection load balancing goal for each service by setting the GOAL and CLB\_GOAL attributes with the DBMS\_SERVICE PL/SQL package.

#### See Also:

- Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for more information on workload management
- Oracle Database Net Services Administrator's Guide for more information on configuring listeners
- Oracle Database Reference for more information on the LOCAL\_ LISTENER and REMOTE\_LISTENER parameters

### 2.3.4 Mirror Oracle Cluster Registry (OCR) and Configure Multiple Voting Disks

The OCR maintains cluster configuration information as well as configuration information about any cluster database within the cluster. The OCR also manages information about processes that Oracle Clusterware controls. Protect the OCR from disk failure by using the ability of Oracle Clusterware to mirror the OCR. If you have external redundancy, create the OCR on the external redundant storage. If you do not have external redundancy, create a minimum of two OCRs across two different controllers.

RAC uses the voting disk to determine which instances are members of a cluster. The voting disk must reside on a shared disk. For high availability, Oracle recommends that you have multiple voting disks. 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 mirroring to provide redundancy.

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for more information on managing OCR and voting disks

### 2.3.5 Regularly Back Up OCR to Tape or Offsite

The OCR contains cluster and database configuration information for RAC and Cluster Ready Services (CRS), such as the cluster database node list, CRS application resource profiles, and Event Manager (EVM) authorizations. Oracle Clusterware automatically creates OCR backups every four hours. 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 at the end of each week. The backup files created by Oracle Clusterware should be backed up as part of the operating system backup using Oracle Secure Backup, standard operating-system tools, or third-party tools.

**Note:** The default location for generating 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.

In addition to using the automatically created OCR backup files, you should also export the OCR contents before and after making significant configuration changes, such as adding or deleting nodes from your environment, modifying Oracle Clusterware resources, or creating a database. Do this by using the ocrconfig -export command. This exports the OCR content to a file format. The export files created by ocrconfig should be backed up as part of the operating system backup using Oracle Secure backup, standard operating-system tools, or third-party tools.

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for more information on backing up OCR

#### 2.3.6 Verify That CRS and RAC Use Same Interconnect Network

For the most efficient network detection and failover, CRS and RAC should use the same interconnect subnet so that they share the same view of connections and accessibility. To verify the interconnect subnet used by RAC, run the Oracle ORADEBUG utility on one of the instances:

SQL> ORADEBUG SETMYPID Statement processed. SQL> ORADEBUG IPC Information written to trace file. SQL> ORADEBUG tracefile\_name /u01/app/oracle/admin/prod/udump/prod1\_ora\_24409.trc

In the trace file, examine the SSKGXPT section to determine the subnet used by RAC. In this example, the subnet in use is 192.168.0.3 and the protocol used is UDP:

SSKGXPT 0xd7be26c flags info for network 0 socket no 7 IP 192.168.0.3 UDP 14727

To verify the interconnect subnet used by CRS, examine the value of the keyname SYSTEM.css.node\_numbers.node<n>.privatename in OCR:

prompt> ocrdump -stdout -keyname SYSTEM.css.node\_numbers

```
[SYSTEM.css.node_numbers.node1.privatename]
ORATEXT : halinux03ic0
```

•

[SYSTEM.css.node\_numbers.node2.privatename] ORATEXT : halinux04ic0

The hostnames (halinux03ic0 and halinux04ic0 in this example) should match the subnet in the trace file produced by ORADEBUG (subnet 192.168.0.3). Use operating system tools to verify. For example, on Linux:

prompt> getent hosts halinux03ic0
192.168.0.3 halinux03ic0.us.oracle.com halinux03ic0

#### 2.3.7 Configure All Databases for Maximum Instances in the Cluster

During initial setup of a RAC database, the online redo log threads and undo tablespaces for any additional instances in the cluster should be created. If the database might be an Oracle Data Guard standby database at some point, then also create the standby redo logs for each thread at this time.

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for an overview of storage in Oracle Real Application Clusters

## 2.4 Configuring Oracle Database 10g with Data Guard

The best practices discussed in this section apply to Oracle Database 10g with Data Guard. These best practices build on the ones described in Section 2.2, "Configuring Oracle Database 10g" on page 2-7. The proper configuration of Oracle Data Guard Redo Apply and SQL Apply is essential to ensuring that all standby databases work properly and perform their roles within service levels after switchovers and failovers. Most Data Guard configuration settings can be made using Oracle Enterprise Manager. For more advanced, less frequently used Data Guard configuration parameters, the Data Guard Broker command-line interface or SQL\*Plus can be used.

Data Guard enables you to use either a physical standby database (Redo Apply) or a logical standby database (SQL Apply), or both, depending on the business requirements. A physical standby database provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. The database schema, including indexes, are the same. A physical standby database is kept synchronized with the primary database by applying the redo data received from the primary database through media recovery.

A logical standby database contains the same logical information as the production database, although the physical organization and structure of the data can be different. It is kept synchronized with the primary database by transforming the data in the redo log files received from the primary database into SQL statements and then executing the SQL statements on the standby database. A logical standby database can be used for other business purposes in addition to disaster recovery requirements.

This section contains configuration best practices for the following aspects of Data Guard:

- Physical or Logical Standby
  - Benefits of a Physical Standby Database
  - Benefits of a Logical Standby Database
  - Determining Which Standby Type Is Best for Your Application
- Data Protection Mode
- Number of Standby Databases
- General Configuration Best Practices for Data Guard
  - Enable Flashback Database for Easy Reinstantiation After Failover
  - Use FORCE LOGGING Mode
  - Use Data Guard Broker
  - Use a Simple, Robust Archiving Strategy and Configuration
  - Use Standby Redo Logs and Configure Size Appropriately
  - Parameter Configuration Example
- Redo Transport Services Best Practices
  - Conduct Performance Assessment with Proposed Network Configuration
  - Best Practices for Primary Database Throughput
  - Best Practices for Network Configuration and Highest Network Redo Rates
- Log Apply Services Best Practices
  - Redo Apply Best Practices for Physical Standby Databases

- SQL Apply Best Practices for Logical Standby Databases
- Role Transition Best Practices
  - Role Transition During Failover
  - Role Transition During Switchover
- Maintaining a Physical Standby Database as a Clone
- Recommendations on Protecting Data Outside of the Database

#### 2.4.1 Physical or Logical Standby

This section contains information that can help you choose between physical standby and logical standby databases.

This section contains these topics:

- Benefits of a Physical Standby Database
- Benefits of a Logical Standby Database
- Determining Which Standby Type Is Best for Your Application

#### 2.4.1.1 Benefits of a Physical Standby Database

A physical standby database provides the following benefits:

Disaster recovery and high availability

A physical standby database enables a robust and efficient disaster recovery and high-availability solution. Easy-to-manage switchover and failover capabilities allow easy role reversals between primary and physical standby databases, minimizing the downtime of the primary database for planned and unplanned outages.

Data protection

Using a physical standby database, Data Guard can ensure no data loss with certain configurations, even in the face of unforeseen disasters. A physical standby database supports all datatypes, and all DDL and DML operations that the primary database can support. It also provides a safeguard against data corruption and user errors. Storage level physical corruption on the primary database do not propagate to the standby database. Similarly, logical corruption or user errors that cause the primary database to be permanently damaged can be resolved. Finally, the redo data is validated when it is applied to the standby database.

Reduction in primary database workload

Oracle Recovery Manager (RMAN) can use physical standby databases to off-load backups from the primary database saving valuable CPU and I/O cycles. The physical standby database can also be opened in read-only mode for reporting and queries.

Performance

The Redo Apply technology used by the physical standby database applies changes using low-level recovery mechanisms, which bypass all SQL level code layers; therefore, it is the most efficient mechanism for applying high volumes of redo data.

Read-write testing and reporting database

Using Flashback Database and a physical standby database, you can configure a temporary clone database for testing and reporting. The temporary clone can later be resynched with the primary database.

#### 2.4.1.2 Benefits of a Logical Standby Database

A logical standby database provides similar disaster recovery, high availability, and data protection benefits as a physical standby database. It also provides the following specialized benefits:

Efficient use of standby hardware resources

A logical standby database can be used for other business purposes in addition to disaster recovery requirements. It can host additional database schemas beyond the ones that are protected in a Data Guard configuration, and users can perform normal DDL or DML operations on those schemas any time. Because the logical standby tables that are protected by Data Guard can be stored in a different physical layout than on the primary database, additional indexes and materialized views can be created to improve query performance and suit specific business requirements.

Reduction in primary database workload

A logical standby database can remain open at the same time its tables are updated from the primary database, and those tables are simultaneously available for read access. This makes a logical standby database an excellent choice to do queries, summations, and reporting activities, thereby off-loading the primary database from those tasks and saving valuable CPU and I/O cycles.

Database rolling upgrade

A logical standby database can be upgraded to the next version and subsequently become the new primary database after a Data Guard switchover. This rolling upgrade procedure can dramatically reduce the planned downtime of a database upgrade.

#### 2.4.1.3 Determining Which Standby Type Is Best for Your Application

Determining which standby type to implement can be accomplished by examining several key areas. Because logical standby does not support all datatypes, you must first determine if your application uses any unsupported datatypes. To determine if your application uses unsupported datatypes, run the following queries on the primary database:

• To list unsupported tables, issue the following query:

```
SET PAGES 200 LINES 132

COL OWNER FORMAT A8

COL DATA_TYPE FORMAT A15

COL TABLE_NAME FORMAT A32

COL COLUMN_NAME FORMAT A25

COL ATTRIBUTES FORMAT A15

SELECT OWNER, TABLE_NAME, REASON

FROM DBA_STREAMS_UNSUPPORTED

WHERE OWNER NOT IN (SELECT OWNER FROM DBA_LOGSTDBY_SKIP

WHERE STATEMENT_OPT='INTERNAL SCHEMA')

ORDER BY OWNER
```

 To list unsupported tables with column and data type information, issue the following query:

COL OWNER FORMAT A9
```
COL DATA_TYPE FORMAT A35
COL TABLE_NAME FORMAT A35
SELECT OWNER, TABLE_NAME, COLUMN_NAME, DATA_TYPE
FROM DBA_TAB_COLS
WHERE OWNER NOT IN (SELECT OWNER FROM DBA_LOGSTDBY_SKIP
WHERE STATEMENT_OPT='INTERNAL SCHEMA')
AND DATA_TYPE NOT IN ('BINARY_DOUBLE', 'BINARY_FLOAT', 'INTERVAL YEAR TO
MONTH', 'INTERVAL DAY TO SECOND', 'BLOB', 'CLOB', 'CHAR', 'DATE', 'LONG',
'LONG RAW', 'NCHAR', 'NCLOB', 'NUMBER', 'NVARCHAR2', 'RAW', 'TIMESTAMP',
'TIMESTAMP(6)', 'TIMESTAMP(6) WITH TIME ZONE', 'VARCHAR2', 'VARCHAR2')
ORDER BY 1,2
```

If either query returns rows with essential application tables, then use a physical standby database or investigate changing the primary database to use only supported datatypes. If the query does not return any rows with essential application tables, then you can use either a physical or a logical standby database.

Next, consider the need for the standby database to be accessible while changes are being applied. If you require that the standby is opened read/write with read-only access to the data being maintained and your application does not make use of unsupported datatypes, then logical standby is your best choice. If access to the standby while changes are being applied is not required or you have datatypes that are not supported by logical standby, then you should implement a physical standby.

If a logical standby database is still a viable choice, then you need to evaluate if it can handle your peak workloads. Because a logical standby database applies changes with SQL instead of a low level recovery mechanism, you need to assess database performance carefully. See "*Oracle Database 10g Release 2 Best Practices: Data Guard SQL Apply*" at

http://www.oracle.com/technology/deploy/availability/htdocs/maa. htm

# 2.4.2 Data Protection Mode

In some situations, a business cannot afford to lose data at any cost. In other situations, the availability of the database might be more important than protecting data. Some applications require maximum database performance and can tolerate a potential loss of data if a disaster occurs.

Based on your business requirements, choose one of the following protection modes:

- Maximum protection mode guarantees that no data loss will occur if the primary database fails. To ensure that data loss cannot occur, the primary database shuts down if a fault prevents it from writing the redo stream to at least one remote standby redo log.
- **Maximum availability mode** provides the highest level of data protection that is possible without compromising the availability of the primary database.
- Maximum performance mode (the default mode) provides the highest level of data protection that is possible without affecting the performance 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.

The redo data stream of the primary database is also written to at least one standby database, but that redo stream is written asynchronously with respect to the commitment of the transactions that create the redo data. When network links with sufficient bandwidth are used, this mode provides a level of data protection that approaches that of maximum availability mode, with minimal effect on primary database performance.

**See Also:** Oracle Data Guard Concepts and Administration for more information about data protection modes and for information about setting the data protection mode

To determine the correct data protection mode for your application, ask the questions in Table 2–1.

Question	Recommendations			
Is data loss acceptable if	Yes: Use any protection mode.			
the primary site fails?	No: Use maximum protection or maximum availability modes.			
How much data loss is	None: Use maximum protection or maximum availability modes.			
tolerated if a site is lost?	Some: Use maximum performance mode with LGWR ASYNC			
Is potential data loss	Yes: Use maximum performance or maximum availability modes.			
between the production and the standby databases tolerated when a standby host or network connection is temporarily unavailable?	<b>No</b> : Use maximum protection mode <i>, or</i> use maximum availability with multiple standby databases.			
How far away should the disaster-recovery site be from the primary site?	The distance between sites and the network infrastructure between the sites determine the network latency and bandwidth, and therefore the protection mode that can be used. In general, latency increases and bandwidth reduces with distance.			
	For a low-latency, high bandwidth network, use maximum protection or maximum availability. In this case, the performance impact is minimal, and you can achieve zero data loss.			
	For a high-latency network, use maximum performance mode with the ASYNC transport. In this case, the performance impact on the primary is minimal, and you can limit data loss to seconds in most cases. Maximum availability mode and maximum protection modes with the SYNC transport can still be used, but you need to assess if the additional COMMIT latency will exceed your application performance requirements. In some cases, the response time or throughput overhead is zero or within acceptable requirements. Large batch applications or a message queuing applications are good examples where maximum availability with SYNC is still applicable across a high-latency network			
What is the current or proposed network bandwidth and latency between sites?	Bandwidth must be greater than maximum redo generation rate. A guideline for two-way communication is for bandwidth to be 50 percent of the stated network capacity, but you also need to consider the network usage of other applications.			
	Using maximum performance mode with asynchronous redo transport or the archive mitigates the effect on performance.			

 Table 2–1
 Determining the Appropriate Protection Mode

# 2.4.3 Number of Standby Databases

When running in maximum protection mode, consider using multiple standby databases. In maximum protection mode, when the standby host or a network connection is temporarily unavailable, the primary database continues to retry connecting to the standby database for the number of seconds specified by the NET\_TIMEOUT parameter in the LOG\_ARCHIVE\_DEST\_n initialization parameter. The primary database preserves zero data loss during this time period. When it is over, the primary database proceeds with subsequent transactions. By configuring multiple standby databases, the primary database transactions are not interrupted, assuming

that the primary database can communicate with at least one standby database that can satisfy the protection mode requirements.

In many cases, logical standby databases can be used for reporting and data protection and recovery. However, if the logical standby database schema requires additional indices or changes to optimize reporting functions, then it is recommended to have a separate physical standby database to maintain to a consistent copy of the primary database as well.

When you use multiple standby databases, consider hosting each one in a different geographic location so that a network outage or natural disaster does not affect multiple standby databases. For example, host one standby database local to the primary database and another standby database at a remote location.

## 2.4.4 General Configuration Best Practices for Data Guard

This section discusses the following configuration best practices for Data Guard:

- Enable Flashback Database for Easy Reinstantiation After Failover
- Use FORCE LOGGING Mode
- Use Data Guard Broker
- Use a Simple, Robust Archiving Strategy and Configuration
- Use Standby Redo Logs and Configure Size Appropriately
- Parameter Configuration Example

#### 2.4.4.1 Enable Flashback Database for Easy Reinstantiation After Failover

Enable Flashback Database on both the primary and standby database so that the old primary database can be easily reinstated 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.

**See Also:** Section 2.2.2.3, "Enable Flashback Database" on page 2-10 for more information about Flashback Database and for information about enabling Flashback Database

#### 2.4.4.2 Use FORCE LOGGING Mode

When the production database is in FORCE LOGGING mode, all database data changes are logged. FORCE LOGGING mode ensures that the standby database remains consistent with the production database. If this is not possible because you require the load performance with NOLOGGING operations, then you must ensure that the corresponding physical standby datafiles are subsequently synchronized. The physical standby datafiles can be synchronized by either applying an incremental backup created from the primary database or by replacing the affected standby datafiles with a backup of the primary datafiles taken after the nologging operations Before the file transfer, the physical standby database must stop recovery.

For logical standby databases, when SQL Apply encounters a redo record for an operation performed with the NOLOGGING clause, it skips over the record and continues applying changes from later records. Later, if an attempt is made to access one of the records that was updated with NOLOGGING in effect, the following error is returned: ORA-01403 no data found. To recover after the NOLOGGING clause is specified for a logical standby database, re-create one or more tables from the primary database, as described in *Oracle Data Guard Concepts and Administration* in Section 9.4.6 "Adding or Re-Creating Tables On a Logical 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
- Oracle Data Guard Concepts and Administration

#### 2.4.4.3 Use Data Guard Broker

Use Data Guard Broker to create, manage, and monitor a Data Guard configuration. The benefits of using Data Guard Broker include:

Integration with RAC

Data Guard Broker is integrated with CRS so that database role changes occur smoothly and seamlessly. This is especially apparent in the case of a planned role switchover (for example, when a physical standby database is directed to take over the primary role while the former primary database assumes the role of standby). Data Guard Broker and CRS work together to temporarily suspend service availability on the primary database, accomplish the actual role change for both databases during which CRS works with Data Guard Broker to properly restart the instances as necessary, and then resume service availability on the new primary database. Data Guard Broker manages the underlying Data Guard configuration and its database roles while CRS manages service availability that depends upon those roles. Applications that rely on CRS for managing service availability will see only a temporary suspension of service as the role change occurs in the Data Guard configuration.

• Automated creation of a Data Guard configuration

Oracle Enterprise Manager provides a wizard that automates the complex tasks involved in creating a Data Guard Broker configuration, including:

- Adding an existing standby database, or a new standby database created from existing backups taken through Enterprise Manager
- Configuring the standby control file, server parameter file, and datafiles
- Initializing communication with the standby databases
- Creating standby redo log files
- Enabling Flashback Database if you plan to use fast-start failover

Although the Data Guard command-line interface (DGMGRL) cannot automatically create a new standby database, you can use DGMGRL commands to configure and monitor an existing standby database, including those created using Enterprise Manager.

Simplified switchover and failover operations

Data Guard Broker simplifies switchovers and failovers by allowing you to invoke them using a single key click in Oracle Enterprise Manager or a single command at the DGMGRL command-line interface. (referred to in this documentation as manual failover). For lights-out administration, you can enable fast-start failover to allow Data Guard Broker to determine if a failover is necessary and initiate the failover to a pre-specified target standby database automatically, with no need for DBA intervention and with no loss of data. Fast-start failover allows you to increase availability with less need for manual intervention, thereby reducing management costs. Manual failover gives you control over exactly when a failover occurs and to which target standby database. Regardless of the method you choose, Data Guard Broker coordinates the role transition on all databases in the configuration.

Built-in monitoring and alert and control mechanisms

Data Guard Broker provides built-in validation that monitors the health of all databases in the configuration. From any system in the configuration connected to any database, you can capture diagnostic information and detect obvious and subtle problems quickly with centralized monitoring, testing, and performance tools. Both Enterprise Manager and DGMGRL retrieve a complete configuration view of the progress of redo transport services on the primary database and the progress of Redo Apply or SQL Apply on the standby database.

The ability to monitor local and remote databases and respond to events is significantly enhanced by the Data Guard Broker health check mechanism and by its tight integration with the Oracle Enterprise Manager event management system.

#### 2.4.4.4 Use a Simple, Robust Archiving Strategy and Configuration

This archiving strategy is based on the following assumptions:

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

Table 2–2 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 the Data Guard Broker is managing a configuration.

Recommendation	Description		
Archiving must be started on the primary database	Maintaining a standby database requires archiving to be enabled and started on the primary database.		
	SQL> SHUTDOWN IMMEDIATE SQL> STARTUP MOUNT; SQL> ALTER DATABASE ARCHIVELOG; SQL> ALTER DATABASE OPEN;		
Remote archiving must be enabled.	REMOTE_ARCHIVE_ENABLE=TRUE		
Use a consistent log format (LOG_ ARCHIVE_FORMAT).	LOG_ARCHIVE_FORMAT should have the thread, sequence, and resetlogs ID attributes and should be consistent across all instances. %S instructs the format to fill the prefix of the sequence number with leading zeros.		
	If the flash recovery is used, then this format is ignored.		
	For example: LOG_ARCHIVE_FORMAT=arch_%t_%S_%r.arc		
Local archiving is done first by the archiver process (ARCH).	The default setting for LOG_ARCHIVE_LOCAL_FIRST is TRUE, which means that there is a dedicated archiver process archiving redo data to the local destination. Using the flash recovery area implicitly uses LOG_ARCHIVE_DEST_10 for local archiving.		

Table 2–2 Archiving Recommendations

Recommendation	Description		
Remote archiving should be done to only one standby instance and node for each standby RAC database.	All production instances archive to one standby destination, using the same net service name. Oracle Net Services connect-time failover is used if you want 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 ASM or some other shared file system is being used for the flash recovery area, then remote archiving can be spread across the different nodes of a standby RAC database.		
The standby archive destination should use the flash recovery area.	For simplicity, the standby archive destination (STANDBY_ ARCHIVE_DEST) should use the flash recovery area, which is the same as the directory for the local archiving. Because SRLs are present, the standby ARCH process writes to the local archive destination.		
	For a logical standby database, STANDBY_ARCHIVE_DEST cannot use the flash recovery area. Set STANDBY_ARCHIVE_DEST to an explicit archive directory.		
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.		
	<b>See Also:</b> Appendix A, "Database SPFILE and Oracle Net Configuration File Samples"		

Table 2–2 (Cont.) Archiving Recommendations

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 LGWR 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 flash recovery area must be accessible to any node within the cluster and use a shared file system technology such as automatic storage management (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 the node currently running standby apply fails and 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 new managed recovery process (MRP) or logical standby process (LSP) on the different node fetches the archived redo log files using the FAL server to retrieve from the production nodes 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?

#### 2.4.4.5 Use Standby Redo Logs and Configure Size Appropriately

Standby redo logs (SRLs) should be used on both sites for improved availability and performance. Use the following formula to determine the number of SRLs:

# of SRLs = sum of all production online log groups for each thread + number of threads

For example, if a primary database has two instances (threads) and each thread has four online log groups, then there should be ten SRLs. Having one more standby log group for each thread than the number of the online redo log groups for the production database reduces the likelihood that the LGWR for the production instance is blocked because an SRL cannot be allocated on the standby.

The following are additional guidelines for creating SRLs:

- Create the same number of SRLs for both production and standby databases
- All online redo logs and SRLs for both production and standby databases should be the same size
- SRLs should exist on both production and standby databases
- SRLs should be created in the Data Area protected through ASM or external redundancy
- In a RAC environment, SRLs must be on a shared disk
- In a RAC environment, assign the SRL to a thread when the SRL is created

#### 2.4.4.6 Parameter Configuration Example

Appendix A, "Database SPFILE and Oracle Net Configuration File Samples" provides detailed examples of parameter settings, including SPFILE samples and Oracle Net configuration files.

## 2.4.5 Redo Transport Services Best Practices

This section discusses best practices for planning and implementing redo transport services for Data Guard.

This section contains these topics:

- Conduct Performance Assessment with Proposed Network Configuration
- Best Practices for Primary Database Throughput
- Best Practices for Network Configuration and Highest Network Redo Rates

#### 2.4.5.1 Conduct Performance Assessment 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 impact between the production and standby databases, and the impact on the primary database throughput, needs to be understood. Because the network between the production 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

- Minimal latency to reduce the performance impact on the production database
- Multiple network paths for network redundancy

The required bandwidth of a dedicated network connection is determined by the maximum redo rate of the production database and actual network efficiency. Depending on the data protection mode, there are other recommended practices and performance considerations. Maximum protection mode and maximum availability mode require LGWR SYNC transport. Maximum performance protection mode uses the ASYNC transport option or the archiver (ARCHn).

When you compare maximum protection mode or maximum availability mode (with LGWR SYNC operations) with maximum performance mode (with LGWR ASYNC operations), measure whether performance or throughput will be degraded due to the incurred latency. You should also check whether the new throughput and response time are within your application performance requirements. Distance and network configuration directly influence latency, while high latency might slow down your potential transaction throughput and increase response time. The network configuration, number of repeaters, the overhead of protocol conversions, and the number of routers also affect the overall network latency and transaction response time.

See Also: Section 2.4.2, "Data Protection Mode" on page 2-23

#### 2.4.5.2 Best Practices for Primary Database Throughput

When sending redo to a standby database using the LGWR SYNC attributes, a transaction on the primary database will not return commit complete to the foreground until the redo associated with that transaction has been written both locally and remotely. The commit time when using LGWR SYNC will be directly impacted by the network latency and bandwidth, as well as the I/O capacity on the standby database. The total commit time is comprised of the primary database's local write (log file parallel write) and the following factors that are captured through the LNS wait event on SENDREQ: network time + standby write (RFS write obtained from the V\$SYSTEM\_EVENT view on the standby database) + network acknowledgement. However, how much the primary database is impacted depends on the application profile. In general, batch updates with infrequent commits and message queuing applications may not observe any noticeable difference.

When sending redo to a standby database using the LGWR ASYNC attributes, the effect on primary database throughput is minimal due to the true asynchronous behavior of ASYNC redo transport. Furthermore, there is little effect on the primary database throughput (redo bytes per second) as network latency increases. With the ASYNC attribute, the log writer process writes to the local online redo log file, while the LGWR network server (LNS*n*) processes (one for each destination) read redo from the online redo log and asynchronously transmit the redo to remote destinations. The LGWR process continues processing the requests without waiting for the LNS network I/O to complete. If redo transport services transmit redo data to multiple remote destinations, then the LNS*n* processes initiate the network I/O to all of the destinations in parallel.

Figure 2–3 shows the LNS*n* process collecting redo data from the online redo log files and transmitting it over Oracle Net to the RFS process on the standby database.



Figure 2–3 LGWR ASYNC Archival with Network Server (LNSn) Processes

See Also: Oracle Data Guard Concepts and Administration

For remote destinations that are serviced by the ARCH process, you can configure the remote transfer to use multiple streams for more efficient use of network resources. Configuration of multiple streams is performed by setting the MAX\_CONNECTIONS attribute in the LOG\_ARCHIVE\_DEST\_n initialization parameter to a value greater than 1. The value determines the maximum number of network connections that will be used to perform the remote archival. The maximum value is five streams for each remote destination.

Because the network connections used in the multiple streams are performed by the ARCH process, care must be taken when setting the LOG\_ARCHIVE\_MAX\_PROCESSES initialization parameter. The value of both LOG\_ARCHIVE\_MAX\_PROCESSES and PARALLEL\_MAX\_SERVERS initialization parameters must be at least one greater than the total number specified for MAX\_CONNECTIONS for all remote destinations.

#### 2.4.5.3 Best Practices for Network Configuration and Highest Network Redo Rates

The following sections include best practices for network configuration and highest redo network redo rates:

- Properly Configure TCP Send / Receive Buffer Sizes
- Increase SDU Size
- Ensure TCP.NODELAY is YES
- Increase PARALLEL\_MAX\_SERVERS

#### Account for Additional Disk I/O with LGWR ASYNC

**2.4.5.3.1 Properly Configure TCP Send / Receive Buffer Sizes** Set the TCP send and receive socket buffer size to twice the bandwidth delay product (BDP) of the network link between the primary and standby systems. BDP is product of the network bandwidth and latency. Socket buffer sizes should be 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.

For example, if bandwidth is 622 Mbits and latency =30 ms, then you should set RECV\_BUF\_SIZE and SEND\_BUF\_SIZE as follows:

RECV\_BUF\_SIZE= SEND\_BUF\_SIZE= 2 x 622,000,000 / 8 x 0.030 = 4,665,000 bytes

**See Also:** Primary Site and Network Configuration Best Practices at http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

**2.4.5.3.2 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). Setting SDU to its maximum value of 32767 can improve performance. SDU designates the size of the Oracle Net buffer used to collect data before it is delivered to the TCP network layer for transmission across the network. Oracle internal testing of Oracle Data Guard has demonstrated that the maximum setting of 32767 performs best. The gain in performance is a result of the reduced number of system calls required to pass the data from Oracle Net buffers to the operating system TCP network layer. SDU can be set on a per connection basis with the SDU parameter in the local naming configuration file (tnsnames.ora) and the listener configuration file (listener.ora), or SDU can be set for all Oracle Net connections with the profile parameter DEFAULT\_SDU\_SIZE in the sqlnet.ora file.

#### See Also:

- Oracle Database Net Services Reference for more information on the SDU and DEFAULT\_SDU\_SIZE parameters
- Primary Site and Network Configuration Best Practices at http://otn.oracle.com/deploy/availability/htdoc s/maa.htm

**2.4.5.3.3 Ensure TCP.NODELAY is 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

**2.4.5.3.4 Increase PARALLEL\_MAX\_SERVERS** Increase the PARALLEL\_MAX\_SERVERS initialization parameter to accommodate the MAX\_CONNECTIONS attribute setting for the LOG\_ARCHIVE\_DEST\_n initialization parameter. The LOG\_ARCHIVE\_MAX\_PROCESSES and PARALLEL\_MAX\_SERVERS initialization parameters are related to the MAX\_CONNECTIONS attribute and affect the actual number of ARCn processes used by an instance. For example, if the total number of connections specified by the MAX\_CONNECTIONS attributes on all destinations exceeds the value of LOG\_ARCHIVE\_MAX\_

PROCESSES, then Data Guard will use as many ARCn processes as possible, but the number of connections may be fewer than what is specified by MAX\_CONNECTIONS.

**2.4.5.3.5** Account for Additional Disk I/O with LGWR ASYNC Allow for sufficient I/O bandwidth to account for additional read I/O operations caused by enabling LGWR ASYNC redo transmission. When the LGWR and ASYNC attributes are specified on the LOG\_ARCHIVE\_DEST\_n initialization parameter, the log writer process writes to the local online redo log file with no additional writes to the ASYNC buffer as in Oracle Database 10g Release 1 (10.1). In Oracle Database 10g Release 2 (10.2), after LGWR has completed writing to the online log, the LNSn process reads the change vectors from the online redo log and ships the redo to the standby database. With this new approach, the LGWR process writes are completely decoupled from LNSn network writes. Because the LNS process is performing a disk read against the online redo log, when it was not in previous releases, care should be taken that sufficient disk I/O bandwidth exist.

## 2.4.6 Log Apply Services Best Practices

This section discusses the best practices for Data Guard log apply services for both physical and logical standby databases.

This section contains these topics:

- Redo Apply Best Practices for Physical Standby Databases
- SQL Apply Best Practices for Logical Standby Databases

#### 2.4.6.1 Redo Apply Best Practices for Physical Standby Databases

To use Oracle Data Guard Redo Apply with a physical standby database, or to use any media recovery operation effectively, you need to tune your database recovery by following these best practices:

1. 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 will always be bounded by the rate at which redo can be read; so ensure that the redo read rate surpasses your required recovery rate.

**2.** 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 Grid Control 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.

3. Use defaults for DB\_BLOCK\_CHECKING and DB\_BLOCK\_CHECKSUM

The default setting for DB\_BLOCK\_CHECKING is OFF. Setting DB\_BLOCK\_ CHECKING to FULL may reduce the recovery rate. Block checking is always recommended on the primary database and might be enabled on the standby database if the recovery rate meets expectations. The default setting for DB\_BLOCK\_CHECKSUM is TYPICAL. Block checksum should always be enabled for both primary and standby databases. It catches most block corruption while incurring negligible overhead.

4. Set parallel\_execution\_message\_size = 4096

Increasing this parameter to 4096 may improve recovery by as much as twenty percent over the default setting of 2152. The message size parameter is used by parallel query operations, so there must be sufficient shared pool to support this increase.

**5.** 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.

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

6. Assess database waits

You can determine the top system and session wait events by querying the standby database's V\$SYSTEM\_EVENTS, V\$SESSION\_WAITS, 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. Unfortunately, there is no equivalent AWR report for physical standby databases.

If recovery is applying a lot of redo efficiently, the system will be I/O bound and the I/O wait should be reasonable for your system. The following table shows the top recovery-related waits that you may observe and the tuning tips appropriate for each wait type. Apply the tuning tips only if the recovery events are in the top ten waits.

Wait Name	Description	Tuning Tips
log file sequential read	Coordinator (recovery session or MRP process) wait for log file read I/O	Tune log read I/O
px deq: par recov reply	Coordinator synchronous wait for slave (wait for checkpoints)	Tune datafile write I/O, increase DBWR processes, increase primary/standby redo log size
px deq credit: send blkd	Coordinator streaming wait for slave (wait for apply)	Increase PARALLEL_EXECUTION_ MESSAGE_SIZE to 8192
free buffer waits	Foreground waiting available free buffer in the buffer cache	Increase DB_CACHE_SIZE and remove any KEEP or RECYCLE POOL settings
recovery read	Wait for data block read	Tune datafile read I/O
direct path read	Coordinator wait for file header read at log boundary checkpoint	Tune datafile read I/O
direct path write	Coordinator wait for file header write at log boundary checkpoint	Tune datafile write I/O

Wait Name	Description	Tuning Tips
checkpoint completed (serial recovery only)	Wait for checkpoint completed	Tune datafile write I/O Increase number of DBWR processes
db file parallel read (serial recovery only)	Wait for data block read	Tune file read I/O

#### **7.** 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).

**8.** Assess system resources.

Use system commands such as UNIX sar and vmstat or system monitoring tools to assess system resources. Alternatively, refer to Enterprise Manager, AWR reports or performance views such as V\$SYSTEM\_EVENT, V\$ASM\_DISK and V\$OSSTAT.

- **a.** 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 ASM disk group. Verify that this is not a bus or controller bottleneck or any other I/O bottleneck. The read I/O from the standby redo log should be greater than the expected recovery rate.
- **b.** Check for excessive swapping or memory paging.
- **c.** Check to ensure the recovery coordinator or MRP is not CPU bound during recovery.
- **9.** Increase log group size for the primary and standby databases.

Increase the online redo log size for the primary database and the standby redo logs size for the standby database to a minimum of 1 GB. Oracle Database does a full checkpoint and updates all the file headers (in an optimized manner) at each log file boundary during media recovery. To reduce the frequency of a full database checkpoint and updating all the file headers, increase the log group size so that a log switch is occurring at a minimum of 15 minute intervals.

If real-time apply is being used and redo is being sent synchronously or asynchronously by way of the LGWR process, then there is no additional data loss risk with this change. If archiver is sending the redo or the primary database is converting to ARCH mode due to heavy load, then you must balance faster recovery rates and higher data loss risk.

To ensure that the crash recovery time for the primary database is minimized even with very large redo group sizes, set FAST\_START\_MTTR\_TARGET to a nonzero

value to enable fast-start fault recovery. If it is currently not set, then set it to 3600. This initialization parameter is relevant only for the primary database.

10. Assess different degrees of recovery parallelism

Parallel recovery is enabled by default for media and crash recovery with the default degree of parallelism set to the number of CPUs available. In most cases this is the optimal setting. However, in some circumstances faster recovery may be obtained by using a degree of parallelism that is different (higher or lower) than the default. To override the default setting, explicitly specify it as follows:

RECOVER MANAGED STANDBY DATABASE PARALLEL <#>;

See Also: Oracle Database 10g Release 2 Best Practices: Data Guard Redo Apply and Media Recovery at http://www.oracle.com/technology/deploy/availability /htdocs/maa.htm

#### 2.4.6.2 SQL Apply Best Practices for Logical Standby Databases

This section discusses recommendations for Data Guard SQL Apply and logical standby databases.

This section contains these topics:

- Set the MAX\_SERVERS Initialization Parameter
- Increase the PARALLEL\_MAX\_SERVERS Initialization Parameter
- Set the PRESERVE\_COMMIT\_ORDER Initialization Parameter
- Skip SQL Apply for Unnecessary Objects
- Set the LOG\_AUTO\_DELETE SQL Apply Parameter

**2.4.6.2.1** Set the MAX\_SERVERS Initialization Parameter If the logical standby database is being used as a reporting or decision-support system, then increase the value of the MAX\_SERVERS initialization parameter to reserve parallel query slaves for such operations. Because the SQL Apply process by default uses all of the parallel query slaves, set the MAX\_SERVERS parameter as shown in the following equation to enable a specific number of parallel query slaves to be reserved:

 $\label{eq:MAX_SERVERS} = < current \ \mbox{MAX}\ \mbox{SERVERS setting} > + < \mbox{PQ Slaves needed for reporting and} \\ decision-support operations>$ 

Oracle recommends that you initially set the MAX\_SERVERS parameter to either 9 or 3 + (3 x CPU), whichever value is larger.

**2.4.6.2.2** Increase the PARALLEL\_MAX\_SERVERS Initialization Parameter If the current settings has not been adjusted for MAX\_SERVER processes, increase the value of the PARALLEL\_MAX\_SERVERS initialization parameter by the value of the MAX\_SERVERS initialization parameter on both the primary and standby database instances:

PARALLEL\_MAX\_SERVERS = current value + MAX\_SERVERS value

**2.4.6.2.3** Set the PRESERVE\_COMMIT\_ORDER Initialization Parameter For a reporting or decision-support system, PRESERVE\_COMMIT\_ORDER should be set to TRUE except when the standby database has fallen behind the primary database. Temporarily set PRESERVE\_COMMIT\_ORDER to FALSE while SQL Apply is getting the standby database caught up to the primary database, but reset the parameter to TRUE after the gap has been resolved.

**2.4.6.2.4** Skip SQL Apply for Unnecessary Objects Database objects that do not require replication to the standby database should be skipped by using the DBMS\_LOGSTDBY.SKIP procedure. Skipping such objects reduces the processing of SQL Apply.

**2.4.6.2.5** Set the LOG\_AUTO\_DELETE SQL Apply Parameter Set the LOG\_AUTO\_DELETE SQL Apply parameter by running the DBMS\_LOGSTDBY.APPLY\_SET procedure. The LOG\_AUTO\_DELETE parameter controls whether SQL Apply automatically deletes archived redo log files sent from the primary database once they have been applied on the logical standby database. Set this parameter to TRUE to enable automatic deletion of archived redo log files. Set it to FALSE to disable automatic deletion. The default value is TRUE.

If the LOG\_AUTO\_DELETE parameter is set to FALSE, then you can use the DBMS\_LOGSTDBY.PURGE\_SESSION procedure to delete archived redo log files manually.

**See Also:** Oracle Database PL/SQL Packages and Types Reference and Oracle Data Guard Concepts and Administration for more information on the DBMS\_LOGSTDBY PL/SQL package

# 2.4.7 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. Whether using physical standby or logical standby databases, MAA testing has determined that switchover and failover times with Oracle Data Guard 10*g* release 2 have been reduced to seconds. This section describes best practices for both switchover and failover.

This section contains these topics:

- Role Transition During Switchover
- Role Transition During Failover

#### 2.4.7.1 Role Transition During Switchover

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 production database. Following a successful switchover operation, the standby database assumes the production role and the production database becomes a standby database. In a RAC environment, a switchover requires that only one instance be active for each database, production and standby.

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.

Data Guard enables you to change these roles dynamically by:

- Using Oracle Enterprise Manager, as described in Section 4.2.3.2.1, "Using Enterprise Manager to Perform a Data Guard Switchover" on page 4-20
- Using the Oracle Data Guard Broker command-line interface
- Issuing SQL statements, as described in Section 4.2.3.2.2, "Using SQL for Data Guard Switchover to a Physical Standby Database" and Section 4.2.3.2.3, "Using SQL for Data Guard Switchover to a Logical Standby Database" beginning on page 4-22

**See Also:** Oracle Data Guard Broker for information about using Enterprise Manager or the Data Guard broker command-line to perform database switchover

**2.4.7.1.1 Switchover Best Practices** To optimize switchover processing, use the following best practices and see the *Oracle Database 10g Release 2 Best Practices: Data Guard Switchover and Failover* white paper available at

http://www.oracle.com/technology/deploy/availability/pdf/MAA\_WP\_
10gR2\_SwitchoverFailoverBestPractices.pdf:

- For logical standby databases:
  - See Oracle Database 10g Release 2 Best Practices: Data Guard SQL Apply at http://www.oracle.com/technology/deploy/availability/htdoc s/maa.htm to obtain an optimal SQL Apply rate.
  - Verify the LogMiner Multiversioned Data Dictionary was received by the primary database by querying the SWITCHOVER\_STATUS column of the V\$DATABASE fixed view on the primary database. When the query returns the TO LOGICAL STANDBY value, you can proceed with the switchover. See the discussion about "Switchovers Involving a Logical Standby Database" in *Oracle Data Guard Concepts and Administration*
- For physical standby databases:
  - See Oracle Database 10g Release 1 Best Practices: Data Guard Redo Apply and Media Recovery at http://www.oracle.com/technology/deploy/availability/htdoc s/maa.htm to obtain an optimal Redo Apply rate.
  - When transitioning from read-only mode to Redo Apply (recovery) mode, restart the database.
- Use real-time apply so that redo data is applied to the standby database as soon as it is received.

To enable real-time apply for a physical standby database, use the following SQL statement:

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT USING CURRENT LOGFILE;

To enable real-time apply for a logical standby database, use the following SQL statement:

ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;

• Enable Flashback Database so that if a failure occurs during the switchover, the process can be easily reversed.

#### 2.4.7.2 Role Transition During Failover

Failover is the operation of taking the production database offline at one site and bringing one of the standby databases online as the new production database. A failover operation can be invoked when a catastrophic failure occurs on the production database, and there is no possibility of recovering the production database in a timely manner.

With Data Guard the process of failover can be completely automated using fast-start failover, or it can be user driven. Fast-start failover eliminates the uncertainty inherent

in a process that requires manual intervention. It automatically executes a zero data loss failover within seconds of an outage being detected.

Oracle recommends that you use fast-start failover. The initial MAA tests running Oracle Database 10g Release 2 (10.2) show that failovers performed using the Data Guard Broker and fast-start failover offer a significant improvement in availability.

See Also: Oracle Database 10g Release 2 Best Practices: Data Guard Fast-Start Failover at http://www.oracle.com/technology/deploy/availability /htdocs/maa.htm for a comprehensive review of Oracle failover best practices

Manual failover allows for a failover process where decisions are user driven. Manual failover can be accomplished by:

- Issuing SQL statements, as described in Section 4.2.2.2.2, "Using SQL to Fail Over to a Physical Standby Database" on page 4-18
- Using Oracle Enterprise Manager, as described in Section 4.2.2.2.1, "Using Enterprise Manager to Perform a Data Guard Failover" on page 4-15
- Using the Oracle Data Guard Broker command-line interface (DGMGRL)

**See Also:** Oracle Data Guard Broker for information about using Enterprise Manager or the Data Guard broker command-line for database failover

This section contains these topics:

- Comparing Fast-Start Failover and Manual Failover
- Fast-Start Failover Best Practices
- Manual Failover Best Practices

**2.4.7.2.1 Comparing Fast-Start Failover and Manual Failover** Fast-start failover can be used only in a Data Guard Broker configuration. It can be configured only through DGMGRL or the Data Guard management pages in Oracle Enterprise Manager. Fast-start failover also requires that the redo transport services be configured for LGWR SYNC, and the Data Guard configuration should be in maximum availability mode to achieve the stated guarantee of zero data loss failover. In addition both the primary and standby databases must have Flashback Database enabled. When enabled, fast-start failover monitors the Data Guard configuration and initiates a failover automatically to the specified target standby database automatically, with no need for DBA intervention and with no loss of data.

The following conditions will trigger a fast-start failover:

- Database instance failure (or last instance failure in a RAC configuration)
- Shutdown abort (or shutdown abort of the last instance in a RAC configuration)
- Datafiles taken offline due to I/O errors
- Both the observer (fast-start failover monitoring process) and the standby database lose their network connection to the primary database, and the standby database confirms that it is in a synchronized state.

Following a fast-start failover, the old primary database is automatically reconfigured as a new standby database upon reconnection to the configuration. This enables Data Guard to restore disaster protection in the configuration quickly and easily, returning the database to a protected state as soon as possible.

A Data Guard manual failover is a series of steps to convert a standby database into a production database. The standby database essentially assumes the role of production database. A Data Guard failover is accompanied by an application failover to fail over the users to the new primary database. After the failover, the former production database must be re-created as a new standby database to restore resiliency. The standby database can be quickly re-created by using Flashback Database. See Section 4.3.2, "Restoring a Standby Database After a Failover" on page 4-50.

**See Also:** Oracle Data Guard Broker for more information on failover operations

**2.4.7.2.2 Fast-Start Failover Best Practices** The following are fast-start failover best practices:

- Use real-time apply so that redo data is applied to the standby database as soon as it is received.
- Enable Flashback Database to protect from logical failures.
- 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, the observer should be run on a system that is equally distant from primary and standby databases. It should connect to the primary and standby databases using the same network as any end-user client. If the designated observer fails, Enterprise Manager can detect it and can be configured to automatically restart the observer. If unable to run at a third site the observer should be installed on the same network as the application.

- If the standby database has been opened in read-only mode, then restart the database before starting Redo Apply.
- Consider configuring multiple standby databases to maintain data protection following a failover.
- Set the value of the FastStartFailoverThreshold property according to your configuration characteristics, as described in Table 2–3.

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
RAC primary	Reconfiguration time + 30 seconds <sup>1</sup>

Table 2–3 Minimum Recommended Settings for FastStartFailoverThreshold

<sup>1</sup> For configurations running Oracle Database software prior to Release 10.2.0.3, use a minimum of RAC miscount + reconfiguration time + 30 seconds

For any of the settings shown in Table 2–3, perform testing to ensure that the fast-start failover threshold is not so aggressive that it will induce false failovers, or so high it does not meet your failover requirements.

#### See Also:

- Oracle Data Guard Broker for the FastStartFailoverThreshold
   property
- Oracle Database 10g Release 2 Best Practices: Data Guard Switchover and Failover at http://www.oracle.com/technology/deploy/availabil ity/htdocs/maa.htm

**2.4.7.2.3** Manual Failover Best Practices A manual failover, which is user-driven, should be used only in case of an emergency and 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 impacts the production application

A failover requires that the initial production database must be reinstated as a standby database to restore fault tolerance to your environment. The standby database can be quickly reinstated by using Flashback Database. See Section 4.3.2, "Restoring a Standby Database After a Failover" on page 4-50.

To optimize failover processing and to maintain high availability, use the following best practices:

- Enable Flashback Database to reinstate databases after the failover operation has completed.
- Use real-time apply in conjunction 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.
- For logical standby databases, see Oracle Database 10g Release 2 Best Practices: Data Guard SQL Apply at http://www.oracle.com/technology/deploy/availability/htdocs/m aa.htm to obtain an optimal SQL Apply rate.
- For physical standby databases:
  - See Oracle Database 10g Release 1 Best Practices: Data Guard Redo Apply and Media Recovery at http://www.oracle.com/technology/deploy/availability/htdoc s/maa.htm.
  - Go directly to the OPEN state from the MOUNTED state instead of restarting the standby database (as required in previous releases).
- When transitioning from read-only mode to Redo Apply (recovery) mode, restart the database.

#### 2.4.8 Maintaining a Physical Standby Database as a Clone

Data Guard, Restore Points, and Recovery Manager can be combined to enable a Redo Apply (physical) standby database to be used as a read/write clone database temporarily for testing, reporting, development, or any other use clone technology might have within your organization. The following are recommendations to consider when using a standby database as a clone:

• Evaluate the performance of this approach

When evaluating using this method for maintaining a clone database, measure the time needed to refresh the clone database, given the degree that the clone database will diverge from the primary database. A clone database that is used heavily and diverges significantly from the primary database (more than 15% of the total data blocks are changed between database) might be synchronized more quickly by simply running RMAN DUPLICATE to copy the entire database from a disk backup.

Also, ensure there are sufficient I/O resources available. The process of flashing back the changes made to the clone database and synchronizing it with the primary by applying an incremental backup is I/O intensive.

Use a separate standby database for disaster protection.

While the standby database is activated as the clone, it is not receiving redo data from the primary database and cannot provide disaster protection. To provide disaster protection while a physical standby database is activated, there should be multiple standby databases running in the Data Guard configuration so that the primary database remains protected while one of the standby databases is activated. Using a second standby database allows the Data Guard configuration to remain in maximum protection or maximum availability mode while the clone database is activated.

• Enable block change tracking on the primary database.

The primary database INCREMENTAL FROM SCN backup will be created more quickly with block change tracking enabled. This backup is used to apply all changes that occurred on the primary database to the clone database while it was diverged. See *Oracle Database Backup and Recovery Basics* for information about enabling block change tracking.

Use RMAN parallelism for creating and applying the incremental backup.

When creating and applying the primary database INCREMENTAL FROM SCN backup, use RMAN parallelism to reduce the create and apply time. See *Oracle Database Backup and Recovery Advanced User's Guide* for more information about RMAN parallelism.

• Create the primary incremental backup and the flashback clone simultaneously.

To reduce the overall synchronization time, create the incremental backup from the primary database at the same time the clone database is being flashed back to the initial restore point.

To use a standby database as a clone, locate a physical standby database that is not being used for disaster protection. The following sections list the general phases and steps but you can find detailed information about using a standby database as a clone in *Oracle Data Guard Concepts and Administration*:.

#### Phase 1: Activate the physical standby database as a clone

- 1. Prepare the physical standby database to be activated.
- 2. Prepare the primary database to have the physical standby be diverged.
- 3. Activate the physical standby database (diverge it from the primary database).
- **4.** Use the activated clone database for testing.

#### Phase 2: Resynchronize the clone to become a physical standby database

- **5.** Flashback the activated database back to a physical standby database.
- 6. Catch up the standby database to the primary database.

#### 2.4.9 Recommendations on Protecting Data Outside of the Database

In a high-availability environment, nondatabase files must be protected along with database files. Oracle Secure Backup provides data protection for heterogeneous UNIX, Linux, Windows, and Network Attached Storage environments. Additionally, for disaster recovery purposes, some third-party tools enable remote synchronization between a set of local and remote files. For example, you can use tools such as rsync, csync2, and DRDB for remote synchronization. These tools are available for download on the internet. The following are recommendations regarding these tools:

- For software updates, use rsync to synchronize the standby system with the changes made to software on the primary system.
- For configuration files, use rsync daily or after a change, or use csync2.
- For important log files, trace files, or debugging files, use rsync daily or hourly, or use DRDB to synchronize the entire file system.
- For transaction logs or metadata files that must be synchronized with the database, use rsync or csync2 frequently, or use a block synchronization tool such as DRDB, a third party mirroring utility, or remote synchronization tool.

**See Also:** Oracle Secure Backup Administrator's Guide

#### 2.4.10 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 Data Guard with the same application profile and load.

Application profile can be quickly assessed by comparing the following statistics:

- Physical reads per transaction
- Physical writes per transaction
- CPU usage per transaction
- Redo generated per transaction

Application performance can be quickly assessed by comparing 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 see there is an application performance on the primary 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 utilization of over 90%, paging and swapping), then you need to tune the system before proceeding with Data Guard. Use the V\$OSSTAT or V\$SYSMETRIC\_HISTORY view to monitor system usage statistics from the operating system.

Higher I/O waits

If you are experiencing higher I/O waits from LGWR and DBWR, then the slower I/O will impact throughput and response time. This can be observed by looking at 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 LGWR SYNC transport, commits will take more time because we need to guarantee that the redo is available on the standby database before foregrounds get an acknowledgement from LGWR that the commit has completed. A LGWR commit includes the following wait events:

- Log file parallel write (local write for LGWR)
- LGWR wait on SENDREQ

This wait event includes:

- Time to put the packet into the network
- Time to send the packet to the standby database
- RFS write or standby write to the standby redo log, which includes the RFS I/O wait event plus additional overhead for checksums
- Time to send a network acknowledgement back to the primary database (for example, single trip latency time)

Higher LGWR (log writer) commit times can cause longer response time and lower throughput especially for small time-sensitive transactions. However, you may obtain sufficient gains by tuning the LGWR local write (log file parallel write wait event) or the different components that make up the LGWR wait on SENDREQ wait.

The disk write I/O (log file parallel write or the RFS I/O) can be tuned by adding more spindles or increasing the I/O bandwidth. The network time can be reduced by tuning the Oracle Net send and receive buffer sizes, setting SDU=32K, increasing the network bandwidth if there is saturation, and possibly finding a closer site to reduce the network latency.

With LNS*n* ASYNC transport and ARCH transport, LGWR never waits for LNS*n* or ARCH before writing its commit record to the current log file; however, both LNS*n* and archiver processes read the online redo logs; thus causing more I/O contention and possibly longer wait times for LGWR writes (log file parallel writes). If I/O bandwidth and sufficient spindles are not allocated, then you will see higher log file parallel

writes and log file sequential reads, which may have an effect on throughput and response time. In most cases, adding sufficient spindles will reduce the I/O latency.

**Note:** To enable most of the new 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
- The "Oracle 10g Data Guard: Primary Site and Network Configuration Best Practices" or tuning LGWR SYNC and LGWR ASYNC transport components. This white paper can be downloaded from http://www.oracle.com/technology/deploy/availabil ity/htdocs/maa.htm

# 2.5 Configuring Backup and Recovery

While it is prudent that every database has a good backup, consider your Recover Time Objective (RTO) and Recovery Point Objective (RPO) when designing a backup and recovery strategy. While many recoveries involve restoring a backup, Oracle provides other database features such as Data Guard and Flashback Technology to minimize the recovery time from a database outage.

This section discusses the best practices for maintaining a good database backup, as well as other backup options and strategies made possible by the available Oracle database features.

This section contains the following topics:

- Use Oracle Database Features and Products
  - Use Recovery Manager to Back Up Database Files
  - Use Oracle Secure Backup
  - Use Restore Points
- Configuration and Administration
  - Understand When to Use Backups
  - Determine a Backup Frequency
  - Use an RMAN Recovery Catalog
  - Enable Block Change Tracking for Incremental Backups
  - Enable Autobackup for Control File and Server Parameter File
- Backup to Disk
  - Determine Disk Backup Methods
  - Create Backups in NOCATALOG Mode and RESYNC CATALOG Afterwards
  - Create Database Backups on Disk in the Flash Recovery Area
  - In a Data Guard Environment, Back Up to Flash Recovery Area on All Sites
- Backup to Tape

- Create Tape Backups from the Flash Recovery Area
- Maintain Offsite Backups
- Backup and Recovery Maintenance
  - Regularly Check Database Files for Corruption
  - Periodically Test Recovery Procedures
  - Regularly Backup the Recovery Catalog Database

# 2.5.1 Use Oracle Database Features and Products

Oracle has multiple database features and products to facilitate Backup and Recovery operations, including Recovery Manager (RMAN), Oracle Secure Backup, the flash recovery area, Flashback Database and restore points.

This section contains these topics:

- Use Recovery Manager to Back Up Database Files
- Use Oracle Secure Backup
- Use Restore Points

## 2.5.1.1 Use Recovery Manager to Back Up 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 need to be backed up. But more importantly, RMAN knows what files need to 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
- Incremental backups
- Data block integrity checks during backup and restore operations
- Test backups and restores without actually performing the operation

RMAN automates backup and recovery. User-managed methods require you to locate backups for each datafile, copy them to the correct place using operating system commands, and choose which logs to apply. RMAN manages these tasks automatically.

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.

## 2.5.1.2 Use Oracle Secure Backup

Oracle Secure Backup provides data protection for heterogeneous UNIX, Linux, Windows and Network Attached Storage (NAS) environments. Oracle Secure Backup provides tape data protection for the entire Oracle Environment:

- Oracle Database through integration with RMAN
- Seamless support of Oracle Real Application Clusters (RAC)
- File system data protection of distributed servers including:
  - Oracle Application Servers

- Oracle Collaboration Suites
- Oracle home and binaries

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

**See Also:** The Oracle Secure Backup web site at http://www.oracle.com/database/secure-backup.html

#### 2.5.1.3 Use Restore Points

Oracle restore points can be used to protect against logical failures at risky conjunctions during database maintenance. Creating a normal restore point assigns a restore point name to a specific point in time or SCN. The restore point name can be used with Flashback Table and Flashback Database operations. Restore points can be guaranteed to ensure that a Flashback Database operation will succeed in rewinding the database back to the restore point.

Guaranteed restore points are recommended for database-wide maintenance such as database or application upgrades, or running batch processes. Guaranteed restore points enable Flashback Database and retain all flashback logs necessary to ensure the database can be flashed back to the restore point. Once maintenance activities complete and results are verified, guaranteed restore points that are no longer needed should be deleted.

**See Also:** Oracle Database Backup and Recovery Basics for more information on Flashback Database

# 2.5.2 Configuration and Administration

This section contains these topics:

- Understand When to Use Backups
- Determine a Backup Frequency
- Use an RMAN Recovery Catalog
- Enable Block Change Tracking for Incremental Backups
- Enable Autobackup for Control File and Server Parameter File

#### 2.5.2.1 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) or SLA. For example, some outages are handled best by using Flashback Database or the standby database. However, some situations require using database backups, including the following:

- Initial Data Guard Environment Setup
- Recovering from Data Failures Using File or Block Media Recovery
- Double Failure Resolution

#### See Also:

- Oracle Data Guard Concepts and Administration for more information on Data Guard setup
- Oracle Database Backup and Recovery Basics for more information on backup strategies

**Initial Data Guard Environment Setup** During initial setup of a standby database, a backup of the production database is required at the secondary site to create the initial standby database.

**Recovering from Data Failures Using File or Block Media Recovery** When a block corruption, media failure, or other data failure occurs in an environment that does not include Data Guard, the only method of recovery is using an existing backup.

**Double Failure Resolution** A double failure scenario affects the availability of both the production and 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. The only 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, therefore, is necessary to restore service in the most dire of circumstances.

#### 2.5.2.2 Determine a Backup Frequency

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 too soon.

**Factors Determining Backup Frequency** Frequent backups are essential for any recovery scheme. Base the frequency of backups on your estimated recovery time objective for outages that cannot be resolved by Data Guard or Flashback technology. Repair time will be dictated by restore time plus recovery time. The frequency of the backup and the location of the backup will impact both of these factors. The other factor that influences how frequently a datafile is backed up is the rate or frequency of database changes such as:

- Addition and deletion of tables
- Insertions and deletions of rows in existing tables
- Updates to data in tables

To simplify database backup and recovery, the Oracle suggested backup strategy implements the flash recovery area while using incremental backups and updated incremental backup features. For more information about the Oracle suggested strategy, see the section titled "Using the Oracle-Suggested Backup Strategy" in *Oracle Database 2 Day DBA*.

**Establishing a Backup Retention Policy** A backup retention policy 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 is old enough to be 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 requirements. A backup that is no longer needed to satisfy the backup retention policy is said to be obsolete.

Backup retention policy can be based on redundancy or a recovery window. In a redundancy-based retention policy, you 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, you specify a time interval in the past (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 Long-Term Backups** Some businesses require the ability to maintain long-term backups that may be needed years into the future. By using RMAN with the KEEP option, it is possible to retain backups that are exempt from the retention policy and never expire, providing the capability to restore and recover the database to any desired point in time. It is important that a recovery catalog be used for the RMAN repository so that backup metadata is not lost due to lack of space, which may occur when using the target database control file for the RMAN repository.

#### 2.5.2.3 Use an RMAN Recovery Catalog

RMAN automatically manages the backup metadata in the control file of the database that is being backed up. To protect and keep backup metadata for long periods of time, the RMAN repository is created in a separate database. This repository is usually referred to as a recovery catalog. The advantages of using a recovery catalog include:

- Stores backup information long-term
- Store metadata for multiple databases
- Restore an available backup onto another system

Another reason to use a recovery catalog is the limited maximum size of the target database 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.

**See Also:** Oracle Database Backup and Recovery Advanced User's Guide for more information on RMAN repository

#### 2.5.2.4 Enable Block Change Tracking for Incremental Backups

Oracle Database 10g includes a change tracking feature for incremental backups, which improves incremental backup performance by recording changed blocks in each datafile in a change tracking file. If change tracking is enabled, then RMAN uses the change tracking file to identify changed blocks for incremental backup. This avoids the need to scan every block in the datafile, reducing the number of disk reads during backup.

**See Also:** Oracle Database Backup and Recovery Basics for more information on block change tracking

#### 2.5.2.5 Enable Autobackup for Control File and Server Parameter File

RMAN can be configured to automatically back up the control file and server parameter file (SPFILE) whenever the database structure metadata in the control file changes or when a backup record is added. The autobackup enables RMAN to recover the database even if the current control file, catalog, and SPFILE are lost. The RMAN autobackup feature is enabled with the CONFIGURE CONTROLFILE AUTOBACKUP ON statement.

**See Also:** *Oracle Database Backup and Recovery Basics* for more information on autobackup

#### 2.5.3 Backup to Disk

This section contains these topics:

- Determine Disk Backup Methods
- Create Backups in NOCATALOG Mode and RESYNC CATALOG Afterwards
- Create Database Backups on Disk in the Flash Recovery Area
- In a Data Guard Environment, Back Up to Flash Recovery Area on All Sites

#### 2.5.3.1 Determine Disk Backup Methods

When selecting a backup mechanism, the following priorities will drive your backup strategy:

- Overall backup time
- Impact on primary
- Space used by backup
- Recovery time

Table 2–4 compares different backup alternatives against the different priorities you might have. The table should guide you to choose the best backup approach for your specific business requirements. You might want to minimize backup space while sacrificing recovery time. Alternatively, a higher priority might be on recovery and backup times while space is not an issue.

Table 2–4	Comparison	of Backup	Options

Backup Option	Overall Backup Time	Impact on Primary	Space Used by Backup	Recovery Time
Full data file copy	Fast	High	Large	Fastest
Full backup sets	Faster	High	Smaller	Slowest
Incremental backup with roll forward immediately	Faster	Low	Large	Fastest
Incremental backup with roll forward deferred until recovery	Fastest	Low	Smallest	Fast

**Best Practices on Optimizing Recovery Times** If restore time is the primary concern, then either a database copy or an incremental backup with roll forward immediately should be performed. These two are the only options that provide an immediately usable backup of the database, which then needs only to be recovered to the time of the failure using archive logs created since the last backup was performed.

**Best Practices on Minimizing Space Usage** If space usage is the primary concern, then an incremental backup with a deferred roll forward should be performed. If a cumulative level 1 incremental backup is performed, then it stores only those blocks that have been changed since the last level 0 backup. With a cumulative incremental backup, the last level 1 backup need only be applied to the level 0 backup. With a differential incremental backup, all level 1 backups have to be applied to the level 0 backup. A cumulative incremental backup will initially consume more space in the flash recovery area than a differential incremental backup. But over time the cumulative incremental backup will consume less space.

#### Best Practices on Minimizing System Resource Consumption (I/O and CPU) If

system resource consumption is the primary concern, then an incremental backup with a block change-tracking file will consume the least amount of resources on the database. This is true when the amount of data changed for each backup window is below 20% of the total database size. When the amount of data changed for each

backup window exceeds 20%, then performing incremental backups will still reduce the amount of disk space required to hold the backup, but may not reduce the backup time.

#### Example

For many applications, only a small percentage of the entire database is changed each day regardless of whether the transaction rate is very high. Frequently, applications modify a same set of blocks frequently; so, the total dirty 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. The first level 0 backup takes about 180 minutes and a subsequent level 1 backup takes 20 minutes, while the merge of the backups take 45 minutes. In this example, we observed the following results:

- Level 0 backup takes 180 minutes, including READs from the data area and WRITES to the flash recovery area
- Level 1 backup takes 20 minutes, including READs from the data area and WRITES to the flash recovery area
- Rolling forward and merging the backups takes only 45 minutes included READS and WRITES from the flash recovery area, which offloads possibly contention to the data area if they use separate storage.
- The net savings are:
  - 115 minutes or 64% time savings to create a complete backup
  - Reduced I/O on the database during backups

For bigger databases, we observed even larger gains.

#### 2.5.3.2 Create Backups in NOCATALOG Mode and RESYNC CATALOG Afterwards

When creating backups to disk or tape, use the target database control file as the RMAN repository, so that backup success does not depend on the availability of the database holding the RMAN repository. This is accomplished by running RMAN with the NOCATALOG option. After the backup is complete, the new backup information stored in the target database control file can be resynchronized with the recovery catalog using the RESYNC CATALOG command.

**See Also:** *Oracle Database Backup and Recovery Reference* for more information on RESYNC CATALOG

#### 2.5.3.3 Create Database Backups on Disk in the Flash Recovery Area

Using automatic disk-based backup and recovery, you can create a flash recovery area which automates management of backup-related files:

**1.** Choose a location on disk.

This location is specified by DB\_RECOVERY\_FILE\_DEST.

2. Choose an upper bound for storage space.

This upper bound is specified by DB\_RECOVERY\_FILE\_DEST\_SIZE.

3. Set a retention policy that governs how long backup files are needed for recovery.

Oracle Database 10g then manages the storage used for backup, archived redo logs, and other recovery-related files for your database within this space. Files no longer needed are eligible for deletion when RMAN must reclaim space for new files.

# 2.5.3.4 In a Data Guard Environment, Back Up to Flash Recovery Area on All Sites

Take backups at the primary and secondary sites. The advantages of this practice include:

- Significantly reduces RTO in certain double outage scenarios
- Avoids introducing new backup procedures upon a switchover or failover
- RMAN file and block media recovery is a recovery option for data failure outages at both primary and secondary sites

Consider a scenario in which backups are done only at the secondary site. Suppose there is a site outage at the secondary site where the estimated time to recover is three days. The primary site is completely vulnerable to an outage that is typically resolved by a failover, or any outage that could be resolved by having a local backup (such as a data failure outage resolved by block media recovery).

In this scenario, a production database outage can be resolved only by physically shipping the off-site tape backups that were taken at the standby site. If primary site backups were available, then restoring locally would be an available option in place of the failover that cannot be done. Data might be lost, but having primary site backups significantly shortens the RTO.

Primary site disk backups are also necessary to ensure a reasonable RTO when using RMAN file or block media recovery. Without a local on disk backup, a backup taken at the standby site must be restored to the primary site, significantly lengthening the RTO for this type of outage.

# 2.5.4 Backup to Tape

This section contains these topics:

- Create Tape Backups from the Flash Recovery Area
- Maintain Offsite Backups

#### 2.5.4.1 Create Tape Backups from the Flash Recovery Area

Use RMAN command BACKUP RECOVERY FILE DESTINATION to move disk backups created in the flash recovery area to tape. Using a single command, all files not backed up to tape previously are backed up. This prevents you from backing up files more than once and wasting tape or tracking files not backed up before. Tape backups are used to handle certain outage scenarios and for offsite and long-term storage.

#### 2.5.4.2 Maintain Offsite Backups

Regardless of the architecture deployed, including the existence of a standby database, it is still important to have offsite backups for business requirements, to protect against disasters, and to comply with legal and regulatory requirements such as the Securities and Exchange Commission (SEC) and Health Insurance Portability and Accountability Act (HIPPA).

# 2.5.5 Backup and Recovery Maintenance

This section contains these topics:

- Regularly Check Database Files for Corruption
- Periodically Test Recovery Procedures

Regularly Backup the Recovery Catalog Database

#### 2.5.5.1 Regularly Check Database Files for Corruption

Use RMAN command BACKUP VALIDATE RMAN 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 verifies content-checking for physical and logical errors, but does not actually perform the backup or recovery operation. Oracle 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.

If BLOCK CHANGE TRACKING is enabled, then do not use the INCREMENTAL LEVEL option with BACKUP VALIDATE to ensure that all data blocks are read and verified.

To detect all types of corruption that are possible to detect, specify the CHECK LOGICAL option. Do not specify the MAXCORRUPT OR NOCHECKSUM option of the BACKUP VALIDATE command.

#### 2.5.5.2 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.

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

#### 2.5.5.3 Regularly Backup the Recovery Catalog Database

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.

# 2.6 Configuring Fast Application Failover

To benefit fully from fast instance and database failover and switchover with Real Application Clusters (RAC) and Data Guard, you should configure fast application failover. When a database service becomes unavailable, fast application 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.

Because client failover features have evolved through several Oracle Database releases, the time required for clients to respond to various outages will vary by release. The time required for failover in certain cases is a direct function of TCP/IP network timeouts.

Table 2–5 shows typical wait times when using client failover features

Oracle Database Version	Client Type	Site Failure	RAC Node Failure	Non-RAC Instance Failure	RAC Instance Failure
8.0, 8 <i>i</i> , 9 <i>i</i>	All	TCP timeout	TCP timeout	Seconds to minutes <sup>1</sup>	Seconds
10g Release 1 (10.1)	JDBC	TCP timeout	Seconds	Seconds to minutes <sup>1</sup>	Seconds

Table 2–5 Typical Wait Times for Client Failover

Oracle Database Version	Client Type	Site Failure	RAC Node Failure	Non-RAC Instance Failure	RAC Instance Failure
10g Release 1 (10.1)	OCI	TCP timeout	TCP timeout	Seconds to minutes <sup>1</sup>	Seconds
10g Release 2 (10.2)	JDBC	Seconds	Seconds	Seconds	Seconds
10g Release 2 (10.2)	OCI	Seconds <sup>2</sup>	Seconds	Seconds	Seconds

Table 2–5 (Cont.) Typical Wait Times for Client Failover

<sup>1</sup> The wait times required in non-RAC instance failures are determined by how much time is needed to activate the standby database as the new primary database and for the client to establish a new connection.

<sup>2</sup> Excluding ODP.NET clients, who suffer an outage equal to that of TCP timeout

With Oracle Database 10g Release 2 (10.2), delays caused by TCP/IP network timeouts are overcome for both JDBC clients and OCI clients by using fast application failover. To use fast application failover for site failovers, a trigger is written that is invoked by the DB\_ROLE\_CHANGE system event. This trigger can also manage post-failover tasks.

A very detailed description of how to achieve seamless client failover is covered in *Oracle Database 10g Release 2 Best Practices: Client Failover for Highly Available Oracle Databases* at

http://www.oracle.com/technology/deploy/availability/pdf/MAA\_WP\_ 10gR2\_ClientFailoverBestPractices.pdf.

This section contains these topics:

- Configuring Clients for Failover
- Client Failover in a RAC Database
- Failover from a RAC Primary Database to a Standby Database

#### 2.6.1 Configuring Clients for Failover

For JDBC client failover best practices, follow these steps:

- 1. Enable fast connection failover for JDBC clients by setting the DataSource property FastConnectionFailoverEnabled to TRUE.
- **2.** Configure JDBC clients to use a connect descriptor that includes an address list which includes the VIP address for each node in the cluster and connects to an existing service.
- **3.** Configure a remote Oracle Notification Service (ONS) subscription on the JDBC client so that an ONS daemon is not required on the client.

For OCI client failover best practices, follow these steps:

- 1. Enable fast application notification (FAN) notifications for OCI clients by initializing the environment with the OCI\_EVENTS parameter.
- **2.** Link the OCI client applications with the thread library.
- **3.** Set the AQ\_HA\_NOTIFICATIONS parameter to TRUE and configure the transparent application failover (TAF) failover attributes for services.

#### 2.6.2 Client Failover in a RAC Database

For client failover best practices in a RAC database:

- 1. Use Oracle Enterprise Manager to create new services.
- 2. Add all hosts in the cluster to the RAC ONS configuration.

# 2.6.3 Failover from a RAC Primary Database to a Standby Database

For client failover best practices when failing over from a RAC primary database to a standby database, follow these steps:

- 1. Create the necessary support files for Oracle Net alias modifications.
- **2.** Configure the FAN ONS publisher program so that JDBC clients are notified of the primary site failure and instructed to reconnect to the new primary database.
- **3.** Create a trigger on the DB\_ROLE\_CHANGE system event to perform the proper steps so that JDBC and OCI clients connect to the new primary database following a role change after a Data Guard failover or switchover.

# **Monitoring Using Oracle Grid Control**

This chapter provides best practices for using Oracle Grid Control to monitor and maintain a highly available environment across all tiers of the application stack.

- Overview of Monitoring and Detection for High Availability
- Using Oracle Grid Control for System Monitoring
- Managing the High-Availability Environment with Oracle Grid Control

# 3.1 Overview of Monitoring and Detection for High Availability

Continuous monitoring of the system, 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 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 to responsible parties for critical failures.

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 of problems.

Oracle Grid Control provides the management and monitoring capabilities with many different notification options. This chapter provides best practices for using Oracle Grid Control to monitor and maintain a highly available environment across all tiers of the application stack. 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.

# 3.2 Using Oracle Grid Control for System Monitoring

This section provides an overview of the concepts and facilities available in Oracle Grid Control.

A major benefit of Oracle Grid Control is its ability to manage components across the entire application stack from the host operating system to a user or packaged application. Oracle Grid Control 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 together by application type. All targets can also be reviewed in a single view. Each target type has a default

generated home page that displays a summary of relevant details for a specific target. Different types of targets can be grouped together 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 machine and is responsible for a set of targets. The targets can be on a machine that is different from the machine that the Management Agent is on. 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.

The Oracle Grid Control home page shown in Figure 3–1 provides a picture of the availability of all of the discovered targets.

Edit View Favorites Tools Help			
ack • → - 🎯 🖄 🖓 Öğ Search 📷 Favorites 🧐 Media 🎯 🖏	- 🎝 🔟 - 🗏 🎄		
ess 👔 hktp://gridconkrol.oraclecorp.com/em/console/home			
RACLE Enterprise Manager 10g	Home Targets Configuration		tup Preferences Help Log Management System
		Page Refres	shed Sep 5, 2003 2:48:57
View All Targets	Target Search		
Status Total Monitored Targets 153	Search All Targets		60
Groups 8 All Targets Availability	Critical Patch Advisories		
5%	No Outstanding Critical Patch Advisories		
20%	View Databases	•	
76 5	Databases A	Targets	Interim Installations Patches
All Targets Alerts	Oracle9i 9.0.1.4.0	0	1 Not available
Critical 🔗 32 Warning 🚳 25	Oracle9i 9.0.1.5.0	0	1 Not available
Errors 🔂 63 All Targets Jobs	Oracle9i 9.2.0.1.0	1	1 Not available
Suspended Executions 0	Oracle9i 9.2.0.4.0	1	2 Not available
Problem Executions Z (Last 7 days)	Oracle9i 9.0.1.4.0a	4	Z Not available
	Oracle Database 10G 10.2.0.1.0	1	2 Not available
	Resource Center		
	Documentation		0뷰 Local intranet

Figure 3–1 Oracle Grid Control Home Page

The Oracle Grid Control home page shows the following major kinds of information:

- A snapshot of the current availability of all targets. The pie chart associated with availability gives the administrator an immediate indication of any target that is unavailable (Down) or has lost communication with the console (Unknown).
- An overview of how many alerts (for events) and problems (for jobs) are known in the entire monitored system. You can display detailed information by clicking the links, or by navigating to Alerts from the upper right portion of any Oracle Grid Control page.
- A target shortcut intended for administrators who have to perform a task for a specific target.
- An overview of what is actually discovered in the system. This list can be shown at the hardware level and the Oracle level.
- A set of useful links to other Oracle online resources.

Alerts are 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. It could be the availability of a component through a simple heartbeat test, or 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 event 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.

The rest of this section includes the following topics:

- Set Up Default Notification Rules for Each System
- Use Database Target Views to Monitor Health, Availability, and Performance
- Use Event Notifications to React to Metric Changes
- Use Events to Monitor Data Guard System Availability

**See Also:** Oracle Enterprise Manager Concepts for more information about monitoring and using metrics in Oracle Grid Control

## 3.2.1 Set Up Default Notification Rules for Each System

**Notification Rules** are defined sets of alerts on metrics that are automatically applied to a target when it is discovered by Oracle Grid Control. For example, an administrator can create a rule that monitors the availability of database targets and generates an e-mail message if a database fails. After that rule is generated, it is applied to all existing databases and any database created in the future. Access these rules by navigating to **Preferences** and then choosing **Rules**.

The rules monitor problems 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.

Oracle Grid Control provides a series of default rules that provide a strong framework for monitoring availability. A default rule is provided for each of the preinstalled target types that come with Oracle Grid Control. These rules can be modified to conform to the policies of each individual site, and new rules can be created for site-specific targets or applications. The rules can also be set to notify users during specific time periods to create an automated coverage policy.

Consider the following recommendations:

 Modify each rule for high-value components in the target architecture to suit the required availability requirements by using the rules modification wizard. For the database rule, set the events in Table 3–1, Table 3–2, and Table 3–3 for each target. The frequency of the monitoring is determined by the service level agreement (SLA) for each component.

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.

#### See Also:

- Oracle Enterprise Manager Concepts for conceptual information
   about Beacons
- Oracle Enterprise Manager Advanced Configuration for information about configuring service tests and Beacons
- Add Notification Methods and use them in each Notification Rule. By default, the easiest method for alerting an administrator to a potential problem is to send e-mail. 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 the problem that might occur if a component of the e-mail system has failed. Set additional Notification Methods by using the **Set-up** link at the top of any Oracle Grid Control page.
- Modify Notification Rules to notify the administrator when there are errors in computing 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.

Figure 3–2 shows the Notification Rule property page for choosing availability states with Down, Agent Unreachable, Agent Unreachable Resolved, and Metric Error Detected chosen.



Figure 3–2 Setting Notification Rules for Availability

In addition, modify the metrics monitored by the database rule to report the metrics shown in Table 3–1, Table 3–2, and Table 3–3. This ensures that these metrics are captured for all database targets and that trend data will be available for future

analysis. All events described in Table 3–1, Table 3–2, and Table 3–3 can be accessed from the **Database Homepage** by choosing **All Metrics** > **Expand All**.

Space management conditions that have the potential to cause a service outage should be monitored using the events shown in Table 3–1.

 Table 3–1
 Recommendations for Monitoring Space

Metric	Recommendation			
Tablespace Space Used (%)	Set this metric to monitor root file systems for any critical hardware server. This metric enables the administrator to choose the threshold percentages that Oracle Grid Control tests against, as well as the number of samples that must occur in error before a message is generated to the administrator. The recommended default settings are 70 percent for a warning and 90 percent for an error, but these values should be adjusted depending on system usage. This metric can be customized to monitor only specific tablespaces.			
	This metric and similar events can be set in the Tablespace Full metric group.			
Archiver Hung Alert Log Error	Set this metric to monitor the alert log for ORA-00257 errors, which indicate a full archive log directory.			
	This metric can be set in the Alert Log Error Status metric group.			
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 default settings are 70 percent for a warning and 90 percent for an error, but these should be adjusted depending on system usage.			
	This metric can be set in the Dump Area metric group.			

From the Alert Log Metric group, set Oracle Grid Control to monitor the alert log for errors as shown in Table 3–2.

Table 3–2	Recommendations	for N	lonitoring	the <i>i</i>	Alert L	og
-----------	-----------------	-------	------------	--------------	---------	----

Metric	Recommendation
Alert	Set this metric to send an alert when an ORA-6XX, ORA-1578 (database corruption), or ORA-0060 (deadlock detected) error occurs. If any other error is recorded, then a warning message is generated.
Data Block Corruption	Set this metric to monitor the alert log for ORA-01157 and ORA-27048 errors. They signal a corruption in an Oracle Database datafile.

Monitor the system to ensure that the processing capacity is not exceeded. The warning and critical levels for these events should be modified based on the usage pattern of the system. Set the events from the Database Limits metric group using the recommendations in Table 3–3.

Table 3–3 Recommendations for Monitoring Processing Capacity

Metric	Recommendation
Process limit	Set thresholds for this metric to warn if the number of current processes approaches the value of the PROCESSES initialization parameter.
Session limit	Set thresholds for this metric to warn if the instance is approaching the maximum number of concurrent connections allowed by the database.

Figure 3–3 shows the Notification Rule property page for setting choosing metrics. The user has chosen Critical and Warning as the severity states for notification. The list of

Available Metrics is shown in the left list box. The metrics that have been selected for notification are shown in the right list box.

Figure 3–3 Setting Notification Rules for Metrics

Proferences Properties Targets Availability M	Home etrics Objet	Targets Configuration Alerts Jobs Management System
it Notification Rule Database Availability and Critica act the metrics and their severities for which you would like to receive notifications.		Cancel Back Step 4 of 7 Next
Severity States		
Select the severity states for which you would like to receive notification.		
Available Metrics		Selected Metrics
Active Sessions Walting VD (%) Active Sessions Walting: Other (%) Archiver Hung Alen LagE Front Audited User Average File Read Time (cent-seconds) Average File With Time (cent-seconds)	Move Move All Remove	Session Limit Usage (%) Blocking Session Court Archiner Hung Alert Log Error Status Data Block Corruption Alert Log Error Status Generic Alert Log Error Status Session Terminated Alert Log Error Status Archine Area Usade (%) Segments Not Able to Extend Count Seaments Not Able To Extend Count

#### See Also:

- *Oracle Database 2 Day DBA* for information about setting up notification rules and metric thresholds
- Enterprise Manager Framework, Host, and Third-Party Metric Reference Manual for information on available metrics

## 3.2.2 Use Database Target Views to Monitor Health, Availability, and Performance

The Database Targets page in Figure 3–4 shows an overview of system performance, space utilization, and the configuration of important availability components like archived redo log status, flashback log status, and estimated instance recovery time. Alerts are displayed immediately. Each of the alert values can be configured from links on this page.

Host: dsumd03 us.coade.com > Dotabase: main_031009_dsumd03 Database: main_031009_dsumdf03
Home Performance Administration Maintenance
Latest Data Collected From Target Nov 11, 2003 2:34:11 PM Refeesh
View Date   Manually
General     Host CPU     Active Sessions       Image: Status Up Statu
High Availability         Space Usage         Diagnostic Summary           Instance Recovery Time (seconds)         1         Performance Problem Tablespaces         Performance Not Configured         0           Last Backup         Oct 21, 2003 10:62:48 AM         Segment Findings         Not Configured         All Policy Violations         334           Archiving         Disable         Duma Area Used (%) < 19

#### Figure 3–4 Overview of System Performance

Many of the metrics from the Oracle Grid Control pertain to performance. A system without adequate performance is not an HA system, regardless of the status of any individual component. 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 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 SLA, 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 Oracle Grid Control to capture a snapshot of database performance as a baseline. Oracle Grid Control compares these values against system performance and displays the result on the database Target page. It can also send alerts if the values deviate too far from the established baseline.

Set the database notification rule to capture the metrics listed in Table 3–4 for all database targets. Analysis of these parameters can then be done using one tool and historical data will be available.

Table 3–4 Recommended Notification Rules for Metrics

Metric	Recommendation				
Disk I/O per Second	This is a database-level metric that monitors I/O operations done by the database. It sends an alert when the number of operations exceeds a user-defined threshold. Use this metric with operating system-level events that are also available with Oracle Grid Control.				
	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.				
% CPU Busy	Set this metric to warn at 75 percent and to show a critical alert between 85 percent and 90 percent. 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	Excessive idle time indicates that a bottleneck for one or more resources is occurring. Set this metric based on the system wait time when the application is performing as expected.				
Network Bytes per Second	This metric reports network traffic that Oracle generates. It can indicate a potential network bottleneck. Set this metric based in actual usage during peak periods.				
Total Parses per Second	This metric measures SQL performance. It can indicate an application change or change in usage that has created a shortage of resources. Set it based on peak periods.				

#### See Also:

- Oracle Database Performance Tuning Guide for more information about performance monitoring
- Oracle Database 2 Day DBA for more information on monitoring and tuning using Enterprise Manager

## 3.2.3 Use Event Notifications to React to Metric Changes

There are many operating system events that can be used to supplement a suggested metric. Such operating system events are not required for each host and instance. All metrics defined here can be set individually by instance or database using the **Manage Metrics** link at the bottom of the navigation bar on the object target page. The values that trigger a warning or critical alert can be changed here, and an operating system script can be activated to respond to an metric threshold, in addition to the standard alert being generated to the Oracle Grid Control.

## 3.2.4 Use Events to Monitor Data Guard System Availability

Set Oracle Grid Control metrics to monitor the availability of logical and physical Data Guard configurations. If a Data Guard environment is registered with the Data Guard Manager extension of Oracle Grid Control, then set the events shown in Table 3–5.

Metric	Recommendation			
Data Guard Status	Set this metric to be notified of system problems in a Data Guard configuration.			
Data Not Applied	Set this metric to be notified when the gap (measured in minutes) between the last archived redo log received and the last log applied on the standby database exceeds a user-defined threshold. This information can be used to warn the administrator if the recovery time for a standby instance will exceed the defined outage recovery service level. Set this metric based on the specifications for log application for the standby database.			
Data Not Received	Set this metric to be notified if there is an extended delay in moving archived redo logs from the production database to the standby database. This metric occurs when the difference between the number of archived redo logs on the production database and the number of archived redo logs shipped to the standby site exceeds a user-defined threshold. The threshold should be based on the amount of time it takes to transport an archived redo log across the network.			
	Set the sample time for the metric to be approximately the redo transport time, and set the number of occurrences to be 2 or greater to avoid false positives. Recommended starting values for the warning and critical thresholds are 1 and 2.			

Table 3–5 Recommendations for Setting Data Guard Events

# 3.3 Managing the High-Availability Environment with Oracle Grid Control

Use Oracle Grid Control as a proactive part of administering any system as well as for problem notification and analysis. This section includes the following recommendations:

- Check Oracle Grid Control Policy Violations
- Use Oracle Grid Control to Manage Oracle Patches and Maintain System Baselines
- Use Oracle Grid Control to Manage Data Guard Targets

**See Also:** Oracle Enterprise Manager Administrator's Guide

## 3.3.1 Check Oracle Grid Control Policy Violations

Oracle Grid Control comes with a pre-installed set of policies and recommendations of best practices for all databases. These policies are checked by default, and the number of violations is displayed on the Targets page shown in Figure 3–4. To see a list of all violations, select **Policy Violations** from the Targets page.

**See Also:** Oracle Enterprise Manager Policy Reference Manual for definitions of existing policies

# 3.3.2 Use Oracle Grid Control to Manage Oracle Patches and Maintain System Baselines

You can use Oracle Grid Control to download and manage patches from https://metalink.oracle.com for any monitored system in the application
environment. A job can be set up to routinely check for patches that are relevant to the
user environment. Those patches can be downloaded and stored directly in the
Management Repository. Patches can be staged from the Management Repository to
multiple systems and applied during maintenance windows.

You can examine patch levels for one machine and compare them between machines in either a one-to-one or one-to-many relationship. In this case, a machine can be identified as a baseline and used to demonstrate maintenance requirements in other machines. This can be done for operating system patches as well as database patches.

# 3.3.3 Use Oracle Grid Control to Manage Data Guard Targets

Oracle Grid Control can be used to set up logical and physical standby databases for any database target. It also provides the ability to manage switchover and failover of database targets other than the database that contains the Management Repository.

Oracle Grid Control can also be used to monitor the health of a Data Guard configuration at a glance. From any database target page, navigate to the Data Guard status section by using the link in the High Availability section. The page shows the active standby databases for the primary target, the amount of log data waiting for shipment and receipt by the standby database, and the data protection mode. You can also modify the data protection mode from this page.

This page contains a link to the **Verify** function, which checks the Data Guard environment and redo transport services to display warnings and errors. The Verify function is not automatic and must be run manually.

See Also: Oracle Data Guard Broker for use case scenarios

# **Managing Outages**

This chapter describes unscheduled and scheduled outages and the Oracle operational best practices that can tolerate or manage each outage type and minimize downtime.

This chapter contains these topics:

- Outage Overview
- Recovering from Unscheduled Outages
- Restoring Fault Tolerance
- Eliminating or Reducing Downtime for Scheduled Outages

# 4.1 Outage Overview

This section describes the types of possible outages and the recommended methods to repair or minimize the downtime associated with each outage.

This section contains these topics:

- Unscheduled Outages
- Scheduled Outages

# 4.1.1 Unscheduled Outages

Unscheduled outages are unanticipated failures in any part of the technology infrastructure that supports the application, including the following components:

- Hardware
- Software
- Network infrastructure
- Naming services infrastructure
- Database

Your monitoring and high-availability infrastructure should provide rapid detection and recovery from downtime. Chapter 3, "Monitoring Using Oracle Grid Control" describes detection, while this chapter focuses on reducing downtime.

Table 4–1 describes unscheduled outages that affect the primary or secondary site components.

Outage Type	Description	Examples		
Site failure	The entire site where the current production database resides is	Disaster at the production site such as a fire, flood, or earthquake		
	unavailable. This includes all tiers of the application.	Malicious attack on the site		
	of the application.	Power outages. If there are multiple power grids and backup generators for critical systems, then this should affect only part of the data center.		
Clusterwide failure	The whole cluster hosting the RAC database is unavailable or fails. This	The last surviving node on the RAC cluster fails and cannot be restarted		
	includes failures of nodes in the cluster, and any other components that result in the cluster being	Both of the redundant cluster interconnects fail or clusterware failure or problem		
	unavailable and the Oracle database and instances on the site	Database corruption is severe enough to disallow continuity on the current data server		
	being unavailable.	Disk storage fails A database tier node fails or has to be shut down because of bad memory or bad CPU		
Computer failure (node)	A node of the RAC cluster is unavailable or fails			
		The database tier node is unreachable		
		Both of the redundant cluster interconnects fail, resulting in another node taking ownership		
Computer failure (instance)	A database instance is unavailable or fails	An instance of the RAC database on the data server fails because of a software bug, an operating system problem, or a hardware problem.		
Storage failure	Storage holding some or all of the	Disk drive failure		
	database contents becomes unavailable, because it has shut	Disk controller failure		
	down or is no longer accessible	Storage array failure		
Data corruption	Parts of the database are	A datafile is accidentally removed or is unavailable		
	unavailable because of media corruption, inaccessibility, or	Media corruption affects blocks of the database		
	inconsistencies	Oracle block corruption is caused by operating system or other node-related problems		
Human error	Parts of the database are unavailable, and transactional or logical data inconsistencies arise.	Localized damage (needs surgical repair) Human err results in a table being dropped or in rows being deleted from a table		
	Usually caused by an operator or bugs in the application code.	Widespread damage (needs drastic action to avoid downtime) Application errors result in logical corruption in the database, or operator error results in a batch job being run more times than specified.		
		<b>Note:</b> This category focuses on human errors that affect database availability and, in particular, cause transactional or logical data inconsistencies.		

Table 4–1 Unscheduled Outages

The best practice recommendations for reducing unscheduled outages on the primary site and the secondary site, estimated recovery times, and recovery steps appear in the following sections:

- Managing Unscheduled Outages on the Primary Site
- Managing Unscheduled Outages on the Secondary Site

#### 4.1.1.1 Managing Unscheduled Outages on the Primary Site

If the primary site contains the production database and the secondary site contains the standby database, then outages on the primary site are the most crucial. Solutions for these outages are critical for maximum availability of the system. Only the Oracle Database 10*g* with Data Guard, and the Oracle Database 10*g* with RAC and Data Guard (MAA) architectures have a secondary site to protect from site disasters.

Table 4–2 summarizes the recovery steps for unscheduled outages on the primary site. For outages that require multiple recovery steps, the table includes links to the detailed descriptions in Section 4.2, "Recovering from Unscheduled Outages" that starts on page 4-9.

Outage Type	Oracle Database 10 <i>g</i> Hours to days		Oracle Database 10g with RAC Hours to days		Oracle Database 10g with Data Guard		Oracle Database 10g - MAA Seconds to 5 minutes <sup>1</sup>	
Site failure								
	1. 2.	Restore site. Restore from tape backups.	1. 2.	Restore site. Restore from tape backups.	1.	Database Failover with a Standby Database on	1.	Database Failover with a Standby Database on
	3.	Recover database.	3.	Recover database.	2.	page 4-13 Complete Site Failover on page 4-10	2.	page 4-13 Complete Site Failover on page 4-10
					3.	Application Failover on page 4-25	3.	Application Failover on page 4-25
Clusterwide	Not applicable		<ol> <li>Hours to days</li> <li>Restore cluster or restore at least one node.</li> <li>Optionally restore from tape backups if the data is lost or corrupted.</li> </ol>		Not applicable		Seconds to 5 minutes	
failure		1. Database Failov with a Standby Database on						
							2.	page 4-13 Application Failover on page 4-25
			3.	Recover database.				
Computer	Minutes to hours <sup>2</sup>		No downtime <sup>3</sup>		Seconds to 5 minutes <sup>2</sup>		No downtime <sup>3</sup>	
failure (node)	restart database.	Restart node and restart database. Reconnect users.	by <mark>R</mark>	anaged automatically RAC Recovery for nscheduled Outages	1.	Database Failover with a Standby Database on page 4-13	aut Ree Un	inaged tomatically by RAC covery for ischeduled Outages
					2.	Application Failover on page 4-25	on	page 4-23
					or			
					Mi	nutes to hours <sup>2</sup>		
					1.	Restart node and restart database.		
					2.	Reconnect users.		

Table 4–2 Recovery Times and Steps for Unscheduled Outages on the Primary Site

Outage Type Oracle Database 10g		Oracle Database 10 <i>g</i> with RAC	Oracle Database 10 <i>g</i> with Data Guard	Oracle Database 10 <i>g</i> - MAA	
Computer	Minutes <sup>2</sup>	No downtime <sup>3</sup>	Minutes <sup>2</sup>	No downtime <sup>3</sup>	
failure (instance)	<ol> <li>Restart instance.</li> <li>Reconnect users.</li> </ol>	Managed automatically by RAC Recovery for Unscheduled Outages	<ol> <li>Restart instance.</li> <li>Reconnect users.</li> <li><i>or</i></li> <li>Seconds to 5 minutes<sup>1</sup></li> <li>Database Failover with a Standby Database on page 4-13</li> <li>Application Failover on page 4-25</li> </ol>	Managed automatically by RAC Recovery for Unscheduled Outages on page 4-23	
Storage failure	No downtime <sup>4</sup> ASM Recovery After Disk and Storage Failures on page 4-25	No downtime <sup>4</sup> ASM Recovery After Disk and Storage Failures on page 4-25	No downtime <sup>4</sup> ASM Recovery After Disk and Storage Failures on page 4-25	No downtime <sup>4</sup> ASM Recovery After Disk and Storage Failures on page 4-25	
Data corruption	HARD prevents data corruption <sup>5</sup>	HARD prevents data corruption <sup>5</sup>	HARD prevents data corruption <sup>5</sup>	HARD prevents data corruption <sup>5</sup>	
	Potentially hours	Potentially hours	or	or	
	Recovering from Data Corruption (Data Failures)	Recovering from Data Corruption (Data Failures)	<ul> <li>Seconds to 5 minutes<sup>1</sup></li> <li>1. Database Failover with a Standby Database on page 4-13</li> <li>2. Application</li> </ul>	<ul> <li>Seconds to 5 minutes<sup>1</sup></li> <li>1. Database Failover with a Standby Database on page 4-13</li> <li>2. Application</li> </ul>	
			Failover on page 4-25	Failover on page 4-25	
Human error	< 30 minutes <sup>6</sup>	< 30 minutes <sup>6</sup>	<30 minutes <sup>6</sup>	< 30 minutes <sup>6</sup>	
		Recovering from Human Error	Recovering from Human Error on page 4-37	Recovering from Human Error on page 4-37	

Table 4–2	(Cont.)	Recovery	Times and	Steps for	r Unscheduled	Outages on	the Primary Site
-----------	---------	----------	-----------	-----------	---------------	------------	------------------

<sup>1</sup> Recovery time indicated applies to database and existing connection failover. Network connection changes and other site-specific failover activities may lengthen overall recovery time.

<sup>2</sup> Recovery time consists largely of the time it takes to restore the failed system.

<sup>3</sup> Database is still available, but portion of application connected to failed system is temporarily affected.

<sup>4</sup> Storage failures are prevented by using Automatic Storage Management (ASM) with mirroring and its automatic rebalance capability.

<sup>5</sup> Not all types of data corruption are prevented. For the most recent information about the HARD initiative, refer to Section 2.1.6, "Consider HARD-Compliant Storage" on page 2-7.

<sup>6</sup> 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 flashback the appropriate transactions.

## 4.1.1.2 Managing Unscheduled Outages on the Secondary Site

For most cases, outages on the secondary site can be managed with no effect on availability of the primary database located on the primary site. However, if the configuration is in maximum protection mode, then unscheduled outages on the last surviving standby database will cause outages on the production database to ensure no data loss when failing over to the standby database. After downgrading the data protection mode, you can restart the production database even without accessibility to the standby databases. Outages on the secondary site might affect the maximum time to recovery (MTTR) if there are concurrent failures on the primary site.

Table 4–3 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 4.2, "Recovering from Unscheduled Outages" that starts on page 4-9.

Table 4–3 Recovery Steps for Unscheduled Outages on the Secondary Site

Outage Type	Oracle Database 10 <i>g</i> with Data Guard	Oracle Database 10 <i>g</i> - MAA	
Computer failure (instance)	<ol> <li>Restart node and standby instance.</li> <li>Restart recovery.</li> <li>If there is only one standby database and</li> </ol>	There is no effect on production availability if the production database Oracle Net descriptor is configured to use connect-time failover to an available standby instance.	
	if maximum database protection is configured, then the production database will shut down to ensure that there is no	Broker will automatically restart the apply process.	
	data divergence with the standby database.	Restart node and instance when they are available.	
Data corruption	Restoring Fault Tolerance After a Standby Database Data Failure on page 4-54	Restoring Fault Tolerance After a Standby Database Data Failure on page 4-54	
Primary database opens with RESETLOGS because of Flashback Database operations or point-in-time media recovery	Restoring Fault Tolerance After the Production Database Was Opened Resetlogs on page 4-55	Restoring Fault Tolerance After the Production Database Was Opened Resetlogs on page 4-55	

## 4.1.2 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 changes and patching
- Changes to improve performance and manageability of systems

These tasks should be scheduled at times best suited for continual application availability.

Table 4–4 describes scheduled outages that affect either the primary or secondary site.

Outage Scope	Description	Examples
Site-wide	The entire site where the current	Scheduled power outages
	production database resides is unavailable. Usually known well in	Site maintenance
	advance.	Regular planned switchovers to test infrastructure
Hardware maintenance (node impact)	Hardware maintenance on a database server. Restricted to a node of the	Repair of a failed component such as a memory card or CPU board
	database cluster.	Addition of memory or CPU to an existing node in the database tier
Hardware maintenance (clusterwide impact)	Hardware maintenance on a database server cluster	Some cases of adding a node to the cluster
		Upgrade or repair of the cluster interconnect
		Upgrade to the storage tier that requires downtime on the database tier
System software maintenance (node impact)	System software maintenance on a database server. The scope of the	Upgrade of a software component such as the operating system
	downtime is restricted to a node.	Changes to the configuration parameters for the operating system
System software maintenance (clusterwide	System software maintenance on a database server cluster	Upgrade or patching of the cluster software
impact)		Upgrade of the volume management software
Oracle patch upgrade for the database	Scheduled outage for installation of an Oracle patch	Patch Oracle software to fix a specific customer issue
Oracle patch set or software upgrade for the database	Scheduled outage for Oracle patch set or software upgrade	Patching Oracle software with a patch set
		Upgrading Oracle software
Database object reorganization	Changes to the logical structure or the physical organization of Oracle Database	Moving an object to a different tablespace
	objects, primarily to improve performance or manageability.	Converting a table to a partitioned table
	Using Oracle Database online reorganization features enables objects to be available during the reorganization.	Renaming or dropping columns of a table
Storage maintenance	Maintenance of storage where database	Converting to ASM
	files reside	Adding or removing storage
Platform migration	Changing operating system platform of the primary and standby databases	Moving to the Linux operating system
Location migration	Changing physical location of the primary database	Moving the primary database from one data center to another.

# Table 4–4 Scheduled Outages

The following sections provide best practice recommendations and preparations for reducing scheduled outages on the primary and secondary sites:

- Managing Scheduled Outages on the Primary Site
- Managing Scheduled Outages on the Secondary Site
- Preparing for Scheduled Outages on the Secondary Site

## 4.1.2.1 Managing Scheduled Outages on the Primary Site

If the primary site contains the production database and the secondary site contains the standby database, then outages on the primary site are the most crucial. Solutions for theses outages are critical for continued availability of the system.

Table 4–5 shows the high-level recovery steps for scheduled outages on the primary site. For outages that require multiple recovery steps, the table includes links to the detailed descriptions in Section 4.4, "Eliminating or Reducing Downtime for Scheduled Outages" beginning on page 4-57.

Outage Scope	Cause	Oracle Database 10 <i>g</i>	Oracle Database 10 <i>g</i> with RAC	Oracle Database 10 <i>g</i> with Data Guard	Oracle Database 10 <i>g</i> - MAA	
Site	Site shutdown	Restart database after outage	Restart database after outage	1. Database Switchover with a Standby Database on page 4-19	1. Database Switchover with a Standby Database on page 4-19	
				2. Complete Site Failover on page 4-10	2. Complete Site Failover on page 4-10	
				3. Application Failover on page 4-25	<b>3.</b> Application Failover on page 4-25	
Primary database	Hardware maintenance (node impact)	Restart database after outage	Managed automatically, see System Maintenance on page 4-70	1. Database Switchover with a Standby Database on page 4-19	Managed automatically, see System Maintenance on page 4-70	
				2. Application Failover on page 4-25		
Primary database	Hardware maintenance (clusterwide impact)	Not applicable	Restart database after outage	Not applicable	1. Database Switchover with a Standby Database on page 4-19	
	System software maintenance (clusterwide impact)				<ol> <li>Application Failover on page 4-25</li> </ol>	
Primary database	System software maintenance (node impact)	Restart database after outage	Managed automatically by System Maintenance on page 4-70	<ol> <li>Database Switchover with a Standby Database on page 4-19</li> </ol>	Managed automatically by System Maintenance on page 4-70	
				2. Application Failover on page 4-25		
Primary database	Real Application Cluster's Cluster Ready Service (CRS) Upgrades	Not applicable	In general, CRS upgrades can be done online and do not require downtime.	Not applicable	In general, CRS upgrades can be done online and do not require downtime.	

Table 4–5 Recovery Steps for Scheduled Outages on the Primary Site

Outage Scope	Cause	Oracle Database 10 <i>g</i>	Oracle Database 10 <i>g</i> with RAC	Oracle Database 10 <i>g</i> with Data Guard	Oracle Database 10 <i>g</i> - MAA
Primary database	Oracle patch upgrade for the database	Restart database after outage	RAC Database Patches on page 4-58	1. Database Switchover with a Standby Database on page 4-19	RAC Database Patches on page 4-58
				2. Application Failover on page 4-25	
Primary database	Oracle patch set or software upgrade for the database, including some Automatic Storage Management (ASM) and CRS upgrades	Restart database after outage	Restart database after outage	Database Upgrades on page 4-61	Database Upgrades on page 4-61
Primary database	Database object reorganization	Database Object Reorganization on page 4-68	Database Object Reorganization on page 4-68	Database Object Reorganization on page 4-68	Database Object Reorganization on page 4-68
Primary database	Storage maintenance	Storage Maintenance on page 4-57	Storage Maintenance on page 4-57	Storage Maintenance on page 4-57	Storage Maintenance on page 4-57
Primary database	Platform maintenance	Database Platform or Location Migration on page 4-63	Database Platform or Location Migration on page 4-63	Database Platform or Location Migration on page 4-63	Database Platform or Location Migration on page 4-63
Primary database	Location maintenance	Database Platform or Location Migration on page 4-63	Database Platform or Location Migration on page 4-63	Database Platform or Location Migration on page 4-63	Database Platform or Location Migration on page 4-63

Table 4–5 (Cont.) Recovery Steps for Scheduled Outages on the Primary Site

#### 4.1.2.2 Managing Scheduled Outages on the Secondary Site

Outages on the secondary site do not affect availability because the clients always access the primary site. Outages on the secondary site might affect the RTO if there are concurrent failures on the primary site. Outages on the secondary site can be managed with no effect on availability. If maximum protection database mode is configured, then downgrade the protection mode before scheduled outages on the standby instance or database so that there will be no downtime on the production database.

Table 4–6 describes the recovery steps for scheduled outages on the secondary site.

Outage Type	Cause	Oracle Database 10 <i>g</i> with Data Guard	Oracle Database 10 <i>g</i> - MAA
Site	Site shutdown	Before the outage:	Before the outage:
		Preparing for Scheduled Outages on the Secondary Site on page 4-9	Preparing for Scheduled Outages on the Secondary Site on page 4-9 After the outage:
		After the outage:	Restoring Fault Tolerance After
		Restoring Fault Tolerance After Planned Downtime on Secondary Site or Clusterwide Outage on page 4-53	Planned Downtime on Secondary Site or Clusterwide Outage on page 4-53
Standby	Hardware or software	Before the outage:	Before the outage:
database maintenance the node that is running the managed recovery process (MRP)		Preparing for Scheduled Outages on the Secondary Site on page 4-9	Preparing for Scheduled Outages on the Secondary Site on page 4-9
Standby database	Hardware or 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.
Standby	Hardware or software	Not applicable	Before the outage:
database	maintenance (clusterwide impact)		Preparing for Scheduled Outages on the Secondary Site on page 4-9
			After the outage:
			Restoring Fault Tolerance After Planned Downtime on Secondary Site or Clusterwide Outage on page 4-53
Standby database	Oracle patch and software upgrades	Downtime needed for upgrade, but there is no effect on primary node unless the configuration is in maximum protection database mode.	Downtime needed for upgrade, but there is no effect on primary node unless the configuration is in maximum protection database mode.

 Table 4–6
 Managing Scheduled Outages on the Secondary Site

#### 4.1.2.3 Preparing for Scheduled Outages on the Secondary Site

To achieve continued service during scheduled outages on a secondary site when in maximum protection mode, downgrade the maximum protection mode to maximum availability or maximum performance temporarily. 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 production database, which in turn lengthens the time to restore fault tolerance. See Section 2.4.2, "Data Protection Mode" on page 2-23 for an overview of the Data Guard protection modes.

# 4.2 Recovering from Unscheduled Outages

This section describes best practices for recovering from various types of unscheduled outages.

This section contains these topics:

Complete Site Failover

- Database Failover with a Standby Database
- Database Switchover with a Standby Database
- RAC Recovery for Unscheduled Outages
- Application Failover
- ASM Recovery After Disk and Storage Failures
- Recovering from Data Corruption (Data Failures)
- Recovering from Human Error

# 4.2.1 Complete Site Failover

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.

## 4.2.1.1 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

## 4.2.1.2 Best Practices for Complete Site Failover

Site failover can be expedited in minutes by using the following practices:

- Use 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
- Maintain a running middle-tier application server on the secondary site to avoid the startup time
- Automate the DNS failover procedure

Data loss is dependent on the Data Guard configuration and the use of synchronous or asynchronous redo shipping.

## 4.2.1.3 Repair Solution

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 4–1 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 a RAC instance.
- 6. Responses are sent back to the application and clients by a similar path.

Figure 4–1 Network Routes Before Site Failover



Figure 4–2 illustrates the network routes after site failover. Client or application requests enter the secondary site at the client tier and follow exactly the same path on the secondary site that they followed on the primary site.





The following steps describe the effect on network traffic of a failover or switchover:

- 1. The administrator has failed over or switched over the production 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 already running.
- **3.** The wide-area traffic manager selection of the secondary site can be automatic in the case of 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 new zone information, and the change is announced with DNS NOTIFY.

The slave DNS servers are notified of the zone update with a DNS NOTIFY announcement, and the slave DNS servers pull the new 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

Apple Mac OS X: lookupd -flushcache

- **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 of time. 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.

## 4.2.2 Database Failover with a Standby Database

*Failover* is the operation of transitioning one of the standby databases to the role of production database. A failover operation can be invoked when an unplanned failure

occurs on the production database, and there is no possibility of recovering the production database in a timely manner.

Data Guard enables you to fail over by:

- Using Oracle Enterprise Manager, as described in Section 4.2.2.2.1, "Using Enterprise Manager to Perform a Data Guard Failover" on page 4-15
- Using the Oracle Data Guard Broker command-line interface (DGMGRL)
- Issuing SQL statements, as described in Section 4.2.2.2.2, "Using SQL to Fail Over to a Physical Standby Database" on page 4-18

**See Also:** Oracle Data Guard Broker for information about using Enterprise Manager or the Data Guard broker command-line for database Failover

Data Guard failover is a series of steps to transition a standby database into a production database. The standby database essentially assumes the role of production. A Data Guard failover is accompanied by an application failover and, in some cases, preceded by a site failover. After the Data Guard failover, the secondary site contains the production database. The former production database must be reinstated as a new standby database to restore resiliency. The standby database can be quickly re-created by using Flashback Database. See Section 4.3.2, "Restoring a Standby Database After a Failover" on page 4-50.

With Data Guard the process of failover can be completely automated using fast-start failover, or the failover operation can be user driven, also referred to as manual failover. Fast-start failover eliminates the uncertainty of a process that requires manual intervention and automatically executes a zero data loss failover within seconds of an outage being detected. A manual failover allows for a failover process where decisions are user driven. Manual failover can be accomplished by using Oracle Enterprise Manager, by issuing commands at the Oracle Data Guard broker command-line interface, or by issuing the SQL statements described in subsequent sections.

A failover operation typically occurs in under a minute, and with little or no data loss. The complete description of a failover can be found in *Oracle Data Guard Concepts and Administration*.

**See Also:** The following white papers for information about optimizing the Failover operation:

 "Oracle Database 10g Release 2 Best Practices: Data Guard Fast-Start Failover" at http://www.oracle.com/technology/deploy/availabil ity/pdf/MAA\_WP\_10gR2\_ FastStartFailoverBestPractices.pdf

"Oracle Database 10g Release 2 Best Practices: Data Guard Switchover and Failover" at http://www.oracle.com/technology/deploy/availability/p df/MAA\_WP\_10gR2\_SwitchoverFailoverBestPractices.pdf

#### 4.2.2.1 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, Data Guard should be used.

A manual failover, which is user initiated, should be performed due to an unplanned outage such as:

- Site disaster which results in the primary database becoming unavailable
- User errors that cannot be repaired in a timely fashion
- Data failures, which impact the production application

A failover requires that the initial production database be reinstated as a standby database to restore fault tolerance to your environment. The standby database can be quickly reinstated by using Flashback Database. See Section 4.3.2, "Restoring a Standby Database After a Failover" on page 4-50.

#### 4.2.2.2 Best Practices for Implementing Data Guard Failover

A fast-start failover is completely automated and requires no user intervention. A manual failover, being user-driven, can be performed using Enterprise Manager, the Data Guard broker command-line interface, or SQL\*Plus commands:

- Fast-start failover: There are no procedural best practices to consider when performing a fast-start failover. However, it is very important to address all of the configuration best practices described in Section 2.4.7.2.2, "Fast-Start Failover Best Practices" on page 2-40.
- Manual failover: When performing a manual failover, follow the best practices described in Section 4.2.2.2.2, "Using SQL to Fail Over to a Physical Standby Database" on page 4-18 and the configuration best practices outlined in Section 2.4.7.2.3, "Manual Failover Best Practices" on page 2-41:

For manual failovers that involve Real Application Clusters, issue the SHUTDOWN ABORT statement on all secondary RAC instances on the standby database prior to performing a failover.

See Also: Oracle Database 10g Release 2 Best Practices: Data Guard Switchover and Failover at http://www.oracle.com/technology/deploy/availability /htdocs/maa.htm

This section contains these topics:

- Using Enterprise Manager to Perform a Data Guard Failover
- Using SQL to Fail Over to a Physical Standby Database
- Using SQL to Fail Over to a Logical Standby Database

**4.2.2.2.1** Using Enterprise Manager to Perform a Data Guard Failover The procedure for Data Guard failover is the same for both physical and logical standby databases. The following screen shots illustrate how to perform a failover using Oracle Enterprise Manager:

In Figure 4–3 the Data Guard Overview page shows the ORA-16625 error status that indicates problems accessing the primary database.

DRACLE Enterprise Manager 10g	Setup Preferences Help Logout
	Home Targets Deployments Alerts Policies Jobs Reports
Databases   Hosts   Web Applications   Services   Systems   Gro	oups   All Targets
st: north.foo.com > Database Instance: DR_Sales >	Logged in As SY
ata Guard	
ge Refreshed May 29, 2005 2:07:09 PM EDT	View Data 🛛 Real Time: Manual Refresh 🔤 📑
Overview	Standby Progress Summary
Data Guard Status × Error Protection Mode <u>Maximum Protection</u> Fast-Start Failover <u>Disabled</u>	The transport lag is the time difference between the primary last update and the standby last receivi- redo. The apply lag is the time difference between the primary last update and the standby last appli- redo.
Primary Database	1.0
Name <u>North_Sales</u> Host <u>north</u> Data Guard Status <mark>× <u>ORA-16625:</u> cannot reach the database</mark> Current Log <u>n/a</u> Properties <u>Edit</u>	0.5 0.0 0.0 0.0 0.0 0 0 0 0 0 0 0 0 0 0
Standby Databases	E.Galo
Standby Databases	(Add Standby Database
Edit Remove Switchover Failover	
Select Name Host Data Guard Status Role	Last Received Log Last Applied Log Estimated Failover Time
OR_Sales north ✓ Normal Physical Stand	by <u>63</u> O seconds
Performance	Additional Administration
Performance Overview Log File Details	Verify Remove Data Guard Configuration

#### Figure 4–3 Data Guard Overview Page Showing ORA-16625 Error

To transition DR\_Sales into the primary role, select DR\_Sales in the Standby Databases table and click **Failover**.

Figure 4–4 shows the Failover Confirmation page.

Figure 4–4 Failover Confirmation Page

GRACLE Enterprise Manager 10g	Setup Preferences Help Loqout			
Home Targets Deployments Alerts	Policies Jobs Reports			
Databases   Hosts   Web Applications   Services   Systems   Groups   All Targets				
	Logged in As SYS			
Confirmation: Failover to DR_Sales				
A failover will cause the standby database to become the primary database. Since DR_Sales is a physical st be shutdown and restarted. The failover operation cannot be cancelled.	andby database, it will			
TIP If the original primary database is still running, it must be manually shut down for the failover to succeed. Please ensure that the original primary database is shut down before continuing.				
Select Failover Option				
C Complete All available redo will be applied on the standby database, thereby minimizing data loss. Oracle recommends this type of failove	er.			
C Immediate No additional data will be applied on the standby database; data may be lost. This is the fastest type of failover.				
Are you sure you want to failover to DR_Sales?				
	No Yes			

If you determine that a failure occurred on the primary database and there is no possibility of recovering the primary database in a timely manner, you can start the Failover operation. In configurations with both physical and logical standby databases, Oracle recommends using the physical standby database as the failover target because it will allow the logical standby database to continue to function as a logical standby to the new primary database. If the failover is made to the logical

standby, any physical standbys in the configuration will need to be re-created from a backup of the new primary database.

The failover operation enables you to choose one of the following two types of failover operations:

Complete

This operation attempts to minimize data loss by applying all available redo on the standby database.

Immediate

No additional data is applied on the standby database; data might be lost. This is the fastest type of failover.

Figure 4–5 shows the progress of the failover operation.

Figure 4–5 Failover Progress Page

ORACLE Enterprise Manager 10g		Setup Preferences Help Logout			
	Home Targets Deploymen	nts Alerts Policies Jobs Reports			
Databases   Hosts   Web Applications   Services	Systems   Groups   All Targets				
		Logged in As SYS			
🛞 Processing: Failover					
Failing over to DR Sales					
1 1 1	This process will take some time. The page will automatically forward to the overview page upon completion. Click on the alert log link to view progress details in a new browser window. View alert log: <u>North_Sales_DR_Sales</u>				
	Performing failover.	1			
-	<ul> <li>Restarting primary database.</li> <li>Waiting for failover to complete.</li> </ul>				
${\ensuremath{\overline{\textbf{O}}}}$ TIP This process cannot be cancelled. It will continue a	ven if the browser window is closed.				

During the failover, the selected standby database (also referred to as the target standby database) transitions into the primary role. If the failover target is a physical standby database, it is restarted. When completed, the Data Guard Overview page reflects the updated configuration, as shown in Figure 4–6.

#### Figure 4–6 Data Guard Overview Page After a Failover Completes

ORACLE Enterprise Ma Grid Control	anager 10 <i>g</i>					Setup Preferences	<u>Help</u> Logout
			Home	E Targets Deplo	yments Alerts	Policies Jobs	Reports
Databases   Hosts   Web	b Applications   Services   Systems	Groups   Al	l Targets				
Host: north.foo.com >	Database Instance: DR_Sales >					Logg	jed in As SYS
(i) Information							
Failover completed succ	essfully.						
Data Guard							
Page Refreshed May 29, 20	05 6:40:11 PM EDT			,	√iew Data Real Time	e: Manual Refresh	<b>×</b> 🖡
Overview		St	andby Pr	ogress Summary			
Data Guard Status Protection Mode Fast-Start Failover			apply lag is t	g is the time difference betw the time difference betweer			
Primary Database							
Name	DR Sales	0.	.5				
Host Data Guard Status	north Normal						
Current Log	n/a	0.	.0				
Properties	Edit			No data is ci	urrently available.		
Standby Databases							
Standby Databases						Add Stand	by Database )
Edit Remove Switch	hover) Failover)					<u> </u>	
Select Name Hos	st Data Guard Status	Role		Last Received Log	Last Applied Log	Estimated Failo	over Time
North Sales north	th Database must be reinstated	Physical 3	Standby	<u>n/a</u>	<u>n/a</u>	n/a	
Performance				tional Administratio	on		
<u>Performance Overview</u> Log File Details			Verify Remi	/ ove Data Guard Configu	ration		

In the figure, the Data Guard Status column indicates that the original primary database (North\_Sales) is disabled and can no longer be managed through Enterprise Manager until it has been re-enabled as a physical standby database.

**4.2.2.2 Using SQL to Fail Over to a Physical Standby Database** Follow these steps to fail over to a physical standby database:

- **1.** If the standby database is a Real Application Clusters database, then issue a SHUTDOWN ABORT on all additional standby instances.
- **2.** Initiate the failover by issuing the following SQL command on the target standby database:

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;

If the network between the primary and standby sites is unavailable, then the standby RFS processes will wait for the network connections to time out through normal TCP timeout processing before shutting down. While the RFS processes are in this TCP timeout processing, the standby database will not be able to fail over unless you include the FORCE keyword on the RECOVER MANAGED STANDBY DATABASE FINISH statement.

**3.** Convert the physical standby database to the primary role:

ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;

**4.** If the standby database was never opened read-only since the last time it was started, then open the new primary database by issuing the following SQL statement:

ALTER DATABASE OPEN;

**5.** If the standby database has been opened read-only, then restart the new primary database before starting Redo Apply.

**4.2.2.3** Using SQL to Fail Over to a Logical Standby Database Follow these steps to fail over to a logical standby database:

- **1.** If the standby database is a Real Application Clusters database, then issue a SHUTDOWN ABORT on all additional standby instances.
- **2.** Initiate the failover by issuing the following SQL command on the target standby database:

ALTER DATABASE ACTIVATE LOGICAL STANDBY DATABASE FINISH APPLY;

This statement stops the RFS process, applies any remaining redo data, stops SQL Apply, and activates the logical standby database in the primary role. To avoid waiting for the redo in the standby redo log file to be applied prior to performing the failover, omit the FINISH APPLY clause on the statement.

Although omitting the FINISH APPLY clause will accelerate failover, omitting it will also cause the loss of any unapplied redo data in the standby redo log. To gauge the amount of redo that will be lost, query the V\$LOGSTDBY\_PROGRESS view. The LATEST\_SCN column value indicates the last SCN received from the primary database, and the APPLIED\_SCN column value indicates the last SCN applied to the standby database. All SCNs between these two values will be lost.

#### 4.2.3 Database Switchover with a Standby Database

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 production database. Following a successful switchover operation, the standby database assumes the production role and the production database becomes a standby database. In a RAC environment, a switchover requires that only one instance is active for each database, production and standby. 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.

Data Guard enables you to change these roles dynamically by:

- Using Enterprise Manager, as described in Section 4.2.3.2.1, "Using Enterprise Manager to Perform a Data Guard Switchover" on page 4-20
- Using the Oracle Data Guard Broker command-line interface (DGMGRL)
- Issuing SQL commands, as described in Section 4.2.3.2.2, "Using SQL for Data Guard Switchover to a Physical Standby Database" on page 4-22 and Section 4.2.3.2.3, "Using SQL for Data Guard Switchover to a Logical Standby Database" on page 4-23

**See Also:** Oracle Data Guard Broker for information about using Enterprise Manager or the Data Guard broker command-line for database switchover

#### 4.2.3.1 When to Perform a Data Guard Switchover

*Switchover* is a planned operation. Switchover is the capability to switch database roles between the production and standby databases. Switchover can occur whenever a production database is started, the target standby database is available, and all the archived redo logs are available. Switchovers are typically completed in less than five

minutes and, in some cases, are optimized to be less than a minute. Switchovers are useful in the following situations:

- Scheduled maintenance such as hardware maintenance or firmware patches on the production host
- Resolution of data failures when the production database is still opened
- Testing and validating the secondary resources, as a means to test disaster recovery readiness

Switchover is not possible or practical under the following circumstances:

- Missing archived redo logs
- Point-in-time recovery is required
- The production database is not open and cannot be opened

#### 4.2.3.2 Best Practices for Implementing Data Guard Switchover

Before performing a switchover the following procedural best practices should be followed in addition to the configuration best practices outlined in Section 2.4.7.1.1, "Switchover Best Practices" on page 2-38:

- 1. Disconnect all sessions possible by issuing SQL command ALTER SYSTEM KILL SESSION.
- **2.** Stop job processing by setting the AQ\_TM\_PROCESSES to 0.
- **3.** Cancel any specified apply delay by using the NODELAY keyword to stop and restart log apply services on the standby database:
  - On a physical standby database:

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL; ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE NODELAY;

On a logical standby database:

ALTER DATABASE STOP LOGICAL STANDBY APPLY; ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE NODELAY;

See Also: Oracle Database 10g Release 2 Best Practices: Data Guard Switchover and Failover at http://www.oracle.com/technology/deploy/availability /htdocs/maa.htm

This following sections describe how to perform a switchover:

- Using Enterprise Manager to Perform a Data Guard Switchover
- Using SQL for Data Guard Switchover to a Physical Standby Database
- Using SQL for Data Guard Switchover to a Logical Standby Database

**4.2.3.2.1 Using Enterprise Manager to Perform a Data Guard Switchover** The procedure for Data Guard switchover using Enterprise Manager is the same for both physical and logical standby databases:

- 1. Select the standby database that you want to become the primary database.
- 2. Click Switchover.

**3.** Click **Yes** to continue with the switchover. Click **No** to cancel.

Figure 4–7 shows the switchover confirmation page.

#### Figure 4–7 Switchover Operation Confirmation

ORACLE Enterprise Manager 10g	Setup Preferences Help Logout
	Home Targets Deployments Alerts Policies Jobs Reports
Databases   Hosts   Web Applications   Services   Systems   Groups   A	ll Targets
	Logged in As SYS
Confirmation: Switchover to DR_Sales	
A switchover will cause the primary and standby databases to switch roles. Since I shut down and restarted. The switchover operation cannot be cancelled.	DR_Sales is a physical standby database, the primary and standby databases will be
Any active sessions connected to the primary database will be closed automaticall Browse Primary Database Sessions	y during the switchover operation.
${f o}$ TIP Standby databases not involved in the switchover will continue to function no	rmally after the switchover.
Are you sure you want to switchover to DR_Sales?	
	(No) (Yes)

## Figure 4–8 shows the processing page during the switchover.

#### Figure 4–8 Processing Page During Switchover

ORACLE <sup>®</sup> Enterprise Manager 10 <i>g</i>	Setup Preferences Help Logout
	Home Targets Deployments Alerts Policies Jobs Reports
Databases   Hosts   Web Applications   Services   Systems   Groups	All Targets
	Logged in As SYS
Processing: Switchover	
Switching over to DR_Sales	
This process takes some time. The page automatically returns to the Data Guard Click on the alert log link to view progress details in a new browser window. View	
<ul> <li>Performing</li> </ul>	0
➡ Restarting o Waiting for	databases switchover to complete
ØTIP This process cannot be cancelled. It will continue even if the browser wind	ow is closed.

Figure 4–9 shows the Data Guard overview page after a successful switchover.



ORACLE Enterprise Manager 10g				Setup Preferences Help Logout
			oloyments Alerts	Policies Jobs Reports
Databases   Hosts   Web Applications   Services   Syst		All Largets		
Host: north.foo.com > Database Instance: DR Sales >	>			Logged in As SYS
① Information				
Switchover completed successfully.				
Data Guard				
Page Refreshed May 29, 2005 1:34:47 PM EDT			View Data Real Ti	me: Manual Refresh 🛛 🖌 🖹
Overview	Stan	dby Progress Summ	nary	
Data Guard Status 🛛 🛩 Normal				ast update and the standby last received
Protection Mode <u>Maximum Protection</u> Fast-Start Failover Disabled	Protection Mode <u>Maximum Protection</u> redo. The apply lag is the time difference between the primary last update and the standby lag redo.			
	1.	<u>م</u>		
Primary Database		Ť		
Name <u>DR_Sales</u> Host north	spu			Transport Lag
Data Guard Status 🗸 Normal	.0 Seconds	5		Apply Lag
Current Log <u>n/a</u>	v.			
Properties <u>Edit</u>	0.	0	OOOOOOOOOOOOOOOOOOOOOOOOO	<u> </u>
		14	or (ii_bales	
Standby Databases				
				Add Standby Database
(Edit) Remove Switchover Failover				
Select Name Host Data Guard Status Role	e L	ast Received Log	Last Applied Log	Estimated Failover Time
North Sales north	rsical Standby n	ı/a	n/a	O seconds
Performance		Additional Adminis	stration	
<u>Performance Overview</u> Log File Details		Verify Remove Data Guard (	Configuration	
Log i no Dolano		Remove Data Otalu (	zonnguration	

**4.2.3.2.2 Using SQL for Data Guard Switchover to a Physical Standby Database** If you are not using Oracle Enterprise Manager, then the high-level steps in this section can be performed using with SQL\*Plus. These steps are described in detail in *Oracle Data Guard Concepts and Administration*.

Follow these steps for a switchover to a physical standby database:

- 1. If possible, disconnect user sessions and disable or stop application processing.
- **2.** If the primary and standby databases are RAC, then cleanly shut down all instances except one. To expedite this operation, issue a SHUTDOWN ABORT.
- **3.** Issue the following SQL statement on the primary database to convert it to a standby database:

ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY WITH SESSION SHUTDOWN;

- **4.** After the statement in the previous step completes:
  - **a.** Issue the following SQL statement on the old standby database:

ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;

- **b.** Immediately after issuing the COMMIT TO SWITCHOVER TO PRIMARY statement, restart the old primary database as the new standby database and bring it to the mount state.
- **5.** When the switchover command completes, issue the ALTER DATABASE OPEN statement on the new primary database to bring it to the open state.

Opening the new primary database from the mount state is possible only if the standby database was never opened read-only since the last time the database was started. If the database has been opened read-only, it will need to be restarted.

- 6. If the primary and standby databases are RAC, then start all instances.
- 7. Restart user sessions and application processing.

**4.2.3.2.3 Using SQL for Data Guard Switchover to a Logical Standby Database** If you are not using Oracle Enterprise Manager, then the high-level steps in this section can be executed with SQL\*Plus. These steps are described in detail in *Oracle Data Guard Concepts and Administration*.

When performing a switchover using SQL\*Plus commands it is possible for the old standby database that is to become the new primary database to build and transmit the LogMiner dictionary to the current primary database (the new standby database) prior to the actual switchover. This reduces the total time needed to perform the switchover. The following steps describe how to perform this optimized method:

**1.** Issue the following SQL statement on the primary database to enable receipt of redo from the current standby database:

ALTER DATABASE PREPARE TO SWITCHOVER TO LOGICAL STANDBY;

**2.** On the current logical standby database, build the LogMiner dictionary and transmit this dictionary to the current primary:

ALTER DATABASE PREPARE TO SWITCHOVER TO PRIMARY;

- **3.** If possible, disconnect user sessions and disable or stop application processing.
- **4.** If the primary and standby databases are RAC, then cleanly shut down all instances except one. To optimize the shutdown operations, use SHUTDOWN ABORT.
- **5.** When the SWITCHOVER\_STATUS column of the V\$DATABASE view returns TO LOGICAL STANDBY, convert the primary database to a standby by issuing:

ALTER DATABASE COMMIT TO SWITCHOVER TO LOGICAL STANDBY WITH SESSION SHUTDOWN;

**6.** Issue the following statement on the old standby database:

ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;

- **7.** If the primary and standby databases are configured in a RAC, then start all instances.
- 8. Restart user sessions and application processing.

#### 4.2.4 RAC Recovery for Unscheduled Outages

This solution is leveraged automatically when there is a node or instance failure. Surviving instances will automatically recover the failed instances and potentially aid in the automatic client failover. Recover times can be bounded by adopting the database and 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.

The following recovery methods can be used:

- Automatic Instance Recovery for Failed Instances
- Automatic Service Relocation
- Oracle Cluster Registry

## 4.2.4.1 Automatic Instance Recovery for Failed Instances

Instance failure occurs when software or hardware problems disable an instance. After instance failure, Oracle automatically uses the online redo log file to perform database recovery as described in this section.

Instance recovery in RAC does not include restarting the failed instance or the recovery of applications that were running on the failed instance. Applications that were running continue by using service relocation and fast application notification (as described in Section 4.2.4.2, "Automatic Service Relocation" on page 4-24).

When one instance performs recovery for another instance, the recovering instance:

- 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 node failures occur, as long as one instance survives, RAC performs instance recovery for any other instances that fail. If all instances of a RAC database fail, then on subsequent restart of any one instance, crash recovery will occur and all committed transactions will be recovered. If Data Guard is available, you can fail over automatically with Data Guard fast-start failover once all instances are down.

## 4.2.4.2 Automatic Service Relocation

Service reliability is achieved by configuring and failing over among redundant instances. More instances are enabled to provide a service than would otherwise be needed. If a hardware failure occurs and adversely affects a RAC database instance, then CRS automatically moves any services on that instance to another available instance, as configured with DBCA or Enterprise Manager. Then, Cluster Ready Services (CRS) attempt to restart the failed nodes and instances.

CRS recognizes when a failure affects a service and automatically fails over the service and redistributes the clients across the surviving instance supporting the service. In parallel, CRS attempts to restart and integrate the failed instances and dependent resources back into the system. Notification of failures using fast application notification (FAN) events occur at various levels within the Oracle Server architecture. The response can include notifying external parties through Oracle Notification Service (ONS), advanced queueing, or FAN callouts, recording the fault for tracking, event logging, and interrupting applications. Notification occurs from a surviving node when the failed node is out of service. The location and number of nodes serving a service is transparent to applications. Auto restart and recovery are automatic, including all the subsystems, such as the listener and the ASM instance, not just database.

# 4.2.4.3 Oracle Cluster Registry

Loss of the Oracle Cluster Registry file affects the availability of RAC and Cluster Ready Services. The OCR file can be restored from a physical backup that is automatically created or from an export file that is manually created by using the ocrconfig tool. Additionally, starting with Oracle Database 10g Release 10.2, Oracle can optionally mirror the OCR so that a single OCR device failure can be tolerated.

**See Also:** "Administering Storage in Real Application Clusters" in *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* 

# 4.2.5 Application Failover

With proper configuration, applications can be configured to receive fast and efficient notification when application services become unavailable. Once notified, application connections occur transparently to surviving resources such as nodes in a RAC database or a standby database in a remote datacenter than has assumed the primary role following a failover.

In a RAC configuration, services are essential to achieving fast and transparent application failover. If a service becomes unavailable for a particular instance, the service will fail over to an available instance in the cluster, thereby allowing applications to continue processing. Clients are notified of the service relocation through Fast Application Notification (FAN).

Services are also essential for client failover across sites in a Data Guard configuration. After a site failure in a Data Guard configuration, the new production database will also be configured to automatically publish the production service while notifying affected clients that the services are no longer available on the failed production database through FAN events.

FAN notifications and service relocation enable automatic and fast redirection of clients in the event of a failures for both RAC and Data Guard environments.

## See Also:

- Section 2.4.7.2, "Role Transition During Failover" on page 2-38
- Oracle Database 10g Release 2 Best Practices: Client Failover for Highly Available Oracle Databases at http://www.oracle.com/technology/deploy/availabil ity/htdocs/maa.htm

# 4.2.6 ASM Recovery After Disk and Storage Failures

The impacts and recommended repairs for various ASM failure types are summarized in Table 4–7.

Failure	Description	Impact	Recommended Repair
ASM instance failure	ASM instance fails	All database instances accessing ASM storage from the same node will shut down.	Automatic RAC Recovery for Unscheduled Outages on page 4-23
			If RAC is not used, use Data Guard failover (see Section 4.2.2.2, "Best Practices for Implementing Data Guard Failover" on page 4-15)
			If RAC and Data Guard are not used, fix the underlying problem and then restart ASM and the database instances
ASM disk failure	One or more ASM disks fail, but all disk groups remain online.	All data remains accessible. This is possible only with normal or high redundancy disk groups.	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
			<b>Note:</b> External redundancy disk groups should use mirroring in the storage array to protect from disk failure. Disk failures should not be exposed to ASM.
Data area disk-group failure	One or more ASM disks fail, and data area disk group goes offline.	Databases accessing the data area disk group will shut down.	Perform Data Guard failover or local recovery as described in Section 4.2.6.3, "Data Area Disk Group Failure" on page 4-29
Flash recovery area disk-group failure	One or more ASM disks fail, and the flash recovery area disk group goes offline.	Databases accessing the flash recovery area disk group will shut down.	Perform local recovery or Data Guard failover as described in Section 4.2.6.4, "Flash Recovery Area Disk Group Failure" on page 4-31

Table 4–7 Types of ASM Failures and Recommended Repair

#### 4.2.6.1 ASM Instance Failure

If the ASM instance fails, database instances accessing ASM storage from the same node will shut down.

If the primary database is using RAC, then application failover will occur automatically and clients connected to the database instance will reconnect to remaining instances providing the required service in the cluster and continue processing. The recovery time is typically in seconds.

If the primary database does not use RAC, then an ASM instance failure will shut down the entire database. If Data Guard is being used and Data Guard fast-start failover is configured, a database failover will be triggered automatically and clients will automatically reconnect to the new primary database following the failover. The recovery time is the time it takes to complete an automatic Data Guard fast-start failover operation. If fast-start failover is not configured, then recovering from this outage is a manual process, which can be accomplished by either restarting the ASM and database instances manually, or by performing a Data Guard failover.

If neither RAC nor Data Guard is being used, then restart the ASM instance and restart database instances manually. The recovery time depends on the length of time to start

the ASM instance, and the length of time to start the database instances and perform crash recovery.

#### 4.2.6.2 ASM Disk Failure

If an ASM disk group is configured as an external redundancy type, then a failure of a single disk should be handled by the storage array and should not be seen by the ASM instance, and all operations of ASM and databases using the disk group will continue normally. However, if the failure of an external redundancy disk group is seen by the ASM instance, then the ASM instance will take the disk group offline immediately, causing Oracle instances accessing the disk group to crash. If the disk failure is temporary, then ASM and the database instances can be restarted, and crash recovery will occur after the disk group is brought back online.

If an ASM disk group is configured as a normal or a high-redundancy type, then disk failure is handled transparently by ASM and the databases accessing the disk group are not affected. An ASM instance automatically starts an ASM rebalance operation to distribute the data on one or more failed disks to alternative disks in the 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 re-mirrored. When the rebalance operation completes successfully, the 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 ASM disk group.

The failure of multiple disks in multiple failure groups where a primary extent and all of its mirror extents have been lost will cause the disk group to go offline.

**See Also:** Section 4.2.6.3, "Data Area Disk Group Failure" on page 4-29 and Section 4.2.6.4, "Flash Recovery Area Disk Group Failure" on page 4-31 for details

This following recovery methods can be used:

- Using Enterprise Manager to Repair ASM Disk Failure
- Using SQL to Add Replacement Disks Back to the Disk Group

#### 4.2.6.2.1 Using Enterprise Manager to Repair ASM Disk Failure

Figure 4–10 shows Enterprise Manager reporting disk failures. Three alerts appear at 11:19:29. The first alert is an Offline Disk Count. The second and third alerts are Disk Status messages for data area disk DATA.XBBT1D06\_DATA and recovery area disk RECO.XBBT1D06\_RECO:

2 disks are offline Disk DATA.XBBT1D06\_DATA is offline. Disk RECO.XBBT1D06\_RECO is offline.
• 📥 - 4	🔁 💿 👫 🗑 http			orage Management: +ASM_halinux emap?type=osm_instance⌖=%2		r hendrixsonsfurniture
k				ance▼ Web Mail ▼ Personal WebSite		
RACLE E	nterprise Manager 10 <i>g</i>			Home Targets	Se	tup Preferences Help Logout icies Jobs Reports
		rs   Web Applications   Servi	ces   Systems   Groups	All Targets   Collaboration Suites		Langed in An OV
	us.oracle.com >	t: +ASM_halinux01.us.o	vracle com			Logged in As SY
Home	Performance Administ		72010.0011			
	<u>remained</u> <u>reamined</u>	<u>oomgaration</u>				
					Data Retrieved March 7, 2	2006 11:19:55 AM PST (Refresh
General				Disk Group Usage (GB)		
$\wedge$			Startup/Shutdown			
	ent Status Up					
	Up Since Feb 1, 2006 11 ability (%) 99.93	1:34:20 AM PST		RECO		
	(Last 24 hours)					
Instan	version 10.2.0.2.0					
	Host halinux01.us.or	racle.com		DATA		
	Alert Log Mar 7, 2006 11	oracle/product/10.2.0/asm				
	Alen Log Mar 7, 2006 11	1: 12: 13 AIVI		0.00 200.00 400.00 Size		
				Free Inte		
				Free 🔲 Inte	rnal 🔲 tpcc	
erviced Da	atabases			Free 🔲 Inte		
	Disk Groups			Space Used (GB)		Alert
lame					rnal 📕 tpcc	
lame xcc	Disk Groups			Space Used (GB)	Availability	
lame xcc lerts	Disk Groups DATA, RECO	Nama	Masanaa	Space Used (GB)	rnal ∎ tpcc Availability ∱	1
ame xcc lerts Severity ⊽	Disk Groups DATA, RECO 7 Category	Name Disk Mode Status	Message Disk DATA XBI	Space Used (GB) 1,097.47	mal ■ tpcc Availability ∱ Ale	1 rt Triggered
lame xcc lerts	Disk Groups DATA, RECO	Name Disk Mode Status Disk Mode Status	Disk DATA.XB	Space Used (GB)	mal ∎ tpcc Availability ☆ Ate Mat	1
lame <u>xcc</u> lerts Severity ⊽ ×	Disk Groups DATA, RECO Zategory Disk Status	Disk Mode Status	Disk DATA.XB	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_REC0 is offline.	mai tpcc Availability © Ale Mai	1 rt Triggered 7, 2006 11:19:29 AM
lame xcc lerts Severity ⊽ × ×	Disk Groups DATA, RECO Category Disk Status Disk Status	Disk Mode Status Disk Mode Status	Disk DATA.XBB Disk RECO.XB 2 disks are offli	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_REC0 is offline.	rnal ■ tpcc Availability ∱ Ale Maa Maa Maa Maa	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame booc Severity ⊽ × × × × (1)	Disk Groups           DATA, RECO           Zetegory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status	Disk Mode Status Disk Mode Status Offline Disk Count	Disk DATA.XBB Disk RECO.XB 2 disks are offli	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne.	rnal ■ tpcc Availability ∱ Ale Maa Maa Maa Maa	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame occ lerts Severity ⊽ × × × ×	Disk Groups           DATA, RECO           Zategory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta	Disk DATA.XBB Disk RECO.XB 2 disks are offli atus 1 distinct types	Space Used (GB) 1,097.47 3T1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th	rnal ∎ tpcc Availability	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame occ lerts Severity ⊽ × × × ×	Disk Groups DATA, RECO Category Disk Status Disk Status Offline Disk Count Alert Log Error Status Severity	Disk Mode Status Disk Mode Status Offline Disk Count	Disk DATA.XBB Disk RECO.XB 2 disks are offli	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne.	rnal ■ tpcc Availability ∱ Ale Maa Maa Maa Maa	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame occ lerts Severity ⊽ × × × ×	Disk Groups           DATA, RECO           Zategory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta	Disk DATA.XBB Disk RECO.XB 2 disks are offli atus 1 distinct types	Space Used (GB) 1,097.47 3T1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th	rnal ∎ tpcc Availability	7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
Name pcc Severity ▽ × × (1) lost Alerts	Disk Groups           DATA, RECO           Zategory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           Severity           (No alerts)	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta	Disk DATA.XBB Disk RECO.XB 2 disks are offli atus 1 distinct types	Space Used (GB) 1,097.47 3T1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th	rnal ∎ tpcc Availability	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
× × (1) lost Alerts Related Lin	Disk Groups           DATA, RECO           Zategory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           Severity           (No alerts)	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta Category Aler	Disk DATA XBE Disk RECO XB 2 disks are offlii 1 distinct types Name	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th Message All	Availability    Availability	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame <u>Doc</u> Severity ▽ × × (1) slost Alerts slelated Lin	Disk Groups           DATA, RECO           Zategory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           Severity           (No alerts)	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta Category Aler	Disk DATA XB8 Disk RECO.XB 2 disks are offlii atus 1 distinct types Name	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th Message All Message	mal tpcc	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame 2000 Llerts Severity ⊽ × × × I lost Alerts Line Ccess Lackouts	Disk Groups           DATA, RECO           Zategory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           Severity           (No alerts)	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta Category Aler Met	Disk DATA XB8 Disk RECO.XB 2 disks are offlii atus 1 distinct types Name	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th Message All Message	Availability  Availability  Alector  Al	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame DCC Severity ▽ × × (1) ost Alerts elated Lin CCESS lackouts	Disk Groups           DATA, RECO           Zatagory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           Severity           (No alerts)	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta Category Aler Met Rep	Disk DATA XB8 Disk RECO.XB 2 disks are offlii atus 1 distinct types Name	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th Message All Message	Availability  Availability  Alector  Al	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
iame 2020 Jerts Severity × × (1) lost Alerts Lelated Lin Ccess lackouts Lonitoring Cc	Disk Groups           DATA, RECO           Zetegory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           S           Severity (No alerts)           nks	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta Category Aler Met Rep	Disk DATA XB8 Disk RECO.XB 2 disks are offlii atus 1 distinct types Name	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th Message All Message	Availability  Availability  Alector  Al	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM
lame xcc lerts Severity ⊽/ × × x (1) oost Alerts elated Lin ccess onitoring Cc	Disk Groups           DATA, RECO           Zetegory           Disk Status           Disk Status           Offline Disk Count           Alert Log Error Status           S           Severity (No alerts)           nks	Disk Mode Status Disk Mode Status Offline Disk Count Generic Alert Log Error Sta Category Aler Meti Rep tration Configuration	Disk DATA XBE Disk RECO.XB 2 disks are offlii atus 1 distinct types Name t Log Content ric and Policy Settings orts	Space Used (GB) 1,097.47 BT1D06_DATA is offline. BT1D06_RECO is offline. ne. of ORA- errors have been found in th Message All Message	mal ■ tpcc Availability	rt Triggered 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM 7, 2006 11:19:29 AM

## Figure 4–10 Enterprise Manager Reports Disk Failures

**Figure 4–11** shows Enterprise Manager reporting the status of data area disk group DATA and recovery area disk group RECO. The red arrows under **Member Disks** indicate that one disk has failed in each disk group. The numbers under **Pending Operations** indicate that one operation is pending for each disk group.

00	A1		N) – Automatic Storage Mana				C
e• 🌪 😥 🥝 (	https://ha-g	c.us.oracle.com/em/console/data	abase/osm/osmAdmin?type=o	sm_instance⌖=%2	BASM_halinux01.us	🗇 🔻 🔘 🤇 🖸 hendr	ixsonsfurniture
etting Started Latest Headlin	es 🔊 Oracle WebSites	s * BUG WebIV iSupport C	urrent Status Finance Vel	Mail      Personal WebSi	tes 🔻		
DRACLE Enterprise Mar rid Control	ager 10g			Home Targe	ts Deployment:		Preferences Help Logout
Hosts   Databases   Appli	cation Servers   Wet	Applications   Services   Sys	sterns   Groups   All Target	Collaboration Suites			ין במנטייינס יייייס א
st: halinux01.us.oracle.com >							Logged in As SY
		A_halinux01.us.oracle.co	om				
		A_halinux01.us.oracle.co	m				52.50
utomatic Storage Ma Home Performance	nagement: +ASM		m				
utomatic Storage Ma Home Performance Disk Groups	nagement: +ASM		om				
Utomatic Storage Ma Home Performance Disk Groups Undicates offline disks	Administration	Configuration		means that redundance	v can be property.		
Utomatic Storage Ma Home Performance Disk Groups Undicates offline disks	Administration			o means that redundanc	y can be properly	(Create) (M	ount All ) (Dismount Al
Utomatic Storage Ma           Home         Performance           Disk Groups         Indicates offline disks           Ø TIP The usable free space restored after a disk	Administration	Configuration		o means that redundanc	y can be properly	Create) (M	ount All ) (Dismount Al
Utomatic Storage Ma           Home         Performance           Disk Groups         Indicates offline disks           Ø TIP The usable free space restored after a disk	Administration	Configuration		o means that redundanc	y can be properly Used (%)	(Create) (M	ount All) (Dismount Al
Home         Performance           Disk Groups ↓ Indicates offline disks	Administration Administration ce specifies the amoun failure. Dunt Co Redundance	Configuration	ed for data. A value above zer	o means that redundanc			
Liomatic Storage Ma Home Performance Disk Groups Undicates offline disks TIP The usable free span restored after a disk Delete Actions MM Select Name State	Administration Administration failure. Unit (Go) Redundance NTED NORMAL	Configuration It of space that can be safely us Usable Free (GB)	ed for data. A value above zer Size (GB) Used (GB)		Used (%)	Member Disks	

#### Figure 4–11 Enterprise Manager Reports ASM Disk Groups Status

Home | Targets | Deployments | Alerts | Policies | Jobs | Reports | Setup | Preferences | Help | Logout

Copyright D 1996, 2005, Oracle. All rights reserved. Oracle. JO Edwards, PeopleSoft, and Retek rar registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. About Oracle. Enterprise Manager

Figure 4–12 shows Enterprise Manager reporting a pending REBAL operation on the DATA disk group. The operation is about one-third done, as shown in % **Complete**, and the **Remaining Time** is estimated to be 16 minutes.

#### Figure 4–12 Enterprise Manager Reports Pending REBAL Operation

$\Theta \Theta \Theta$			Oracle Enterprise Manage	er (SYSMAN) - Pending Operations: DATA		e
🔶 · 🧼 · 🌌	🙆 🟠 💽	https://ha-gc.us.oracle.co	m/em/console/database/osm	n/osmAdmin?pageName=%2Fdatabase%2Fosm%2F	FviewOperatic 🗂 🔻 🔘	G + hendrixsonsfurniture
Getting Started Late	st Headlines 为 🛛 Or	acle WebSites 🔻 BUG 🛛 W	ebIV iSupport Current Stat	tus Finance ▼ Web Mail ▼ Personal WebSites ▼		
ORACLE Enterp	rise Manager 10g	1		Home Targets	Deployments Alerts	Setup Preferences Help Logout Policies Jobs Reports
107022/2011 7011 Store (2010)	10 10 10 10 10 1 10 1		<ul> <li>A DEPARTMENT OF THE PROPERTY AND ADDRESS OF THE PROPERTY ADDRESS</li></ul>	AND AND A REAL PROPERTY AND A R		and the second
Hosts Database	s Application Se	rvers   Web Applications	Services   Systems   G	Groups   All Targets   Collaboration Suites		
the second second second second second		and the second	Services   Systems   G			Logged in As SY
Host: halinux01.us.ora	cle.com > Automat	and the second				Logged in As SY
the second second second second second second	cle.com > Automat	and the second			PST Refresh Real Time	Logged in As SY
Host: halinux01.us.ora	cle.com > Automat	and the second		2	PST Refresh Real Time	

Copyright © 1998, 2005, Oracle. All rights reserved. Oracle. JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. <u>About Oracle : Enterprise Manager</u>

**4.2.6.2.2** Using SQL to Add Replacement Disks Back to the Disk Group Perform these steps after one or more failed disks have been replaced, and access to the storage has been restored:

**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;

## 4.2.6.3 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 ASM. However, multiple disk failures in a storage array may be seen by 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, but the ASM disk group is accessible, there is no loss of data and no immediate loss of accessibility. An ASM instance automatically starts an ASM rebalance operation to distribute the data on the one or more failed disks to alternative disks in the ASM disk group. When the rebalance operation completes successfully, the ASM disk group is no longer at risk in the event of a second failure. There must be enough disk space on the remaining disks in the disk group for the rebalance to complete successfully.

The two possible solutions when the data area disk group fails are summarized in Table 4–8.

Recovery Option	Recovery Time Objective (RTO)	Recovery Point Objective (RPO)
Data Guard failover	Five minutes or less	Varies depending on the data protection level chosen
Local recovery	Database restore and recovery time	Zero

Table 4–8 Recovery Options for Data Area Disk Group Failure

If Data Guard is being used and fast-start failover is configured, then an automatic failover will occur 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**) will be expressed in terms of minutes or perhaps 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 on the standby site, then data loss might result.

After Data Guard failover has completed, and the application is available, the data area disk group failure must still be resolved. Continue with the following "Local Recovery Steps" procedure to resolve the ASM disk group failure.

The RTO for local recovery only is based primarily 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 log members that reside in the flash recovery area, so complete media recovery is possible.

If Data Guard is not being used, 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 reintroduce the database into the Data Guard environment. Also, see Section 4.3.2, "Restoring a Standby Database After a Failover" on page 4-50.

1. Rebuild the ASM disk group using the new storage location:

SQL> CREATE DISKGROUP DATA NORMAL REDUNDANCY DISK 'path1', 'path2', ...;

**2.** Start the instance NOMOUNT:

RMAN> STARTUP FORCE NOMOUNT;

- 3. Restore the control file from the surviving copy located in the recovery area: RMAN> RESTORE CONTROLFILE FROM 'recovery\_area\_controlfile';
- **4.** Start the instance MOUNT:

RMAN> STARTUP FORCE MOUNT;

**5.** Restore the database:

RMAN> RESTORE DATABASE

**6.** Recover the database:

RMAN> RECOVER DATABASE;

**7.** If you use block change tracking, then disable and reenable the block change tracking file:

SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING; SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

**8.** Open the database:

SQL> ALTER DATABASE OPEN;

**9.** Re-create the log file members on the failed ASM disk group:

SQL> ALTER DATABASE DROP LOGFILE MEMBER '*filename*'; SQL> ALTER DATABASE ADD LOGFILE MEMBER '*disk\_group*' TO GROUP *group\_no*;

**10.** Perform a new incremental level 0 backup:

RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;

### 4.2.6.4 Flash Recovery Area Disk Group Failure

When the flash recovery-area disk group fails, the database crashes because the control file member usually resides in the flash recovery area and Oracle requires that all control file members are accessible. The flash recovery area can also contain the flashback logs, redo log members and all backups.

A flash recovery area disk group failure should occur only when there have been multiple failures. For example, if the flash recovery-area disk group is defined as external redundancy, a single-disk failure should not be exposed to ASM. However, multiple-disk failures in a storage array may be seen by 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.

Table 4–9 summarizes the two possible solutions when the flash recovery-area disk group fails.

	, ,	
<b>Recovery Option</b>	Recovery Time Objective (RTO)	Recovery Point Objective (RPO)
Local Recovery	Five minutes or less	Zero
Data Guard Failover or Switchover	Five minutes or less	Zero

Table 4–9 Recovery Options for Flash Recovery Area Disk Group Failure

Because the loss affects only the flash 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 local restart is to startup the primary database after removing the controlfile member located in the failed flash recovery area and point to a new flash recovery area for local archiving (see "Local Restart Steps" discussion later in this section for the step-by-step procedure). However, this is a temporary fix until a new flash recovery area is created to replace the failed storage components. Oracle recommends using the "Local Recovery Steps" discussion later in this section.

If you decide to perform a Data Guard failover, then the RTO will be expressed in terms of minutes or perhaps seconds depending on the presence of the Data Guard observer process and fast-start failover. After Data Guard failover has completed, and the application is available, the flash recovery area disk group failure must still be resolved. Continue with the instructions in the following "Local Recovery Steps" section to resolve the ASM disk group failure.

If the protection level is maximum performance or the standby database is *unsynchronized* with the primary database, then temporarily start up the primary database by removing the controlfile member and pointing to a temporary flash recovery area (file system) in the SPFILE. Issue a Data Guard switchover to ensure no data loss. After Data Guard switchover has completed, and the application is available, the flash recovery area disk group failure must still be resolved. Shut down the affected database and continue with the instructions in the following "Local Recovery Steps" section to resolve the ASM disk group failure.

The RTO for local recovery only is based primarily on the time to repair and replace the failed storage components and then on the time to restore the control-file copy. Because the loss affects only the flash 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. As mentioned previously, you can start up the primary database by removing the controlfile member and pointing to a new flash recovery area. However, this is a temporary fix filled with availability and performance risks unless the flash recovery area is configured properly. Therefore, Oracle recommends the "Local Recovery Steps" that follow.

#### Local Restart Steps

For a fast local restart, perform the following steps on the primary database:

1. Change the CONTROL\_FILES initialization parameter to refer only to members in the Data Area. For example:

ALTER SYSTEM SET CONTROL\_FILES='+DATA/sales/control1.dbf' SCOPE=spfile;

**2.** Change local archive destinations and/or the flash recovery area to the local redundant, scalable destination. For example:

ALTER SYSTEM SET DB\_RECOVERY\_FILE\_DEST='+DATA' SCOPE=spfile;

**3.** Startup with new settings:

STARTUP MOUNT:

You may need to disable and reenable Flashback Database because the flashback logs were damaged or lost:

ALTER DATABASE FLASHBACK OFF; ALTER DATABASE FLASHBACK ON; ALTER DATABASE OPEN;

#### 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 old 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 reinstantiation of the standby database.

- 1. Replace or get access to new storage to be leveraged as flash recovery area
- 2. Rebuild the ASM disk group using the new storage location:

SQL> CREATE DISKGROUP RECO NORMAL REDUNDANCY DISK 'path1', 'path2',...;

**3.** Start the instance NOMOUNT:

RMAN> STARTUP FORCE NOMOUNT;

- Restore the control file from the surviving copy located in the data area:
   RMAN> RESTORE CONTROLFILE FROM 'data\_area\_controlfile';
- **5.** Start the instance MOUNT:

RMAN> STARTUP FORCE MOUNT;

6. If you use Flashback Database, then disable it:

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 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 flash recovery area:

```
RMAN> CATALOG RECOVERY AREA;
RMAN> CROSSCHECK ARCHIVELOG ALL;
RMAN> CROSSCHECK BACKUPSET;
RMAN> CROSSCHECK DATAFILECOPY ALL;
RMAN> LIST EXPIRED type;
RMAN> DELETE EXPIRED type;
```

**11.** Assuming that data has been lost in some way, take a new backup:

```
RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

## 4.2.7 Recovering from Data Corruption (Data Failures)

Recovering from data corruption is an unscheduled outage scenario. Data corruption is usually—but not always—caused by some activity or failure that occurs outside the database, even though the problem might be evident within the database.

Data corruption in data files has two categories:

Data file block corruption

A corrupt data file block can be accessed, but the contents in the block are invalid or inconsistent. The typical cause of data file corruption is a faulty hardware or software component in the I/O stack, which includes, but is not limited to, the file system, volume manager, device driver, host bus adapter, storage controller, and disk drive.

The database usually remains available when corrupt blocks have been detected, but some corrupt blocks might cause widespread problems, such as corruption in a file header or with a data dictionary object, or corruption in a critical table that renders an application unusable.

A data fault is detected when it is recognized by the user, administrator, RMAN backup, or application because it has affected the availability of the application. For example:

- A single corrupt data block in a user table that cannot be read by the application because of a bad spot of the physical disk
- A single corrupt data block because of block inconsistencies detected by Oracle. The block will be marked corrupted and any application accessing the block will receive an ORA-1578 error.
- A database that automatically shuts down because of the invalid blocks of a data file in the SYSTEM tablespace caused by a failing disk controller
- Media failure

This category of data corruption results from a physical hardware problem or user error. The system cannot successfully read or write to a file that is necessary to operate the database.

In all environments, you can resolve a data corruption outage by one of the following methods:

- RMAN block media restoration and recovery
- Data Guard switchover or failover to a standby database
- RMAN datafile media restoration and recovery

Manually re-create the object

RMAN block media restoration and recovery provides the highest application availability if targeted blocks are not critical to application functionality. Data Guard switchover or failover to standby database provides the fastest predictable RTO.

Other outages that result in database objects becoming unavailable or inconsistent are caused by human error, such as dropping a table or erroneously updating table data. Information about recovering from human error can be found in Section 4.2.8, "Recovering from Human Error" on page 4-37.

If the data corruption impacts nondata files, then the repair may be slightly different. Table 4–10 provides a matrix of the key non database object corruption and the recommended repair.

Object or Component		
Affected	Impact	Repair
Any control file	Database fails	Data Guard fast-start failover will automatically fail over to the standby database
Redo log member	None	1. Investigate failure and check system
		2. Drop and re-create redo log member
Active log group that is archived and not needed for crash recovery	Database fails	Restart database after dropping affected redo log group
Active redo log group not archived	Database fails	1. Restart database after dropping affected log group
and not needed for crash recovery		2. Create a new backup
clustriccovery		<ol> <li>Refresh the standby database either by applying an incremental backup or re-creating the standby database from the primary or a backup of the primary database</li> </ol>
Active or current	Database fails	Use one of the following solutions:
redo log group that is still needed		<ul> <li>Data Guard failover</li> </ul>
for crash recovery		<ul> <li>Flashback Database—flash the database back to a consistent time and then issue an OPEN RESETLOGS</li> </ul>
Archived redo log	None	1. Create database backup
file		2. Refresh the standby database either by applying an incremental backup or re-creating the standby database from the primary or a backup of the primary database
SPFILE	None	Restore SPFILE from a backup and revise

Table 4–10 Non Database Object Corruption and Recommended Repair

The following recovery methods can be used:

- Use Data Guard to Recover From Data Corruption and Data Failure
- Use RMAN Block Media Recovery
- Use RMAN Data File Media Recovery
- Re-Create Objects Manually

## 4.2.7.1 Use Data Guard to Recover From Data Corruption and Data Failure

Failover is the operation of transitioning the standby databases as the new production database. A database switchover is a planned transition in which a standby database and a production database switch roles. Either of these operations can occur in less than 5 minutes and with no data loss.

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 will be longer than the business SLA or RTO.

**See Also:** Database Switchover with a Standby Database on page 4-19 and Database Failover with a Standby Database on page 4-13

## 4.2.7.2 Use RMAN Block Media Recovery

Block media recovery (BMR) recovers one block or a set of data blocks marked "media corrupt" in a data file by using the RMAN BLOCKRECOVER 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. This results in lower RTO because only blocks that need recovery are restored and only necessary corrupt blocks undergo recovery. Block media recovery minimizes redo application time and avoids I/O overhead during recovery. It 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 the blocks that need recovery are known. If a significant portion of the datafile is corrupt, or if the amount of corruption is unknown, then a different recovery method should be used.
- Blocks are marked corrupt (verified with the RMAN BACKUP VALIDATE command) and only when complete recovery is required.
- Backup of the data file containing the corrupted blocks is available locally or can be retrieved from a remote location including from a a physical standby database.

Block media recovery cannot be used to recover from the following:

- User error or software bugs that cause logical corruption where the data blocks are intact. See Section 4.2.8, "Recovering from Human Error" on page 4-37 for additional details for this type of recovery.
- Changes caused by corrupt redo data. Block media recovery requires that all available redo data be applied to the blocks being recovered.

For example, to recover a specific corrupt block using RMAN block media recovery: RMAN> BLOCKRECOVER DATAFILE 7 BLOCK 3;

When the corruption is detected, it would be easy to recover this block through Grid Control.

**See Also:** Oracle Database Backup and Recovery Advanced User's Guide and the white paper titled *Using Recovery Manager with Oracle Data Guard in Oracle Database 10g* available at http://www.oracle.com/technology/deploy/availabili ty/pdf/RMAN\_DataGuard\_10g\_wp.pdf

#### 4.2.7.3 Use RMAN Data File Media Recovery

Data file media recovery recovers an entire datafile 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, the affected data files must be restored and recovered.

Use RMAN file media recovery when the following conditions are true:

- The number of blocks requiring recovery is large or unknown
- Block media recovery is not available (for example, if incomplete recovery is required, or if only incremental backups are available for the data file requiring recovery)

See Also: "Advanced User-Managed Recovery Scenarios" in Oracle Database Backup and Recovery Advanced User's Guide and the white paper titled Using Recovery Manager with Oracle Data Guard in Oracle Database 10g available at http://www.oracle.com/technology/deploy/availabili ty/pdf/RMAN\_DataGuard\_10g\_wp.pdf

#### 4.2.7.4 Re-Create Objects Manually

Some database objects, such as small look-up tables or indexes, can be recovered quickly by manually re-creating the object instead of doing media recovery.

Use manual object re-creation when:

- You must re-create a small index because of media corruption. Creating an index online enables the base object to be used concurrently.
- You must re-create a look-up table or when the scripts to re-create the table are readily available. Dropping and re-creating the table might be the fastest option.

## 4.2.8 Recovering from Human Error

Oracle flashback technology revolutionizes data recovery. In the past it took seconds to damage a database but 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 4.2.7, "Recovering from Data Corruption (Data Failures)" on page 4-34 and Section 4.2.2, "Database Failover with a Standby Database" on page 4-13 to repair these outages.

Table 4–23 summarizes the flashback solutions for each type of outage.

 Table 4–11
 Flashback Solutions for Different Outages

Impact of Outage	Examples of Human Errors	Flashback Solutions
Row or transaction	Accidental deletion of row	Flashback Query
See Also: "Resolving Row and	Erroneous transaction	Flashback Version Query
Transaction Inconsistencies" on page 4-39		Flashback Transaction Query
Table	Dropped table	Flashback Drop
See Also:"Resolving Table Inconsistencies" on page 4-39	Erroneous transactions affecting one table or a set of tables	Flashback Table
Tablespace or database	Erroneous batch job affecting many	Flashback Database
See Also: "Resolving	tables or an unknown set of tables	
Database-Wide Inconsistencies" on page 4-42	Series of database-wide malicious transactions	

Table 4–12 summarizes each flashback feature.

Table 4–12	Summary of Flashback Features
------------	-------------------------------

Flashback Feature	Description
Flashback Query	Flashback Query enables you to view data at a point in time in the past. It can be used 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.
	Note: Changes are propagated to 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.
	Note: Changes are propagated to 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.
	Note: Changes are propagated to physical and logical standby databases.
Flashback Drop	Flashback Drop provides a way to restore accidentally dropped tables.
	Note: Changes are propagated to physical standby databases.
Flashback Table	Flashback Table enables you to quickly recover a table to a point in time in the past without restoring a backup.
	Note: Changes are propagated to 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.

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. See the configuration best practices for the database—as documented in Section 2.2, "Configuring Oracle Database 10g" on page 2-7—for

configuring flashback technologies to ensure that the resources from these solutions are available at a time of failure.

**See Also:** Oracle Database Administrator's Guide, Oracle Database Backup and Recovery Basics, and Oracle Database Concepts for more information about flashback technology and automatic undo management

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.

The following recovery methods can be used:

- Resolving Table Inconsistencies
- Resolving Row and Transaction Inconsistencies
- Resolving Database-Wide Inconsistencies

#### 4.2.8.1 Resolving Table Inconsistencies

Oracle provides a FLASHBACK DROP statement to recover from an accidental DROP TABLE statement, and a FLASHBACK TABLE statement to restore a table to a previous point in the database.

Flashback Table provides the DBA the ability to recover a table, or a set of tables, to a specified point in time quickly and easily. 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-06 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 specified timestamp in the past. Flashback Table performs this operation online and in place, and it maintains referential integrity constraints between the tables.

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.

Flashback Drop provides a safety net when dropping objects in Oracle Database 10g. 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;

#### 4.2.8.2 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, a feature introduced in the Oracle9*i* Database, enables an administrator or user to query any data at some point in time in the past. This powerful feature can be used to view and reconstruct data that might have been deleted or changed by accident. For example:

```
SELECT * FROM EMPLOYEES
AS OF TIMESTAMP
TO_DATE('28-Jun-06 14:00','DD-Mon-YY HH24:MI')
WHERE ...
```

This partial statement displays rows from the EMPLOYEES table starting from 2 p.m. on June 28, 2006. Developers can use this feature to build self-service error correction into their applications, empowering end users to undo and correct their errors without delay, rather than burdening administrators to perform this task. Flashback Query is very simple to manage, because the database automatically keeps the necessary information to reconstruct data for a configurable time into the past.

Flashback Version Query provides a way to view changes made to the database at the row level. It 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-06 14:00','dd-Mon-YY hh24:mi') AND
TO_DATE('28-Jun-06 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. today. A DBA can use this to pinpoint when and how data is changed and trace it back to the user, application, or transaction. This enables the DBA 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 provides a way to view changes made to the database at the transaction level. It 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 = '00020003000002D';

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 this, the DBA 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 DBA 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. with Flashback Query.

To 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 past 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
FROM EMP
VERSIONS BETWEEN TIMESTAMP MINVALUE AND MAXVALUE
WHERE EMPNO = 7521
ORDER BY NVL(VERSIONS_STARTSCN,1);
```

EMPNO	ENAME	SAL	VERSIONS_ST	FARTTIME		VERSIONS_E	NDTIME	
7521	WARD	1250	28-JUN-06	08.48.43	AM	28-JUN-06	08.54.49 AM	
7521	WARD	1875	28-JUN-06	08.54.49	AM	28-JUN-06	09.10.09 AM	
7521	WARD	1250	28-JUN-06	09.10.09	AM			

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 modify the query to determine the transaction information for each of the changes affecting WARD using a similar Flashback Version Query. This time use the VERSIONS\_XID pseudocolumn.

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('000900050000089');
UNDO_SQL
update "SCOTT"."EMP" set "SAL" = '950' where ROWID = 'AAACV4AAFAAAAKtAAL';
update "SCOTT"."EMP" set "SAL" = '1500' where ROWID = 'AAACV4AAFAAAAKtAAL';
update "SCOTT"."EMP" set "SAL" = '2850' where ROWID = 'AAACV4AAFAAAAKtAAL';
update "SCOTT"."EMP" set "SAL" = '1250' where ROWID = 'AAACV4AAFAAAAKtAAE';
update "SCOTT"."EMP" set "SAL" = '1250' where ROWID = 'AAACV4AAFAAAAKtAAE';
update "SCOTT"."EMP" set "SAL" = '1600' where ROWID = 'AAACV4AAFAAAAKtAAE';
```

6 rows selected.

You can see that WARD's salary was not the only change that occurred in the transaction. The information that was changed for the other four employees at the same time as WARD can now be passed 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 DBA can execute those 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.

#### 4.2.8.3 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 will take hours or even days just to restore the whole database.

Flashback Database is a new strategy for doing point-in-time recovery. It 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. One way to think of it is as a continuous backup or storage snapshot. 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. It 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 of the instances 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 production 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 | CHANGE] date | 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 one of the following alternatives:
  - Use Data Guard to recover to the target time if the standby 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 4.3, "Restoring Fault Tolerance" on page 4-44.

Flashback Database does not automatically fix a dropped tablespace, but it can be used to dramatically reduce the downtime. You can flash back the production database to a point before the tablespace was dropped and then restore a backup of the corresponding datafiles from the affected tablespace and recover to a time before the tablespace was dropped.

Follow these recommended steps to use Flashback Database to repair a dropped tablespace:

- 1. Determine the SCN or time you dropped the tablespace.
- **2.** Flash back the database to a time before the tablespace was dropped. You can use a statement similar to the following:

FLASHBACK DATABASE TO BEFORE SCN drop\_scn;

- 3. Restore, rename, and bring datafiles online.
  - **a.** Restore only the datafiles from the affected tablespace from a backup.
  - b. Rename the unnamed files to the backup files.

ALTER DATABASE RENAME FILE '.../UNNAMED00005' to 'restored\_file';

#### 4. Bring the datafiles online.

ALTER DATABASE DATAFILE 'name' ONLINE;

#### 5. Query and recover the database.

SELECT CHECKPOINT\_CHANGE# FROM V\$DATAFILE\_HEADER WHERE FILE#=1; RECOVER DATABASE UNTIL CHANGE checkpoint\_change#;

**6.** Open the database.

ALTER DATABASE OPEN RESETLOGS;

# 4.3 Restoring Fault Tolerance

Whenever a component in a high-availability architecture fails, then 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 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 10g with RAC
  - Restoring Failed Nodes or Instances in a RAC Cluster
- For Oracle Database 10g with Data Guard and Oracle Database 10g with RAC and Data Guard MAA
  - Restoring a Standby Database After a Failover
  - Restoring ASM Disk Groups after a Failure
  - Restoring Fault Tolerance After Planned Downtime on Secondary Site or Clusterwide Outage
  - Restoring Fault Tolerance After a Standby Database Data Failure
  - Restoring Fault Tolerance After the Production Database Was Opened Resetlogs

- Restoring Fault Tolerance After Dual Failures

# 4.3.1 Restoring Failed Nodes or Instances in a RAC Cluster

Ensuring that application services fail over quickly and automatically in a RAC cluster—or between primary and secondary sites—is important when planning for both scheduled and unscheduled outages. 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 a RAC cluster or databases between sites.

Adding a failed node back into the cluster or restarting a failed RAC 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 so as to incur minimal or no effect on the current running environment
- Resetting network components (such as load balancer) which were modified for failover and now must be reset
- 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 RAC environment at any time. Processing to complete the reconfiguration of a node may require additional system resources.

Table 4–13 summarizes additional processing that may be required when adding a node.

Table 4–13 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 new 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 a RAC instance	When you restart a 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.

#### See Also:

- Your vendor-specified cluster management documentation for detailed steps on how to start and join a node back into a cluster
- Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for more information about restarting a RAC instance

The following recovery methods can be used:

- Recovering Service Availability
- Considerations for Client Connections After Restoring a RAC Instance

## 4.3.1.1 Recovering Service Availability

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 up 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 Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide

## 4.3.1.2 Considerations for Client Connections After Restoring a RAC Instance

After a RAC instance has been restored, additional steps might be required, depending on the current resource utilization and performance of the system, the application configuration, and the network load balancing that has been implemented.

Existing connections (which might have failed over or started as a new session) on the surviving 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 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.

New connections are started as they are needed, on the least-used node, assuming connection load balancing has been configured. Therefore, the new connections are automatically load-balanced over time.

An application service can be:

- Partitioned with services running on a subset of 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 the cases where an application is partitioned or has a combination of partitioning and nonpartitioning, the response time and availability aspects for each service should be considered. If redistribution or failback of connections for a particular service is required, then you can rebalance workloads manually with the DBMS\_SERVICE.DISCONNECT\_SESSION PL/SQL procedure. You can use this procedure to disconnect sessions associated with a service while the service is running.

For load-balancing application services across multiple 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 new 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 4–14 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.

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 DBMS_ SERVICE.DISCONNECT_SESSION to manually disconnect sessions and allow them to be reestablished on one of the remaining instances that provides the service.	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

Table 4–14 Restoration and Connection Failback

Figure 4–13 shows a two-node partitioned 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 4–13 Partitioned Two-Node RAC Database

Figure 4–14 shows what happens when one RAC instance fails.



Figure 4–14 RAC Instance Failover in a Partitioned Database

If one RAC instance fails, then the service and existing client connections can be automatically failed over to another RAC instance. In this example, the HR and Sales services are both supported by the remaining RAC instance. In addition, new client connections for the Sales service can be routed to the instance now supporting this service.

After the failed instance has been repaired and restored to the state shown in Figure 4–13 and the Sales service is relocated to the restored instance failed-over clients and any new clients that had connected to the Sales service on the failed-over instance might have to be identified and failed back. New client connections, which are 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 4–15 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 4–15 Nonpartitioned RAC Instances

Heartbeat --- hb---

If one RAC instance fails, then CRS moves the services that were running on the failed instance. In addition, new client connections are routed only to the available RAC instances that offer that service.

After the failed instance has been repaired and restored to the state shown in Figure 4–15, 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 is not able to adequately service the requests.

New client connections that are 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 will again evenly balance across all available RAC instances. Rebalancing the load immediately after restoration depends on the resource utilization and application response times.

## 4.3.2 Restoring a Standby Database After a Failover

Following unplanned downtime on a production 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 former primary database as a standby database. **Reinstatement** restores high availability to the broker configuration so that, in the event of a failure of the new primary database, another fast-start failover can occur. The reinstated database can act as the fast-start failover target for the new primary database, making a subsequent fast-start failover possible. The new standby database is a viable target of a failover when it begins applying redo data received from the new primary database.

The broker can reinstate the former primary database as a standby database without the need to reenable the primary database or to manually perform a Flashback Database operation. To reinstate the former 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 former standby database.

If the former 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 reinstantiated 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:** The following topics in *Oracle Data Guard Concepts and Administration*:

- Flashing Back a Failed Primary Database into a Physical Standby Database
- Flashing Back a Failed Primary Database into a Logical Standby Database

This section includes the following topics:

- Restoring a Standby Database After a Fast-Start Failover
- Reinstating a Standby Database Using Enterprise Manager After a Failover

#### 4.3.2.1 Restoring a Standby 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 Data Guard broker to automatically reinstate it as a standby database to the new primary. This quickly restores disaster protection and high availability for the new primary database.

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, as shown in Figure 4–16.

ORACLE Enterprise Manager 10g				Onter Destaurant links i servit
Grid Control				Setup Preferences Help Logout
			eployments Alerts	Policies Jobs Reports
Databases   Hosts   Web Applications   Services   System	ms   Groups   .	All Targets		
ost: north.foo.com > Database Instance: North_Sales >	>			Logged in As SY
Information				
The fast-start failover mode has been successfully changed	l.			
ata Guard				
age Refreshed June 8, 2005 2:44:45 PM EDT			View Data Real T	ïme: Manual Refresh 🛛 🖌 📳
Overview	St	andby Progress Summ	ary	
Data Guard Status  Vinnal Protection Mode Maximum Availability Fast-Start Failover Enabled to DR_Sales Observer Location north		lo. The apply lag is the time differ		update and the standby last received st update and the standby last applied
Primary Database	<u>.</u>	30		
Name <u>North Sales</u> Host <u>north</u> Data Guard Status <u>v Normal</u> Current Log <u>174</u> Properties <u>Edit</u>		30 15 0 0 0 DR_Sales	3 West_Sales	Apply Lag
Standby Databases				
(Edit) Remove) (Switchover) (Failover)				Add Standby Databas
Select Name Host Data Guard Status Ro	ole	Last Received Log	Last Applied Log	Estimated Failover Time
OR_Sales north ✓ Normal Pr	nysical Standby	<u>173</u>	173	O seconds
○ West_Sales north	gical Standby	173	173	O seconds
Performance Performance Overview Log File Details		Additional Admini Verify Remove Data Guard		

#### Figure 4–16 Fast-Start Failover and the Observer Are Successfully Enabled

## 4.3.2.2 Reinstating a Standby Database Using Enterprise Manager After a Failover

Furthermore, you can leverage Enterprise Manager to reinstate the old primary as the new standby. Figure 4–17 shows an example of the warning message that shows in Enterprise Manager when a reinstatement is needed.

Figure 4–17 Reinstating the Former Primary Database After a Fast-Start Failover

ORACLE Enterprise Manager 10g				Setup Preferences Help Logout
Grid Control	Home	Targets Deplo	vments Alerts	Policies lobs Reports
Databases   Hosts   Web Applications   Services   Systems   Group	s   All Target		yments Alerts	Toncies Jobs Keports
Host: north.foo.com > Database Instance: North_Sales >				Logged in As SYS
Data Guard				
Page Refreshed June 8, 2005 3:22:24 PM EDT			View Data Real Tim	e: Manual Refresh 🛛 🛛 🕏
Overview	Standby Pr	ogress Summary		
Data Guard Status (Maximum Availability (Unsynchronized) Protection Mode <u>Maximum Availability (Unsynchronized)</u> Fast-Stath Failowe <u>Enabled to DR_Sales (Not Ready</u> ) Observer Location <b>north</b>				date and the standby last received apdate and the standby last applied
Primary Database				
Name North Sales	0.5			
Host <u>north</u> Data Guard Status <u>ORA-16817: unsynchronized Fast-Start</u>	L			
Data Guard Status A <u>Failover configuration</u> Current Log 187 Properties <u>Edit</u>	0.0	No data is	currently available.	
Standby Databases				
				Add Standby Database
(Edit) Remove (Switchover) Failover				
Select Name Host Data Guard Status Role		Last Received Log	Last Applied Log	Estimated Failover Time
	sical Standby	<u>n/a</u>	<u>n/a</u>	n/a
○ West_Sales north	cal Standby	<u>186</u>	<u>186</u>	n/a
Performance Performance Overview Log File Details	Verif	itional Administrat ¥ ove Data Guard Confi		

# 4.3.3 Restoring ASM Disk Groups after a Failure

Follow the steps in Section 4.2.6.3, "Data Area Disk Group Failure" on page 4-29 or Section 4.2.6.4, "Flash Recovery Area Disk Group Failure" on page 4-31.

# 4.3.4 Restoring Fault Tolerance After Planned Downtime on Secondary Site or Clusterwide Outage

After performing the planned maintenance on the secondary site, the standby database and log apply services must be restarted, and then Data Guard will automatically catch up. You can leverage Enterprise Manager and Data Guard 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 production 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 4–15
 SQL Statements for Starting Physical and Logical Standby Databases

Type of Standby Database	SQL Statement
Physical	STARTUP MOUNT;
Logical	STARTUP;

2. Start Redo Apply (physical standby) or SQL Apply (logical standby):

 Table 4–16
 SQL Statements to Start Redo Apply and SQL Apply

Type of Standby Database	SQL Statement
Physical	RECOVER MANAGED STANDBY DATABASE DISCONNECT;
Logical	ALTER DATABASE START LOGICAL STANDBY APPLY;

3. Verify redo transport services on production database

You might have to reenable the production 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 production 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 production database protection mode

If you had to change the protection mode of the production database from maximum protection to either maximum availability or maximum performance because of the standby database outage, then change the production 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

# 4.3.5 Restoring Fault Tolerance After a Standby Database Data Failure

Following unplanned downtime on the standby database that requires a full or partial datafile 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 need to reinstantiate or re-create the relevant objects that got affected by the corruption.

To repair data corruption or data failures on the standby database, you can leverage the following repair solutions:

- Use RMAN Block Media Recovery (described in Section 4.2.7.2 on page 4-36)
- Use RMAN Data File Media Recovery (described in Section 4.2.7.3 on page 4-37)
- Re-Create Objects Manually for logical standby databases only (described in Section 4.2.7.4 on page 4-37)

If you had to change the protection mode of the production database from maximum protection to either maximum availability or maximum performance because of the standby database outage, then change the production database protection mode back to maximum protection (depending on your business requirements).

ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE [PROTECTION | AVAILABILITY];

# 4.3.6 Restoring Fault Tolerance After the Production Database Was Opened Resetlogs

If the production 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 production database did complete recovery with no resetlogs.

After opening the production database with the RESETLOGS option, execute the queries shown in Table 4–17.

Table 4–17Queries to Determine RESETLOGS SCN and Current SCN OPENRESETLOGS

Database	Query
Production	SELECT TO_CHAR(RESETLOGS_CHANGE# - 2) FROM V\$DATABASE;
Physical standby	SELECT TO_CHAR(CURRENT_SCN) FROM V\$DATABASE;
Logical standby	SELECT APPLIED_SCN FROM DBA_LOGSTDBY_PROGRESS;

Table 4–18 shows the actions you take to restore fault tolerance if the standby database is behind the primary database's resetlogs SCN.

Database	Action	
Physical standby	<b>1.</b> Ensure that the standby database has received a new archived redo log file from the production database.	
	<b>See Also</b> : "Verify redo transport services on production database" on page 4-53	
	2. Restart Redo Apply.	
Logical standby	Ensure that the standby database has received a new archived redo log file from the production database.	
	<b>See Also</b> : "Verify redo transport services on production database" on page 4-53	

Table 4–18SCN on Standby Database is Behind Resetlogs SCN on the ProductionDatabase

Table 4–19 shows the actions you take to restore fault tolerance if the standby database is ahead of the primary database's resetlogs SCN.

Database	Ac	tion
Physical standby	1.	Ensure that the standby database has received a new archived redo log file from the production database.
		See Also: "Verify redo transport services on production database" on page 4-53
	2.	Issue the SHUTDOWN IMMEDIATE statement, if necessary
	3.	Issue the STARTUP MOUNT statement
	4.	Issue the FLASHBACK DATABASE TO SCN <i>flashback_scn</i> statement wher <i>flashback_scn</i> is the SCN returned from the production database query in Table 4–17. The SCN returned from the production database query is 2 less than the RESETLOGS_CHANGE#.
		Issue the FLASHBACK DATABASE TO SCN resetlogs_change#_minus_ 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;
Logical standby	1.	Retrieve production database flashback time or SCN. The flashback time or SCN must be extracted from the production database alert log. This assumes that the clocks are the same between the database machines or the flashback time will need to be adjusted.
	2.	Stop SQL Apply:
		ALTER DATABASE STOP LOGICAL STANDBY APPLY; SELECT APPLIED_SCN FROM DBA_LOGSTDBY_PROGRESS;
	3.	Issue the following SQL statements to flash back the logical standby database to the same time that was used to flash back the primary database:
		SHUTDOWN; STARTUP MOUNT EXCLUSIVE; FLASHBACK DATABASE TO TIMESTAMP time_of_primary_database_flashback; ALTER DATABASE OPEN READ ONLY; SELECT APPLIED_SCN FROM DBA_LOGSTDBY_PROGRESS;
		The last SQL statement queries the APPLIED_SCN column of the DBA_ LOGSTDBY_PROGRESS view, the results of this query should confirm that SQI Apply has applied less than or up to the APPLIED_SCN obtained in step 2.If not, you need to flash back the database further.
	4.	Open the logical standby database with resetlogs:
		SHUTDOWN; STARTUP MOUNT EXCLUSIVE; ALTER DATABASE OPEN RESETLOGS;
	5.	Archive the current log on the primary database:
		ALTER SYSTEM ARCHIVE LOG CURRENT;
	6.	Start SQL Apply:
		ALTER DATABASE START LOGICAL STANDBY APPLY;

 Table 4–19
 SCN on the Standby is Ahead of Resetlogs SCN on the Production Database

# 4.3.7 Restoring Fault Tolerance After Dual Failures

If a dual failure affecting both the standby and production databases occurs, then you must re-create the production database first. Because the sites are identical, the production database can be created wherever the most recent backup resides.

Table 4–20 summarizes the recovery strategy depending on the type of backups that are available.

Table 4–20 Re-Creating the Production and Standby Databases

Available Backups	Re-Creating the Production Database
Local backup on production and standby databases	Restore backup from the production database. Recover and activate the database as the new production 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 production database.
Tape backups only	Restore tape backups locally. Recover the database and activate it as the new production database.

**See Also:** After the production database is re-created, follow the steps for creating a new standby database that are described in*Oracle Data Guard Concepts and Administration* 

# 4.4 Eliminating or Reducing Downtime for Scheduled Outages

This section describes best practices for eliminating or reducing downtime due to scheduled outages and contains the following topics:

- Storage Maintenance
- RAC Database Patches
- Database Upgrades
- Database Platform or Location Migration
- Online Database and Application Upgrades
- Database Object Reorganization
- System Maintenance

## 4.4.1 Storage Maintenance

The following procedure should be used when you add storage to the system. The procedures in the following sections assume that you are adding storage to an ASM Disk Group.

- Migrating to ASM Storage
- Adding and Removing Storage

## 4.4.1.1 Migrating to ASM Storage

#### See Also:

- Oracle Database Backup and Recovery Advanced User's Guide for detailed instructions on migrating a database to ASM using RMAN
- Oracle Database PL/SQL Packages and Types Reference for the DBMS\_ FILE\_TRANSFER package

#### 4.4.1.2 Adding and Removing Storage

Disks can be added to and removed from ASM with no downtime. When disks are added or removed, 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:

- Investigate methods of adding storage to, and removing storage from, the host operating system with no downtime.
- Use a single ALTER DISKGROUP command when adding or removing multiple disk drives.

For example, if the storage maintenance is to add new drives and remove existing drives, use a single ALTER DISKGROUP command with the ADD DISK clause to add the new drives, and the DROP DISK clause to remove the existing drives. For example:

```
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. Once 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: The "Using Automated Storage Management (ASM)" chapter in Oracle Database Administrator's Guide
```

## 4.4.2 RAC Database Patches

With RAC, you can apply certain database patches to one node or instance at a time, which enables continual application and database availability. "One-off" patches or interim patches 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 Real Application Clusters with little or no database downtime. The tool used to achieve this is the opatch command-line utility.

The advantage of a RAC rolling upgrade is that it enables at least some instances of the RAC installation to be available during the scheduled outage required for patch upgrades. Only the RAC instance that is currently being patched must be brought down. The other instances can continue to remain available. This means that 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 RAC installation.

Rolling upgrade is available only for patches that have been certified by Oracle to be eligible for rolling upgrades. Typically, patches that can be installed in a rolling upgrade include:

- Patches that do not affect the contents of the database such as the data dictionary
- Patches not related to RAC internode communication
- Patches related to client-side tools such as SQL\*PLUS, Oracle utilities, development libraries, and Oracle Net
- Patches that do not change shared database resources such as datafile headers, control files, and common header definitions of kernel modules

Rolling upgrade of patches is currently available for one-off patches only. It is not available for patch sets.

Rolling patch upgrades are 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.

### 4.4.2.1 Best Practices To Minimize Downtime

Use the following recommended practices for all database patch upgrades:

- 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 as well as 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 RAC rolling upgrade and you can incur the downtime for applying the patch, go to Section 4.4.3, "Database Upgrades" on page 4-61 to assess whether or not other solutions are feasible.

The following are additional recommended practices for RAC rolling upgrades.

If multiple instances share an Oracle home, then all of them will be affected by application of a patch. The DBA should verify that this will not cause unintentional side effects. Also, all such instances on a node must be shut down during the patch application. Scheduled outage planning should take this into account. 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 a repository of the Oracle Database software installed on the node. 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 exactly 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 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 rolling patch upgrade enables it to be applied to only some nodes of the 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 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 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 RAC installation. When instances of a RAC cluster have similar patch software, services can be migrated among instances without running into the problem a patch might have fixed.

 All patches (including those applied by rolling upgrades) should be maintained online and not removed once they have been applied. This is useful if a patch must be rolled back or applied again.

The patches should be stored 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.

- Rolling patch upgrades, just like any other patch upgrade, should be done when no other patch upgrade or Oracle installation is being done on the node. Application of multiple patches is a sequential process. The scheduled outage should be planned accordingly.
- If multiple patches have to be applied and they must be applied at the same time, and if only some of these patches are eligible for rolling upgrade, then apply all of them in a nonrolling manner. This reduces the overall time required to get through the patching process.
- For patches that are not eligible for rolling upgrade, the next best option for RAC deployments is the minimize\_downtime option of the apply command.
- Perform the rolling upgrade when system usage is low. This ensures minimal disruption of service for the end user.

**See Also:** *Oracle Universal Installer and OPatch User's Guide* for more information on the opatch utility

# 4.4.3 Database Upgrades

The following Oracle features are available to perform database upgrades:

- Database Upgrade Assistant (DBUA)
- Data Guard SQL Apply (logical standby database)
- Oracle Streams
- Transportable tablespaces

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 prior to the downtime
- Temporary additional resources necessary (for example, disk space or CPU)
- Complexity of the steps needed to complete the upgrade

Table 4–21 lists the methods that can be used for platform migrations and database upgrades, and recommends what method to use.

 Table 4–21
 Platform Migration and Database Upgrade Options

Upgrade Method	Use This Method When
Database Upgrade Assistant	Recommended method when maintenance window is sufficient
Data Guard SQL Apply (Logical Standby)	DBUA will not finish within the maintenance window and the database is not a candidate for RAC rolling patch upgrade
Oracle Streams	Already a Streams implementation or Data Guard SQL Apply rolling upgrade does not support database versions in use
Transportable Tablespaces	Database is using data types unsupported by Data Guard SQL Apply or Streams

### 4.4.3.1 Database Upgrade Assistant

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, Streams, 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
  - Recompile all PL/SQL
  - Reconnect the clients to the upgraded database

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 on DBUA and upgrading your Oracle Database software

# 4.4.3.2 Data Guard SQL Apply (Logical Standby)

Data Guard SQL Apply can be used to upgrade a database with minimal downtime by means of a process called rolling upgrade. Data Guard currently supports homogeneous environments where the primary and standby databases are running on the same platform.

Note the following points when deciding if Data Guard SQL Apply is the appropriate method for minimizing downtime during a database upgrade:

- The Data Guard SQL Apply infrastructure uses Oracle Streams and therefore inherits Oracle Streams data type restrictions on user-defined types, such as object types, REF values, varrays, and nested tables.
- Support for rolling upgrade starts with Oracle Database 10g release 1 (10.1.0.3). The supported versions, for both the source database and the target database, are more restrictive than Oracle Streams.
- 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

Use Data Guard SQL Apply for rolling database upgrade when DBUA will not complete the upgrade within the maintenance window and the application does not use user-defined types.

**See Also:** Oracle Data Guard Concepts and Administration for more information on SQL Apply

## 4.4.3.3 Oracle Streams

Oracle Streams can be used to upgrade the database software from one version to another with minimal downtime. This is because Oracle Streams supports a configuration in which the primary database and its replica are running on different database versions.

Note the following points when deciding if Oracle Streams is an appropriate method for a database upgrade:

- Oracle Streams does not support user-defined types, such as object types, REF values, varrays, and nested tables. However, shadow tables can be created on the primary database that do not have the unsupported data types and the shadow tables can be replicated.
- The source database must be running Oracle9*i* release 2 or higher.
- The administrative effort required to set up and maintain the Oracle Streams environment is more than if using Data Guard SQL Apply for a database upgrade.
- There might be a performance impact on the source database while the source and target databases run in parallel as changes are propagated to the target database.
- Downtime required for a database upgrade when using Oracle Streams is determined by the time needed to apply the remaining transactions in the queue and to reconnect the clients to the new database.

Consider using Oracle Streams if the application already uses Streams or when clients do not use user-defined types and the extra administrative effort is worth the opportunity for a very small outage time.

**See Also:** Oracle Streams Concepts and Administration for more information on database upgrading using Oracle Streams

## 4.4.3.4 Transportable Tablespaces

Transportable tablespaces can be used to accomplish a database upgrade by transporting all user datafiles into a pre-created, prepared target database.

Note the following points when deciding if transportable tablespaces is the appropriate method for performing a database upgrade:

- 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.
- Downtime required for a 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
  - If the target database is on a remote system, then include the time to transfer all datafiles from the source system to the target system

The time it takes to transfer the datafiles can be reduced significantly by using a storage infrastructure that can make the datafiles available to the target system without the need to physically move the files, or by using a physical standby database.

Use transportable tablespaces to perform a database upgrade when DBUA will not complete within the maintenance window, and Oracle Streams or Data Guard SQL Apply cannot be used due to data type restrictions.

**See Also:** *Oracle Database Administrator's Guide* for more information on transportable tablespaces

## 4.4.4 Database Platform or Location Migration

The following Oracle features are available to perform platform migrations:

- Transportable Database
- Oracle Streams
- Oracle Data Pump
- Transportable Tablespaces
- Data Guard Redo Apply (Physical Standby Database)

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 prior to the downtime
- Amount of temporary additional resources necessary, such as disk space or CPU
- Complexity of the steps needed to complete maintenance operations

The following table summarizes the methods that can be used for platform migrations and database upgrades, and recommends which method to use for each operation.
Operation	Method	When to Use
Platform migration	Transportable Database	Recommended method
to same endian platform	Oracle Streams	Transportable Database will not finish within the maintenance window
Platform migration	Oracle Data Pump	Recommended method
to different endian platform	Oracle Streams	Data Pump will not finish within the maintenance window
	Transportable Tablespaces	Database is using data types unsupported by Oracle Streams
Location Migration	Data Guard Redo Apply (Physical Standby Database)	Recommended method

Table 4–22 Platform and Location Migration Options

**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.

#### 4.4.4.1 Transportable Database

Transportable Database is a new feature in Oracle Database 10g Release 2 (10.2) that is the recommended method for migrating an entire database to another platform that has the same endian format.

Note 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 that are of 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 all data files to the new platform format
  - Transfer all data files from the source system to the target system

You can significantly reduce this time by using a storage infrastructure that can make the data files available to the target system without the need to physically move the files.

Transportable Databases is the recommended method for migrating an entire database to another platform that has the same endian format because it's the simplest approach.

**See Also:** Oracle Database Backup and Recovery Advanced User's Guide for more information on cross-platform use of Transportable Database

#### 4.4.4.2 Oracle Streams

Oracle Streams can be used to move a database from one platform to another with minimal downtime. This is because Oracle Streams supports a configuration in which the primary database and its replica are running on different platforms.

Note the following points when deciding if Oracle Streams is an appropriate method for a platform migration:

- Oracle Streams does not support user-defined types, such as object types, REF values, varrays, and nested tables.
- To perform an upgrade using Oracle Streams, the source database must be running Oracle9*i* release 2 or higher.
- The administrative effort required to set up and maintain the Oracle Streams environment is more than if using Data Guard SQL Apply for a database upgrade.
- There might be a performance impact on the source database while the source and target databases run in parallel as changes are propagated to the target database.
- Downtime required for a platform migration when using Oracle Streams is determined by the time needed to apply the remaining transactions in the queue and to reconnect clients to the new database.

Consider using Oracle Streams when the application does not use user-defined types and the extra administrative effort is worth the opportunity for a very small outage time.

**See Also:** Oracle Streams Concepts and Administration for more information on database upgrading using Oracle Streams

## 4.4.4.3 Oracle Data Pump

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 points when deciding if Data Pump is an appropriate method for a platform migration:

- Oracle Data Pump is available only on Oracle Database 10g Release 1 (10.1) and later releases.
- Downtime required for a platform migration when using Data Pump is determined by the time needed to perform a full database network import. A network import uses a database link between the target system and the remote source system to retrieve data and write it directly into the target system, without the use of dump files.

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 and the Export and Import utilities
- Oracle Database Upgrade Guide for more information about upgrading your Oracle Database software

#### 4.4.4.4 Transportable Tablespaces

Transportable tablespaces can be used to accomplish a platform migration by transporting all user datafiles into a pre-created, prepared target database.

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 datafiles from the source system to the target system

This time can be reduced significantly by using a storage infrastructure that can make the datafiles available to the target system without the need to physically move the files

- Convert all datafiles to the new platform format using RMAN

Use transportable tablespaces to migrate to a new platform when Oracle Data Pump will not complete within the maintenance window, and Oracle Streams or Data Guard SQL Apply cannot be used due to data type restrictions.

**See Also:** *Oracle Database Administrator's Guide* for more information on transportable tablespaces

## 4.4.4.5 Data Guard Redo Apply (Physical Standby Database)

Data Guard Redo Apply can be used 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.

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 on Redo Apply and physical standby databases

# 4.4.5 Online Database and Application Upgrades

An Oracle database upgrade is the process of transforming an existing, prior release of an Oracle Database system into the current release of the Oracle Database system and can be a very lengthy process. An application upgrade may include a database upgrade and any application code and schema changes required. If database upgrade with Data Guard is not applicable and zero to minimum downtime is required for the database or application upgrade, then configure Oracle Streams to perform a database upgrade with little or no downtime. To do so, you use Oracle Streams to configure a single-source replication environment with the following databases:

- Source Database: The original database that is being upgraded
- Capture Database: The database where a capture process captures changes made to the source database during the upgrade
- Destination Database: The copy of the source database where an apply process applies changes made to the source database during the upgrade process. The apply process can apply to the same or different schema and object structure using

Specifically, you can use the following general steps to perform a database upgrade while the database is online:

**1.** Create an empty destination database.

- **2.** Configure an Oracle Streams single-source replication environment where the original database is the source database and a copy of the database is the destination database for the changes made at the source.
- **3.** Perform the database upgrade on the destination database. During this time the original source database is available online.
- **4.** Use Oracle Streams to apply the changes made at the source database to the destination database.
- **5.** When the destination database has caught up with the changes made at the source database, take the source database offline and make the destination database available for applications and users.

If the schema or object structure is different at the destination database, then Streams transformations need to be incorporated to manipulate the change to its new structure.

Figure 4–18 provides an overview of this process.



Figure 4–18 Online Database Upgrade with Oracle Streams

**See Also:** Appendix C "Online Database Maintenance with Streams" in *Oracle Streams Concepts and Administration* 

## 4.4.6 Database Object Reorganization

Many scheduled outages related to the data server involve some reorganization of the database objects. The database object reorganization must be accomplished with continued availability of the database. Oracle's online object reorganization capabilities have been available since Oracle8*i*. These capabilities enable object reorganization to be performed even while the underlying data is being modified.

Table 4–23 describes a few of the object reorganization capabilities available with Oracle Database 10*g*.

Object Type	Example of Object Reorganization Solution	Description of Solution
Table	DBMS_REDEFINITION PL/SQL package	A PL/SQL package that provides a mechanism to redefine tables online. This is Oracle's recommended best practice.
Index	Rebuild index	Rebuild an index that has previously been marked as unusable.
Tablespace	Rename tablespace	Enables an existing tablespace to be renamed without rebuilding the tablespace and its contents.

Table 4–23 Some Object Reorganization Capabilities

In highly available systems, it is occasionally necessary to redefine large tables that are constantly accessed to improve the performance of queries or DML. The Online Reorganization and Redefinition feature in Oracle Database 10g, offers administrators unprecedented flexibility to modify table physical attributes and transform both data and table structure, while allowing users 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 and it can make the application upgrade process easier, safer and faster.

Oracle's recommended 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 automatically through Oracle Enterprise Manager, the entire reorganization process occurs while users have full access to the table thus ensuring system availability.

Figure 4–19 shows the Reorganize Objects Wizard in Oracle Enterprise Manager 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 automatically generates the script and performs the reorganization.

DRACLE Enterprise Manager 10g	Selue Preferences Helz Looz Database
	0 0
Type Objects Options Impact Report Sch	nedule Review
Reorganize Objects: Type	
Database orcl Logged in As SYSTEM	Cancel Step 1 of 6 N
ou can reorganize individual schema objects or an entire tablespace. Jelect the reorganization type:	Overview
€ Schema Objects C Tablespace	Reorganization is necessary for: - Rebuilding indexes that are fragmented - Rebuilding tables that are fragmented - Relocating objects to another tablespace - Recreating objects with optimal storage attributes

#### Figure 4–19 Database Object Reorganization Using Oracle Enterprise Manager

Using the DBMS\_REDEFINITION approach, an interim table is created that contains all the desired attributes. The reorganization begins by calling the procedure START\_ REDEF\_TABLE, which is where the column mappings between the current and new version of the table are described. All the dependent objects such as triggers, constraints and indexes are automatically copied to the interim table using the procedure COPY\_TABLE\_DEPENDENTS. During the reorganization, any changes made to the original table are added to the interim table by calling the procedure SYNC\_ INTERIM\_TABLE. The reorganization is complete when the procedure FINISH\_ REDEF\_TABLE is called and the interim table is renamed as the main table.

A tablespace can be renamed in Oracle Database 10*g*, similar to the ability to rename a column, table and datafile. Previously, the only way to change a tablespace name was to drop and re-create the tablespace, but this meant that the contents of the tablespace had to be dropped and rebuilt later. With the ability to rename a tablespace online, there is no interruption to the users.

ALTER TABLESPACE USERS RENAME TO new\_tablespace\_name;

Tablespace altered.

Additionally, consider the following when performing data reorganization:

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 impact less than ten percent of the table while online operation is in progress. Also the database administrator can use the Database Resource Manager to minimize the data reorganization impact to users by allocating enough resources to users.

 Oracle does not recommend running online operations at peak times or running a batch job that modifies large amount of data during an online data reorganization.

In fact, parallel DML, direct load and import/export cannot be performed during an online operation.

Rebuilding index online vs. dropping an index and then re-creating a new 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.

Coalescing an index online vs. 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 needs to move an index to a new tablespace, use online index rebuild.

Local and global indexes.

Oracle Database 10g 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 on redefining tables online
- The Online Data Reorganization & Redefinition web site on OTN at

http://www.oracle.com/technology/deploy/availabil ity/htdocs/online\_ops.html

 The "Online Reorganization using Oracle Database 10g" white paper that provides additional redefinition solutions online at http://www.oracle.com/technology/deploy/availabil ity/pdf/ha\_10gR2\_online\_reorg\_twp.pdf

## 4.4.7 System Maintenance

For a scheduled outage that requires an instance, node, or other component to be isolated, RAC provides the ability to relocate, disable, and enable services. 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 re-enabled 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 at the end of the maintenance outage.

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for information about administering services with Enterprise Manager, DBCA, PL/SQL, and SRVCTL

When using RAC, Oracle Clusterware daemons start automatically at the time the node is started. When performing maintenance that requires one or more system reboots 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. Once maintenance is complete, enable and start the Oracle Clusterware daemons with crsctl commands.

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for information about using the crsctl command

# **Migrating to an MAA Environment**

This chapter provides the best practice recommendations for migrating your current configuration to an Maximum Availability Architecture (MAA) environment to create a redundant, reliable system and database, without sacrificing simplicity and performance.

This chapter contains these topics:

- Overview of Migrating to MAA
- Migrating to RAC from a Single Instance
- Adding a Data Guard Configuration to a RAC Primary

# 5.1 Overview of Migrating to MAA

MAA combines the scalability and availability advantages of Oracle Real Application Clusters (RAC) with the site protection capabilities of Oracle Data Guard.

An **MAA environment** consists of a site containing a RAC production database and a second site containing a cluster that minimally hosts at least one physical or logical standby database, but ideally hosts a combination of logical and physical standby databases. This environment provides the most comprehensive solution for both unplanned and planned outages because it inherits the capabilities and advantages of both Oracle Database 10g with RAC and Oracle Database 10g with Data Guard.

However, while the ideal MAA configuration includes a RAC primary database with a RAC standby database, business requirements or other considerations might indicate that you choose a different ending configuration or that you perform a phased migration. That is, some of the ending configurations could actually be intermediate steps in a phased implementation to a RAC primary with RAC standby configuration.

The setup of your current configuration will determine which sections in this chapter you should complete. For example, Table 5–1 describes migration instructions for some possible starting configurations.

IF your starting configuration includes	THEN
A single-instance primary database	Migrate the primary database to RAC using the instructions in "Migrating to RAC from a Single Instance" on page 5-2
A single-instance standby database	Migrate the standby database to RAC using the instructions in "Migrating to RAC from a Single Instance" on page 5-2

Table 5–1 Starting configurations Before Migrating to an MAA Environment

IF your starting configuration includes	THEN
A single-instance Data Guard configurations	Migrate the primary and/or standby databases to RAC using the instructions in "Migrating to RAC from a Single Instance" on page 5-2
A RAC primary database, but no Data Guard configuration	See "Adding a Data Guard Configuration to a RAC Primary" on page 5-2 to add a single-instance standby or a RAC standby database to the configuration.

Table 5–1 (Cont.) Starting configurations Before Migrating to an MAA Environment

# 5.2 Migrating to RAC from a Single Instance

The process of adding nodes to form a RAC database involves first cloning Oracle Clusterware and RAC software to new nodes and then adding new RAC instances. Basically, the steps include the following tasks:

- 1. Connect new nodes to the cluster
- 2. Extend clusterware and Oracle software to new nodes
- 3. Prepare storage for RAC on new nodes
- 4. Add nodes at the Oracle RAC database layer
- 5. Add database instances to new nodes

**See Also:** Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide for complete step-by-step information

# 5.3 Adding a Data Guard Configuration to a RAC Primary

The process of adding a Data Guard configuration to a RAC primary database includes the following basic tasks:

- 1. Configure Oracle Net Services on the standby database
- 2. Create the standby instances and database
- 3. Configure the primary database for Data Guard
- 4. Verify the Data Guard configuration

A series of MAA white papers have been published that provide step-by-step instructions on how to perform these tasks. The following white papers describe how to create either a single-instance Data Guard standby database, or RAC standby database:

 "MAA / Data Guard 10g Setup Guide—Creating a RAC Standby for a RAC Primary"

http://www.oracle.com/technology/deploy/availability/pdf/MAA\_ WP\_10g\_RACPrimaryRACStandby.pdf

 "MAA / Data Guard 10g Setup Guide—Creating a Single Instance Standby for a RAC Primary"

http://www.oracle.com/technology/deploy/availability/pdf/MAA\_ WP\_10g\_RACPrimarySingleInstanceStandby.pdf

 "MAA/ Data Guard 10g Setup Guide—Creating a RAC Logical Standby for a RAC Primary Database"

#### http://www.oracle.com/technology/deploy/availability/pdf/MAA\_ WP\_10gR2\_RACPrimaryRACLogicalStandby.pdf

This white paper describes how to create a RAC logical standby database from an existing RAC physical standby database and includes the following basic steps:

- 1. Prepare the physical standby database environment
- 2. Convert the physical standby database to a logical standby database
- 3. Verify the Data Guard configuration

**See Also:** Chapters 3 and 4 in *Oracle Data Guard Concepts and Administration* for detailed information about converting a single-instance physical standby database into a logical standby database, refer to:

Α

# Database SPFILE and Oracle Net Configuration File Samples

The tables and file samples in this appendix are included to illustrate the best practices as they relate to different high-availability architectures. These samples also clarify how the database server parameter file (SPFILE) relates to the Oracle Net configuration for dynamic service registration.

This appendix includes the following tables and sample files:

- SPFILE Samples
  - Table A–1, "Generic SPFILE Parameters for Primary, Physical Standby, and Logical Standby Databases"
  - Table A–2, "RAC SPFILE Parameters for Primary, Physical Standby, and Logical Standby Databases"
  - Table A–6, "Data Guard SPFILE Parameters for Primary and Physical Standby Database Only"
  - Table A-7, "Data Guard SPFILE Parameters for Primary and Logical Standby Database Only"
  - Table A–8, "Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, and Logical Standby Database: Maximum Availability or Maximum Protection Modes"
  - Table A–9, " Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, and Logical Standby Database: Maximum Performance Mode"
- Oracle Net Configuration Files
  - SQLNET.ORA Example for All Hosts Using Dynamic Instance Registration
  - LISTENER.ORA Example for All Hosts Using Dynamic Instance Registration
  - TNSNAMES.ORA Example for All Hosts Using Dynamic Instance Registration

The tables and files are shown for the following configuration:

- ORACLE\_BASE=/mnt/app/oracle
- Database flash recovery area is /flash\_recovery

# A.1 SPFILE Samples

The tables in this section represent the database, RAC, and Data Guard parameter file values. Some parameters appear in both the generic database parameter table and the RAC parameter table. If RAC is being used, then the value in the RAC parameter table should be used instead of the value in the generic database parameter table.

The parameters show the configuration for a database in Chicago and an option for a physical standby database and a logical standby database in Boston. The primary database is the SALES database. For a single instance database, the ORACLE\_SID parameter values are SALES, SALES\_PHYS, and SALES\_LOG. In a RAC configuration, the corresponding instance number is appended to each of the ORACLE\_SID parameter values.

Table A–1 shows generic best practice SPFILE parameters for primary, physical standby, and logical standby databases

 Table A-1
 Generic SPFILE Parameters for Primary, Physical Standby, and Logical Standby Databases

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.COMPATIBLE='10.2.0'	Same as Chicago	Same as Chicago
<pre>*.CONTROL_FILES= '_+DATA/SALES/controlfiles/ control.265.263563526', '+RECO/SALES/controlfiles/ control.276.263563526'</pre>	*.CONTROL_FILES= '+DATA/SALES/controlfiles/ backup.474.3736463483', '+RECO/SALES/cnortolfiles/ backup.363.3736463483'	*.CONTROL_FILES= '+DATA/SALES_LOG/controlfiles/ backup.354.25365373', '+RECO/SALES_LOG/controlfiles/ backup.352.25365373'
*.CONTROL_FILE_RECORD_KEEP_TIME=30	Same as Chicago	Same as Chicago
*.DB_NAME='SALES'	Same as Chicago	*.DB_NAME='SALES_LOG'
*.DB_CREATE_FILE_DEST=+DATA	Same as Chicago	Same as Chicago
*.DB_RECOVERY_FILE_DEST=+RECO	Same as Chicago	Same as Chicago
*.DB_RECOVERY_FILE_DEST_SIZE=100G	Same as Chicago	Same as Chicago
*.DB_FLASHBACK_RETENTION_TARGET=240	Same as Chicago	Same as Chicago
*.BACKGROUND_CORE_DUMP=FULL	Same as Chicago	Same as Chicago
*.BACKGROUND_DUMP_DEST= 'mnt/app/oracle/admin/SALES/bdump'	*.BACKGROUND_DUMP_DEST= 'mnt/app/oracle/admin/ SALES/bdump'	*.BACKGROUND_DUMP_DEST= 'mnt/app/oracle/admin/ SALES_LOG/bdump'
*.CORE_DUMP_DEST= '/mnt/app/oracle/admin/SALES/cdump'	*.CORE_DUMP_DEST= '/mnt/app/oracle/admin/ SALES/cdump'	*.CORE_DUMP_DEST= '/mnt/app/oracle/admin/ SALES_LOG/cdump'
*.USER_DUMP_DEST= '/mnt/app/oracle/admin/SALES/udump'	*.USER_DUMP_DEST= '/mnt/app/oracle/admin/ SALES/udump'	*.USER_DUMP_DEST= '/mnt/app/oracle/admin/ SALES_LOG/udump'
*.DB_BLOCK_CHECKING=MEDIUM	Same as Chicago <sup>1</sup>	Same as Chicago
*.DB_BLOCK_CHECKSUM=FULL	Same as Chicago	Same as Chicago
*.LOG_ARCHIVE_FORMAT= 'arch_%t_%S_%r.log'	Same as Chicago	Same as Chicago
*.LOG_ARCHIVE_TRACE=0	Same as Chicago	Same as Chicago
*.FAST_START_MTTR_TARGET=300	Same as Chicago	Same as Chicago
*.STATISTICS_LEVEL=TYPICAL	Same as Chicago	Same as Chicago
*.LOCAL_LISTENER='SALES_lsnr'	Same as Chicago	Same as Chicago

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.REMOTE_LISTENER= 'SALES_remotelsnr_CHICAGO'	*.REMOTE_LISTENER= 'SALES_remotelsnr_BOSTON'	*.REMOTE_LISTENER= 'SALES_remotelsnr_BOSTON'
*.UNDO_MANAGEMENT=AUTO	Same as Chicago	Same as Chicago
*.UNDO_RETENTION=900	Same as Chicago	Same as Chicago
*.UNDO_TABLESPACE='UNDOTBS'	Same as Chicago	Same as Chicago
*.RESUMABLE_TIMEOUT=900	Same as Chicago	Same as Chicago

Table A–1 (Cont.) Generic SPFILE Parameters for Primary, Physical Standby, and Logical Standby

<sup>1</sup> This can be turned off by setting DB\_BLOCK\_CHECKING=FALSE if recovery performance is adversely affected.

Table A–2 shows RAC best practice SPFILE parameters for primary, physical standby, and logical standby databases.

Table A–2 RAC SPFILE Parameters for Primary, Physical Standby, and Logical Standby Databases

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.CLUSTER_DATABASE=TRUE	Same as Chicago	Same as Chicago
SALES1.THREAD=1	SALES_PHYS1.THREAD=1	SALES_LOG1.THREAD=1
SALES2.THREAD=2	SALES_PHYS2.THREAD=2	SALES_LOG2.THREAD=2
SALES1.INSTANCE_NUMBER=1	SALES_PHYS1.INSTANCE_NUMBER=1	SALES_LOG1.INSTANCE_NUMBER=1
SALES2.INSTANCE_NUMBER=2	SALES_PHYS2.INSTANCE_NUMBER=2	SALES_LOG2.INSTANCE_NUMBER=2
SALES1.INSTANCE_NAME= SALES_CHICAGO1	SALES_PHYS1.INSTANCE_NAME= SALES_BOSTON1	SALES_LOG1.INSTANCE_NAME= SALES_BOSTON_LOG1
SALES2.INSTANCE_NAME= SALES_CHICAGO2	SALES_PHYS2.INSTANCE_NAME= SALES_BOSTON2	SALES_LOG2.INSTANCE_NAME= SALES_BOSTON_LOG2
SALES1.UNDO_TABLESPACE= 'UNDOTBS1'	SALES_PHYS1.UNDO_TABLESPACE= 'UNDOTBS1'	SALES_LOG1.UNDO_TABLESPACE= 'UNDOTBS1'
SALES2.UNDO_TABLESPACE= 'UNDOTBS2'	SALES_PHYS2.UNDO_TABLESPACE= 'UNDOTBS2'	SALES_LOG2.UNDO_TABLESPACE= 'UNDOTBS2'

Table A–3 shows Data Guard best practice SPFILE parameters for primary, physical standby, and logical standby databases. These parameters must be set whether or not you use the Data Guard broker.

Table A–3 Data Guard SPFILE Parameters for Primary, Physical Standby, and Logical Standby Databases

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.DB_UNIQUE_NAME='SALES_CHICAGO'	*.DB_UNIQUE_NAME='SALES_BOSTON'	*.DB_UNIQUE_NAME='SALES_BOSTON_LOG'

Table A–4 shows Data Guard best practice SPFILE parameters for primary database and physical standby database only. If you are using Data Guard Broker to manage your database environment, then you need set *only* the values in Table A–3 and Table A–4.

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.DB_BROKER_CONFIG_FILE_1= '+DATA/SALES_CHICAGO/dr1SALES_ CHICAGO.dat'	*.DB_BROKER_CONFIG_FILE_1= '+DATA/SALES_ BOSTON/dr1SALES_ BOSTON.dat'	*.DB_BROKER_CONFIG_FILE_1= '+DATA/SALES_BOSTON_ LOG/dr1SALES_ BOSTON_LOG.dat'
*.DB_BROKER_CONFIG_FILE_2= '+DATA/SALES_CHICAGO/dr2SALES_ CHICAGO.dat'	*.DB_BROKER_CONFIG_FILE_2= '+DATA/SALES_ BOSTON/dr2SALES_ BOSTON.dat'	*.DB_BROKER_CONFIG_FILE_2= '+DATA/SALES_BOSTON_ LOG/dr2SALES_ BOSTON_LOG.dat'
*.DG_BROKER_START=TRUE	Same as Chicago	Same as Chicago

 Table A-4
 Data Guard Broker SPFILE Parameters for Primary, Physical Standby, and Logical Standby

 Databases
 Databases

Table A–5 shows Data Guard best practice SPFILE parameters for primary, physical standby, and logical standby databases if you are *not* using Data Guard Broker to manage your database environment. If you are not using Data Guard Broker, you must also set the parameters in Table A–6 through Table A–9.

 Table A–5
 Data Guard (No Broker) SPFILE Parameters for Primary, Physical Standby, and Logical Standby

 Databases

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.LOG_FILE_NAME_CONVERT=' ',' '	Same as Chicago	Same as Chicago
*.STANDBY_FILE_MANAGEMENT=AUTO	Same as Chicago	Same as Chicago
*.REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE	Same as Chicago	Same as Chicago

Table A–6 shows Data Guard best practice SPFILE parameters for primary and physical standby databases only. You must set these parameters if you are not using Data Guard Broker to manage your database environment.

Table A–6	Data Guard SPFILE Parameters for Primar	y and Physical Standby Database Only
-----------	---	--------------------------------------

Chicago (Primary Database)	Boston (Physical Standby Database)
*.FAL_CLIENT='SALES_CHICAGO'	*.FAL_CLIENT='SALES_BOSTON'
*.FAL_SERVER='SALES_BOSTON'	*.FAL_SERVER='SALES_CHICAGO'
*.DB_UNIQUE_NAME='SALES_CHICAGO'	*.DB_UNIQUE_NAME='SALES_BOSTON'
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(SALES_ CHICAGO,SALES_BOSTON)'	Same as Chicago
*.LOG_ARCHIVE_DEST_1='location=USE_DB_ RECOVERY_FILE_DEST arch mandatory valid_ for=(ALL_LOGFILES,ALL_ROLES) db_unique_ name=SALES_CHICAGO'	*.LOG_ARCHIVE_DEST_1='location=USE_DB_ RECOVERY_FILE_DEST arch mandatory valid_ for=(ALL_LOGFILES,ALL_ROLES) db_unique_ name=SALES_BOSTON'
*.LOG_ARCHIVE_DEST_2='service=SALES_ BOSTON lgwr sync affirm net_timeout=30 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_BOSTON'	*.LOG_ARCHIVE_DEST_2='service=SALES_ CHICAGO lgwr sync affirm net_timeout=30 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_CHICAGO'

Table A–7 shows Data Guard best practice SPFILE parameters for primary and logical standby databases only. You must set these parameters if you are not using Data Guard Broker to manage your database environment.

Chicago (Primary Database)	Boston (Logical Standby Database)
*.FAL_CLIENT='SALES_CHICAGO'	*.FAL_CLIENT='SALES_BOSTON_LOG'
*.FAL_SERVER='SALES_BOSTON_LOG'	*.FAL_SERVER='SALES_CHICAGO'
*.LOG_ARCHIVE_CONFIG= 'DG_CONFIG=(SALES_CHICAGO,SALES_BOSTON_LOG)'	Same as Chicago
*.STANDBY_ARCHIVE_DEST=+RECO/SALES_ CHICAGO/archivelog/FAL/	*.STANDBY_ARCHIVE_DEST=+RECO/SALES_ BOSTON/archivelog/FAL/
<pre>*.LOG_ARCHIVE_DEST_1= 'location=USE_DB_RECOVERY_FILE_DEST arch max_failure=0 mandatory valid_for=(ONLINE_LOGFILES,ALL_ROLES) db_unique_name=SALES_CHICAGO'</pre>	*.LOG_ARCHIVE_DEST_1= 'location=USE_DB_RECOVERY_FILE_DEST arch max_failure=0 mandatory valid_for=(ONLINE_LOGFILES,ALL_ROLES) db_unique_name=SALES_BOSTON_LOG'
*.LOG_ARCHIVE_DEST_2= 'service=SALES_BOSTON_LOG reopen=15 max_ failure=10 lgwr sync affirm net_timeout=30 valid_for=(ONLINE_LOGFILES, PRIMARY_ROLE) db_ unique_name=SALES_BOSTON_LOG'	*.LOG_ARCHIVE_DEST_2= 'service=SALES_CHICAGO lgwr sync affirm net_timeout=30 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_CHICAGO'
<pre>*.LOG_ARCHIVE_DEST_3 = 'location=+RECO/SALES_ CHICAGO/archivelog/SRL/ arch max_ failure=0 mandatory valid_for=(STANDBY_ LOGFILES,STANDBY_ROLE) db_unique_ name=SALES_CHICAGO'</pre>	*.LOG_ARCHIVE_DEST_3= 'location=+RECO/SALES_ BOSTON/archivelog/SRL/ arch max_failure=0 mandatory valid_for=(STANDBY_ LOGFILES,STANDBY_ROLE) db_unique_ name=SALES_BOSTON_LOG'

Table A–7 Data Guard SPFILE Parameters for Primary and Logical Standby Database Only

Table A–8 applies to a Data Guard environment running in either maximum availability mode or maximum protection mode.

Table A-8Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, and LogicalStandby Database: Maximum Availability or Maximum Protection Modes

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.FAL_CLIENT='SALES_CHICAGO'	*.FAL_CLIENT='SALES_BOSTON'	*.FAL_CLIENT='SALES_BOSTON_LOG'
*.FAL_SERVER= 'SALES_BOSTON','SALES_BOSTON_LOG'	*.FAL_SERVER= 'SALES_CHICAGO','SALES_BOSTON_ LOG'	*.FAL_SERVER= 'SALES_CHICAGO','SALES_BOSTON'
*.LOG_ARCHIVE_CONFIG= 'DG_CONFIG=(SALES_CHICAGO, SALES_BOSTON,SALES_BOSTON_LOG)'	Same as Chicago	Same as Chicago
*.STANDBY_ARCHIVE_DEST= '+RECO/SALES_ CHICAGO/archivelog/FAL/'	*.STANDBY_ARCHIVE_DEST= '+RECO/SALES_ BOSTON/archivelog/FAL/'	*.STANDBY_ARCHIVE_DEST= '+RECO/SALES_BOSTON_ LOG/archivelog/FAL/
*.LOG_ARCHIVE_DEST_1= 'location=USE_DB_RECOVERY_FILE_ DEST arch mandatory valid_for=(ONLINE_ LOGFILES,ALL_ROLES) db_unique_ name=SALES_CHICAGO'	*.LOG_ARCHIVE_DEST_1= 'location=USE_DB_RECOVERY_FILE_ DEST arch mandatory valid_for=(ONLINE_ LOGFILES,ALL_ROLES) db_unique_ name=SALES_BOSTON'	*.LOG_ARCHIVE_DEST_1= 'location=USE_DB_RECOVERY_FILE_ DEST arch max_failure=0 mandatory valid_for=(ONLINE_ LOGFILES,ALL_ROLES) db_unique_ name=SALES_BOSTON_LOG'

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.LOG_ARCHIVE_DEST_2= 'service=SALES_BOSTON lgwr sync affirm net_timeout=30 valid_ for=(ONLINE_LOGFILES,ALL_ROLES) db_unique_name=SALES_BOSTON'	*.LOG_ARCHIVE_DEST_2= 'service=SALES_CHICAGO lgwr sync affirm net_timeout=30 valid_ for=(ONLINE_LOGFILES,ALL_ROLES) db_unique_name=SALES_CHICAGO'	Not applicable
*.LOG_ARCHIVE_DEST_3= 'service=SALES_BOSTON_LOG lgwr sync affirm net_timeout=30 valid_for=(ONLINE_ LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_BOSTON_LOG'	*.LOG_ARCHIVE_DEST_3= 'service=SALES_BOSTON_LOG lgwr sync affirm net_ timeout=30 valid_for=(ONLINE_ LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_BOSTON_LOG'	*.LOG_ARCHIVE_DEST_3 ='service=SALES_CHICAGO lgwr sync affirm net_timeout=30 valid_for=(ONLINE_ LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_CHICAGO'
*.LOG_ARCHIVE_DEST_4= 'location=+RECO/SALES_ CHICAGO/archivelog/ SRL/ arch mandatory valid_for=(STANDBY_ LOGFILES,STAMDBY_ROLE) db_unique_name=SALES_CHICAGO'	*.LOG_ARCHIVE_DEST_4= 'location=+RECO/SALES_ BOSTON/archivelog/ SRL/ arch mandatory valid_for=(STANDBY_ LOGFILES,STANDBY_ROLE) db_unique_name=SALES_BOSTON'	*.LOG_ARCHIVE_DEST_4= 'location=+RECO/SALES_BOSTON_LOG/ archivelog/SRL/ arch mandatory valid_ for=(STANDBY_LOGFILES,STANDBY_ ROLES) db_unique_name=SALES_ BOSTON_LOG'
*.PARALLEL_MAX_SERVERS=9	Same as Chicago	Same as Chicago

 Table A–8 (Cont.) Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, and

 Logical Standby Database: Maximum Availability or Maximum Protection Modes

Table A–9 shows how to change the parameters for a Data Guard environment that is running in maximum performance mode.

 Table A–9
 Data Guard SPFILE Parameters for Primary Database, Physical Standby Database, and Logical Standby Database: Maximum Performance Mode

Chicago (Primary Database)	Boston (Physical Standby)	Boston (Logical Standby)
*.LOG_ARCHIVE_DEST_2= 'service=SALES_BOSTON lgwr async net_timeout=30 valid_for=(ONLINE_ LOGFILES,ALL_ROLES) db_unique_name=SALES_BOSTON'	*.LOG_ARCHIVE_DEST_2= 'service=SALES_CHICAGO lgwr async net_timeout=30 valid_ for=(ONLINE_LOGFILES,ALL_ROLES) db_unique_name=SALES_CHICAGO'	Not applicable
*.LOG_ARCHIVE_DEST_3= 'service=SALES_BOSTON_LOG lgwr async net_timeout=30 valid_for= (ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_BOSTON_LOG'	*.LOG_ARCHIVE_DEST_3= 'service=SALES_BOSTON_LOG lgwr async net_timeout=30 valid_for= (ONLINE_LOGFILES, PRIMARY_ROLE) db_unique_name=SALES_BOSTON_ LOG'	*.LOG_ARCHIVE_DEST_3= 'service=SALES_CHICAGO lgwr async net_timeout=30 valid_for= (ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=SALES_CHICAGO'

# A.2 Oracle Net Configuration Files

This section contains examples of the following Oracle Net configuration file settings:

- SQLNET.ORA Example for All Hosts Using Dynamic Instance Registration
- LISTENER.ORA Example for All Hosts Using Dynamic Instance Registration
- TNSNAMES.ORA Example for All Hosts Using Dynamic Instance Registration

# A.2.1 SQLNET.ORA Example for All Hosts Using Dynamic Instance Registration

# Set dead connection time

SQLNET.EXPIRE\_TIME = 1
# Disable Nagle's algorithm
TCP.NODELAY=yes
# Set default SDU for all connections
DEFAULT\_SDU\_SIZE=32767

See Also: Oracle Database 10g Release 2 Best Practices: Data Guard Redo Apply and Media Recovery located on the OTN web site at http://www.oracle.com/technology/deploy/availability /htdocs/maa.htm

This white paper contains instructions for calculating your bandwidth delay product.

## A.2.2 LISTENER.ORA Example for All Hosts Using Dynamic Instance Registration

For a RAC environment, listeners must be listening on the virtual IP addresses (VIP), rather than the local host name.

```
lsnr_SALES =
 (DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS_LIST=
        (ADDRESS=(PROTOCOL=tcp)(HOST=<local_host_name>)(PORT=1513)
              (QUEUESIZE=1024)))))
PASSWORDS_lsnr_SALES = 876EAE4513718ED9
# Prevent listener administration
ADMIN_RESTRICTIONS_lsnr_SALES=ON
```

**See Also:** Oracle Database Net Services Administrator's Guide for more information on listener password protection

## A.2.3 TNSNAMES.ORA Example for All Hosts Using Dynamic Instance Registration

```
# Used for database parameter local_listener
SALES_lsnr =
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(PORT=1513)))
SALES_remotelsnr_CHICAGO =
  (DESCRIPTION=
    (ADDRESS_LIST=
       (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<chicago_host1>))
       (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<chicago_host2>)))
SALES_remotelsnr_BOSTON =
   (DESCRIPTION=
     (ADDRESS_LIST=
       (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<boston_host1>))
       (ADDRESS=(PROTOCOL=tcp) (PORT=1513) (HOST=<boston_host2>)))
# Net service used for communication with SALES database in Chicago
SALES_CHICAGO =
   (DESCRIPTION=
     (ADDRESS_LIST=
       (SEND_BUF_SIZE=4665000) (RECV_BUF_SIZE=4665000)
       (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<chicago_host1>))
       (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<chicago_host2>)))
     (CONNECT_DATA=(SERVICE_NAME=SALES_CHICAGO)))
# Net service used for communication with SALES database in Boston
SALES BOSTON =
   (DESCRIPTION=
```

```
(ADDRESS_LIST=
  (SEND_BUF_SIZE=4665000)(RECV_BUF_SIZE=4665000)
  (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<boston_host1>))
  (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<boston_host2>)))
  (CONNECT_DATA=(SERVICE_NAME=SALES_BOSTON)))
# Net service used for communication with Logical Standby SALES database in Boston
SALES_BOSTON_LOG =
  (DESCRIPTION=
  (ADDRESS_LIST=
   (SEND_BUF_SIZE=4665000)(RECV_BUF_SIZE=4665000)
   (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<boston_host1>))
   (ADDRESS=(PROTOCOL=tcp)(PORT=1513)(HOST=<boston_host2>)))
   (CONNECT_DATA=(SERVICE_NAME=SALES_BOSTON_LOG)))
```

# Glossary

#### 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.

#### **MAA** environment

An architecture that provides the most comprehensive set of solutions for both unplanned and planned outages because it inherits the capabilities and advantages of both the Oracle Database 10g with RAC and the Oracle Database 10g with Data Guard.

The MAA environment consists of a site containing a RAC production database and a second site containing a cluster that hosts both logical and physical standby databases, or at least one physical or logical standby 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.

# Index

## Α

alerts Enterprise Manager, 3-2 ALTER SESSION ENABLE RESUMABLE statement, 2-15 ANALYZE TABLE tablename VALIDATE STRUCTURE CASCADE, 2-12 applications datatype support, 2-22 defining as services, 2-16 failover, 4-25 fast failover, 2-53 to 2-55 managing with Oracle Clusterware, 2-17 service brownouts, 3-7 tracking performance with Beacon, 3-4 upgrades, 4-66 architecture high availability, 1-1 ARCHIVELOG mode, 2-8 archiving strategy, 2-27 ASM See Automatic Storage Management (ASM) asm\_diskstring parameter, 2-5 ASMLib, 2-3 asynchronous disk I/O, 2-12 Automatic Database Diagnostic Monitor (ADDM), 2-13 automatic performance tuning, 2-13 automatic segment space management, 2-15 using, 2-15 Automatic Shared Memory Management, 2-12 Automatic Storage Management (ASM) and HARD-compliant storage, 2-7 asm\_diskstring parameter, 2-5 ASMLib, 2-3 database file management, 2-2 disk device allocation, 2-4 disk group size, 2-6 migrating databases to and from, 2-3, 4-57 recovery, 4-25 redundancy, 2-5 with disk multipathing software, 2-5 automatic undo management described, 2-14 Automatic Workload Repository (AWR), 2-13

best practices, 2-13 evaluating performance requirements, 2-2 AWR See Automatic Workload Repository (AWR)

## В

backup and recovery best practices, 2-45 to 2-53 checksums calculated during, 2-9 enabling with ARCHIVELOG mode, 2-8 recommendations, 2-45 backups configuring, 2-45 to 2-53 OCR, 2-18 Beacons, 3-4 configuring, 3-4 benefits Grid Control, 1-3 high availability best practices, 1-1 logical standby database, 2-22 physical standby database, 2-21 best practices, 1-1 AWR, 2-13 backup and recovery, 2-45 to 2-53 Data Guard configurations, 2-20 to 2-45 failover (fast-start), 2-40, 4-15 failover (manual), 2-41, 4-15, 4-18 operational, 1-2 to 1-3 RAC configurations, 2-16, 2-16 to 2-19 security policy, 1-3 storage subsystems, 2-1 switchover, 2-38, 4-20 upgrades, 4-59 block checking preventing corruption, 2-11 block checksums, 2-9 block media recovery, 4-36 block validation end-to-end, 2-7 broker benefits, 2-26 brownouts, 3-7

## С

checksums, 2-9 client failover best practices, 2-54 clients application failover, 4-25 configuring for failover, 2-54 load balancing, 2-17 Cluster Ready Services (CRS) described, 4-46 recovering service availability, 4-46 clusters management, 2-17 clusterwide outage restoring the standby database after, 4-53 complete site failover recovery time objective (RTO), 4-10 configurations RAC, 2-16 connect-time failover, 4-47 corruption detecing in-memory, 2-9 detecting, 2-9 detecting redo record, 2-9 recovery, 4-34 corruptions preventing memory and data, 2-11 crash recovery bounded, 2-12 CREATE DISKGROUP statement examples, 2-3, 2-5 CRS See Cluster Ready Services (CRS) CRSD process OCR backups, 2-18

## D

data protecting against corruption, 2-7 protecting outside of the database, 2-43 data area disk group failure recovery options, 4-30 data corruption HARD protection, 2-7, 4-4 recovery with Data Guard, 4-36 data failure manual re-creation, 4-37 recovery, 4-34, 4-36 restoring fault tolerance on standby database, 4-54 RMAN block media recovery, 4-36 RMAN datafile media recovery, 4-37 Data Guard adding to a RAC primary, 5-2 archiving strategies, 2-27 broker, 2-26 choosing a standby type, 2-22 cloning physical standby databases, 2-41 failover

best practices (fast-start), 2-40 best practices (manual), 2-41 recovery for data area disk group failures, 4-30 using Enterprise Manager, 4-15 using SOL\*Plus, 4-18 to 4-19 when to perform, 4-14 log apply services, 2-33 to 2-37 managing targets, 3-10 monitoring, 3-8 performance, 2-43 recovery from data corruption and data failure, 4-36 redo transport services, 2-29 to 2-33 restoring standby databases, 4-50 role transitions, 2-37 to 2-41 switchover best practices, 2-38 using Enterprise Manager, 4-20 using SQL\*Plus, 4-22 to 4-23 data protection modes, 2-23 to 2-24 Data Pump moving the contents of the SYSTEM tablespace, 4-66 data-area disk group failure See Also Data Guard failover, fast-start failover, local recovery database area contents, 2-3 disk partitioning, 2-4 database block checking, 2-11 database configuration recommendations, 2-7 database files ASM integration, 2-2 management optimizations, 2-2 recovery related, 2-3 database patch upgrades recommendations, 4-59 Database Resource Manager, 2-15 Database Upgrade Assistant (DBUA), 4-61 databases configuring maximum RAC instance, 2-19 evaluating performance requirements, 2-2 migration, 4-64 object reorganization, 4-68 resolving inconsistencies, 4-42 switching primary and standby roles amongst, 4-19 upgrades upgrades methods, 4-61 datafile block corruption recovery, 4-34 datatypes supported by standby databases, 2-22 DB\_BLOCK\_CHECKING initialization parameter, 2-11 DB\_BLOCK\_CHECKSUM initialization parameter, 2-9

DB\_CACHE\_SIZE initialization parameter, 2-34 DB\_CREATE\_FILE\_DEST initialization parameter enabling Oracle managed files (OMF), 2-3 DB\_FLASHBACK\_RETENTION\_TARGET initialization parameter, 2-11 DB\_KEEP\_CACHE\_SIZE initialization parameter, 2-34 DB\_RECOVERY\_FILE\_DEST initialization parameter enabling Oracle managed files (OMF), 2-3 flash recovery area, 2-10 DB\_RECOVERY\_FILE\_DEST\_SIZE initialization parameter enabling Oracle managed files (OMF), 2-3 limit for flash recovery area, 2-10 DB\_RECYCLE\_CACHE\_SIZE initialization parameter, 2-34 DBCA balancing client connections, 2-17 DBMS\_FILE\_TRANSFER package, 4-57 DBMS\_REDEFINITION PL/SQL package, 4-68 DBVERIFY utility, 2-12 DEFAULT TEMPORARY TABLESPACE clause of the CREATE DATABASE statement, 2-15 Device Mapper disk multipathing, 2-5 disabling parallel recovery, 2-12 disaster-recovery site distanced from the primary site, 2-24 disk devices configuration, 2-3, 2-5 multipathing, 2-5 naming asm\_diskstring parameter, 2-5 ASMLib, 2-3 partitioning for ASM, 2-4 protecting from failures, 2-5 disk failures protection from, 2-5 restoring redundancy after, 2-6 disk group configuration, 2-3 disk groups determining proper size of, 2-6 determining size of, 2-6 disk multipathing, 2-5 DISK\_ASYNCH\_IO initialization parameter, 2-12, 2-35distances between the disaster-recovery site and the primary site, 2-24 DNS failover, 4-12 downtime reducing, 1-2 dropped tablespace using Flashback Database, 4-43 dropping database objects, 4-39 dual failures restoring, 4-57 dynamic instance registration LISTENER.ORA file example, A-7 SQLNET.ORA file example, A-6

TNSNAMES.ORA file example, A-7

## Е

enabling, 2-11 endian format determining, 4-64 end-to-end block validation, 2-7 **Enterprise Manager** alerts, 3-2 Data Guard failover, 4-15 Data Guard switchover, 4-20 Database Targets page, 3-6 managing patches, 3-9 metrics, 3-2, 3-8 notification rules, 3-3, 3-7 performance, 3-6 equations setting PARALLEL\_MIN\_SERVERS, 2-12 standby redo log files, 2-29 EXTERNAL REDUNDANCY clause on the CREATE DISKGROUP statement, 2-5

## F

failover (fast-start) best practices, 2-40 reinstating the former primary database, 4-51 failover (manual) best practices, 2-41 when to perform, 4-14 failovers application, 4-25 complete site, 4-10 defined, 4-13 described, 4-15 effect on network routes, 4-10 Enterprise Manager, 4-15 fast application, 2-53 to 2-55 nondisruptive, 2-5 restoring standby databases after, 4-50 failure groups defining, 2-6 failures space allocation, 2-15 fast application failover, 2-53 to 2-55 Fast Application Notification (FAN), 4-25 FAST\_START\_MTTR\_TARGET initialization parameter, 2-12, 2-35 controlling instance recovery time, 2-11 fast-start fault recovery instance recovery, 2-11 fault tolerance configuring storage subsystems, 2-1 restoring, 4-44 to 4-57 restoring after OPEN RESETLOGS, 4-55 flash recovery area contents, 2-3 disk group failures, 4-31 disk partitioning, 2-4

local recovery steps, 4-33 local restart steps, 4-32 low-cost storage grid, 2-2 using, 2-10 Flashback Database, 4-38, 4-42 correcting human errors, 2-11 enabling, 2-10 in Data Guard configurations, 2-25 setting maximum memory, 2-12 Flashback Drop, 4-38, 4-39 flashback free buffer waits wait event, 2-10 Flashback Query, 4-38, 4-40 Flashback Table, 4-38, 4-39 flashback technologies resolving database-wide inconsistencies, 4-42 flashback technology example, 4-40 recovering from user error, 4-37 solutions, 4-38 Flashback Transaction Query, 4-38, 4-40 Flashback Version Query, 4-38, 4-40 FORCE LOGGING mode, 2-25

# G

Grid Control See Oracle Grid Control, Enterprise Manager GV\$SYSSTAT view gathering workload statistics, 2-2

## Η

HARD See Hardware Assisted Resilient Data (HARD) Hardware Assisted Resilient Data (HARD) data corruption, preventing, 2-7, 4-4 when using ASM, 2-7 high availability described, 1-1 requirements, 2-8 to 2-12 restoring after fast-start failover, 4-51 host bus adapters (HBA) load balancing across, 2-5 hosts using dynamic instance registration LISTENER.ORA file example, A-7 SQLNET.ORA file example, A-6 TNSNAMES.ORA file example, A-7 HR service scenarios, 4-46 human errors correcting with Flashback Database, 2-11 recovery, 4-37

## 

index blocks, 2-11 index rebuilding, 4-68 initialization parameters primary and physical standby example, 2-28 in-memory block change checking, 2-11 in-memory corruption detecting, 2-9 instance failure recovery, 2-11 instance failures single, 4-24 instance recovery bounded, 2-12 controlling with fast-start fault recovery, 2-11 parallel execution, 2-12 interconnect subnet verification with Oracle ORADEBUG utility, 2-19 interim patch upgrade, 4-58 I/O bandwidth for flash recovery area, 2-10 I/O operations load balancing, 2-5 tuning, 2-35

## L

library ASMLib support for ASM, 2-3 LISTENER.ORA file sample, A-7 listeners balancing clients across, 2-17 LISTENER.ORA file example, A-7 SQLNET.ORA file example, A-6 TNSNAMES.ORA file example, A-7 load balancing client connections, 2-17 I/O operations, 2-5 through disk multipathing, 2-5 LOAD\_BALANCE parameter balancing client connections, 2-17 load-balancing application services, 4-47 local archiving first, 2-27 local recovery for data area disk group failures, 4-30 for flash recovery area disk group failures, 4-33 local restart steps for fast recovery of flash recovery area, 4-32 locally managed tablespaces, 2-15 described, 2-15 location migration, 4-63 log apply services best practices, 2-33 to 2-37 LOG\_ARCHIVE\_FORMAT initialization parameter, 2-27 LOG\_ARCHIVE\_LOCAL\_FIRST initialization parameter, 2-27 LOG\_BUFFER initialization parameter, 2-12 logical standby databases benefits of, 2-22 failover, 4-19 switchover, 4-23 upgrades on, 4-62 logical unit numbers (LUNs), 2-5 defined, Glossary-1 low-cost storage subsystems, 2-2

LUNs

See Also logical unit numbers (LUNs) See logical unit numbers (LUNs), 2-5

#### Μ

MAA See Oracle Maximum Availability Architecture (MAA) manageability improving, 2-13 to 2-16 manual failover best practices, 2-41, 4-18 when to perform, 4-14 maximum availability mode described, 2-23 redo transport requirements, 2-30 when to use, 2-24 maximum performance mode described, 2-23 redo transport requirements, 2-30 when to use, 2-24 maximum protection mode described, 2-23 initialization parameter example, 2-28 when to use, 2-24 media failure recovery, 4-34 memory management, 2-12 metrics Enterprise Manager, 3-2 migrating Data Guard to a RAC primary, 5-2 databases to and from ASM, 2-3 to MAA, 5-1 to RAC from a single instance, 5-2 Transportable Database, 4-64 mirroring across storage arrays, 2-6 monitoring Oracle Grid Control, 1-3 rebalance operations, 4-58 multipathing (disks) path abstraction, 2-5

## Ν

Network Attached Storage (NAS), 2-35 network detection and failover CRS and RAC, 2-19 network routes after site failover, 4-11 before site failover, 4-10 node failures multiple, 4-24 node recovery bounded, 2-12 non database object corruption and recommended repair, 4-35 nondisruptive failovers, 2-5 nonindex organized-table blocks, 2-11 checking, 2-11 NORMAL REDUNDANCY clause on the CREATE DISKGROUP statement, 2-6 notification rules recommended, 3-7 SLA influence on monitoring, 3-4

# 0

OCR backups of, 2-18 described, 2-18 recovering, 4-24 ocrconfig -export command, 2-19 OMF See Oracle managed files online log groups minimum, 2-10 online object reorganization, 4-68 online redo log files multiplex, 2-9 Online Reorganization and Redefinition, 4-68 opatch command-line utility, 4-59 operational best practices, 1-2 to 1-3 ORA-1578 error, 2-9 ORA-16625 error, 4-15 Oracle Cluster Registry (OCR) See OCR Oracle Clusterware described, 2-17 OCR mirroring, 2-18 system maintenance, 4-70 Oracle Data Pump platform migrations, 4-65 Oracle Database 10g configuration recommendations, 2-7 Data Guard, 2-20 RAC configuration recommendations, 2-16 Oracle Grid Control benefits, 1-3 home page, 3-2 managing Data Guard targets, 3-10 Policy Violations, 3-9 Oracle managed files (OMF) database file management, 2-3 disk and disk group configuration, 2-3 flash recovery area, 2-10 Oracle Management Agent, 3-2 Oracle Maximum Availability Architecture (MAA) defined, Glossary-1 described, 1-2 environment, 5-1 web site, 1-2 Oracle Net configuration file examples, A-6 to A-8 Oracle ORADEBUG utility verifying interconnect subnet, 2-19 Oracle Resilient Low-Cost Storage Initiative, 2-2 Oracle Secure Backup

OCR backups, 2-18 Oracle Streams database migration, 4-65 online database upgrade, 4-67 upgrades using, 4-62 Oracle Universal Installer, 4-60 outages managing unscheduled and scheduled, 4-1 scheduled, 4-5 types, 4-2 unscheduled, 4-1

## Ρ

parallel executions, accelerating, 2-12 parallel recovery disabling, 2-12 PARALLEL\_EXECUTION\_MESSAGE\_SIZE initialization parameter, 2-34 setting, 2-12 PARALLEL\_MIN\_SERVERS initialization parameter tuning, 2-12 partitions allocating disks for ASM use, 2-4 patch sets rolling upgrades, 4-59 patches managing with Enterprise Manager, 3-9 path failures protection from, 2-5 performance application, tracking with Beacon, 3-4 asynchronous disk I/O, 2-12 automatic tuning, 2-13 Data Guard, 2-43 database, gathering requirements, 2-2 overhead of enabling checksums, 2-9 physical standby databases benefits of, 2-21 failover, 4-18 location migrations, 4-66 switchover, 4-22 using as a clone, 2-41 platform migration endian format for, 4-64 platform migrations, 4-61, 4-63 pool resizing, 2-12 primary database distance from the disaster-recovery site, 2-24 reinstating after a fast-start failover, 4-51 restoring fault tolerance, 4-55 protection modes described, 2-23 determining appropriate, 2-24 See Also data protection modes, maximum protection mode, maximum availability mode, maximum performance mode

## R

RAC adding Data Guard, 5-2 application failover, 4-25 client failover, 2-54 configurations, 2-16 migrating from a single instance, 5-2 network detection and failover, 2-19 recovery from unscheduled outages, 4-23 restoring failed nodes or instances, 4-45 rolling patch upgrade, 4-60 rolling upgrades, 4-58 system maintenance, 4-70 voting disk, 2-18 **RAC** environments adding disks to nodes, 2-3 LISTENER.ORA file sample for, A-7 SQLNET.ORA file sample for, A-6 TNSNAMES.ORA file sample for, A-7 RAID. 2-5 logical unit numbers (LUNs), 2-5 RAID protection, 2-5 rebalance operations ASM disk partitions, 2-4, 2-5 monitoring, 4-58 recommendations database configuration, 2-7 recovery bounded, 2-12 recovery files created in the recovery area location, 2-10 recovery point objective (RPO) defined, Glossary-1 for data area disk group failures, 4-30 recovery steps for scheduled outages, 4-7 to 4-9 recovery steps for unscheduled outages, 4-3 recovery time objective (RTO) defined, Glossary-1 described, 4-10 for data-area disk group failures, 4-30 RECOVERY\_ESTIMATED\_IOS initialization parameter for parallel recovery, 2-12 **RECOVERY\_PARALLELISM** initialization parameter, 2-12 redo log files and groups checksum, 2-9 size, 2-9 redo transport services best practices, 2-29 to 2-33 redundancy CREATE DISKGROUP DATA statement, 2-5 disk devices, 2-5 restoring after disk failures, 2-6 reinstatement, 4-51 remote archiving, 2-28 REMOTE\_ARCHIVE\_ENABLE initialization parameter, 2-27 resetlogs on production database restoring standby database, 4-55

resource management using Database Resource Manager, 2-15 restoring client connections, 4-46 failed instances, 4-45 failed nodes, 4-45 services, 4-46 resumable space allocation, 2-15 space allocation failures, 2-15 **RESUMABLE\_TIMEOUT** initialization parameter, 2-15 RMAN calculates checksums, 2-9 RMAN BACKUP command VALIDATE option, 2-12 RMAN BACKUP VALIDATE command, 4-36 RMAN block media recovery, 4-36 RMAN BLOCKRECOVER command, 4-36 RMAN data file media recovery, 4-37 RMAN datafile media recovery, 4-37 role transitions best practices, 2-37 to 2-41 role-based destinations, 2-28 rolling upgrades patch set, 4-59 row and transaction inconsistencies, 4-39 RPO See recovery point objective (RPO) RTO See recovery time objective (RPO)

# S

SALES scenarios setting initialization parameters, 2-28 SAME See stripe and mirror everything (SAME) scenarios ASM disk failure and repair, 4-27 Data Guard failover, 4-15 Data Guard switchover, 4-20 fast-start failover, 4-51 HR service, 4-46 object reorganization, 4-68 recovering from human error, 4-40 SALES, 2-28 verifying interconnect subnet, 2-19 scheduled outages described, 4-5 to 4-9 recovery from, 4-7 to 4-9 reducing downtime for, 4-57 to 4-70 types of, 4-5 See Also unscheduled outages secondary site outage restoring the standby database after, 4-53 security recommendations, 1-3 semantic block checking, 2-11 server parameter file

See SPFILE service availability recovering, 4-46 service level agreements (SLA), 1-2 effect on monitoring and notification, 3-4 operational best practices, 1-3 service tests and Beacons configuring, 3-4 services RAC application failover, 4-25 RAC application workloads, 2-16 tools for administration, 2-16 SGA\_TARGET initialization parameter, 2-12 site failover network routes, 4-11 SLA See service level agreements (SLA) sort operations improving, 2-15 space management, 2-15 SPFILE samples, A-2 to A-6 SQL Access Advisor, 2-13 SQL Tuning Advisor, 2-13 SQLNET.ORA file sample, A-6 standby databases choosing physical versus logical, 2-22 comparing logical and physical, 2-21 datatype support, 2-22 distance from the primary site, 2-24 restoring, 4-50 standby redo log files (SRLs) determining number of, 2-29 STANDBY\_ARCHIVE\_DEST initialization parameter, 2-28 Storage Area Network (SAN), 2-35 Storage array based RAID, 2-5 storage arrays mirroring across, 2-6 storage subsystems, 2-1 to 2-7 configuring ASM, 2-2 configuring redundancy, 2-5 performance requirements, 2-2 stripe and mirror everything (SAME), 2-2 switchover described, 4-19 Enterprise Manager, 4-20 See Also Data Guard to a logical standby database, 4-23 to a physical standby database, 4-22 system failure recovery, 2-11 system maintenance, 4-70 system resources assessing, 2-35 SYSTEM tablespace moving the contents of, 4-66

## Т

table inconsistencies, 4-39 tablespaces locally managed, 2-15 renaming, 4-68, 4-69 temporary, 2-15 TCP Nagle algorithm disabling, 2-32 temporary tablespaces, 2-15 test environments operational best practices for, 1-2 TNSNAMES.ORA file sample, A-7 Transportable Database, 4-64 transportable tablespaces database upgrades, 4-63 platform migration, 4-65

## U

undo retention tuning, 2-14undo space managing, 2-14 UNDO\_MANAGEMENT initialization parameter automatic undo management, 2-14 UNDO\_RETENTION initialization parameter automatic undo management, 2-14 UNDO\_TABLESPACE initialization parameter automatic undo management, 2-14 unscheduled outages described, 4-1 to 4-5 RAC recovery, 4-23 recovery from, 4-3, 4-9 to 4-44 types, 4-1 See Also scheduled outages upgrades application, 4-66 best practices, 4-59 Database Upgrade Assistant (DBUA), 4-61 USABLE\_FILE\_MB column on the V\$ASM\_DISKGROUP view, 2-6 user error flashback technology, 4-37

## V

V\$ASM\_DISK view, 2-35 V\$ASM\_DISKGROUP REQUIRED\_MIRROR\_FREE\_MB column, 2-6 V\$ASM\_DISKGROUP view REQUIRED\_MIRROR\_FREE\_MB column, 2-6 U\$ABLE\_FILE\_MB column, 2-6 V\$ASM\_OPERATION view monitoring rebalance operations, 4-58 V\$EVENT\_HISTOGRAM view, 2-34 V\$INSTANCE\_RECOVERY view tuning recovery processes, 2-12 V\$OSSTAT view, 2-35 V\$SESSION\_WAITS view, 2-34 V\$SYSTEM\_EVENT view, 2-35

V\$SYSTEM\_EVENTS view, 2-34 VALID\_FOR attribute, 2-28 VALIDATE option on the RMAN BACKUP command, 2-12 validation checksums during RMAN backup, 2-9 end-to-end block validation technology, 2-7 VIP address connecting to applications, 2-16 described, 2-17 during recovery, 4-46 workload management, 2-17 Virtual Internet Protocol (VIP) Address See AlsoVIP address Virtual Internet Protocol Configuration Assistant (VIPCA) VIP address configuration, 2-17 voting disk (RAC) best practices, 2-18

## W

wait events flashback free buffer waits, 2-10 web sites ASMLib, 2-3 MAA, 1-2 workload management connecting through VIP address, 2-17 workloads examples, 2-2 gathering statistics, 2-2