Oracle® Data Guard

Broker

10*g* Release 1 (10.1)

Part No. B10822-01

December 2003

Oracle Data Guard Broker, 10*g* Release 1 (10.1)

Part No. B10822-01

Primary Author: Rhonda Day

Contributors: Gary Allison, Pamela Bantis, Wei Chen, Sean Connolly, Ray Dutcher, Michael Harvey, Susan Hillson, Nitin Karkhanis, Sadhana Kyathappala, Steve Lee, Jiangbin Luo, Venkat Maddali, Bob McGuirk, Deborah Owens, Ashish Ray, Viv Schupmann, Stephen Vivian

Contents

5 Data Guard Scenarios - Using Oracle Enterprise Manager

6 Data Guard Scenarios - Using DGMGRL CLI

7 Data Guard Command-Line Interface Reference

9 Troubleshooting Data Guard

A Data Guard Broker Changed and Deprecated Features

Glossary

Index

List of Examples

List of Figures

List of Tables

Send Us Your Comments

Oracle Data Guard Broker, 10*g* Release 1 (10.1)

Part No. B10822-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: nedc-doc_us@oracle.com
- FAX: 603-897-3825 Attn: Oracle Data Guard Broker
- Postal service:
  Oracle
  Oracle Data Guard
  One Oracle Drive
  Nashua, NH 03062-2804
  U.S.A.

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This document provides information about Oracle Data Guard broker, a management and monitoring interface that helps you configure, monitor, and control an Oracle Data Guard broker configuration.

This preface contains these topics:

Audience

Oracle Data Guard Broker is intended for database administrators (DBAs) and system administrators who want to use the Oracle Data Guard broker to automate many of the tasks involved in configuring and monitoring an Oracle Data Guard configuration.

The discussions herein assume that readers are already familiar with Oracle Data Guard, Oracle Enterprise Manager, and the network services provided by Oracle Net Services.

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. 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 additional information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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.

Organization

This document contains:

Chapter 1, "Oracle Data Guard Broker Concepts"

This chapter introduces Oracle Data Guard broker concepts and terminology.

Chapter 2, "Managing Broker Configurations"

This chapter helps you set up and install Oracle Data Guard and configure a Data Guard broker configuration.

Chapter 3, "Managing Databases"

This chapter describes configuring and managing databases. It also describes states, status, and properties of databases.

Chapter 4, "Role Management"

This chapter describes managing database role transitions, including switchover, and failover.

Chapter 5, "Data Guard Scenarios - Using Oracle Enterprise Manager"

This chapter shows how to use the Data Guard graphical user interface (GUI) to create, manage, and monitor a broker configuration.

Chapter 6, "Data Guard Scenarios - Using DGMGRL CLI"

This chapter describes how to use the Data Guard command-line interface to create, manage, and monitor a broker configuration.

Chapter 7, "Data Guard Command-Line Interface Reference"

This chapter provides reference information for the Data Guard command-line interface.

Chapter 8, "Database Properties"

This chapter provides reference information about database properties.

Chapter 9, "Troubleshooting Data Guard"

This chapter provides troubleshooting information for Data Guard.

Appendix A, "Data Guard Broker Changed and Deprecated Features"

This appendix provides information about changed and deprecated features.

Glossary

Related Documentation

Refer to the following documentation for more information about Oracle Data Guard:

- *Oracle Data Guard Concepts and Administration*.
- Oracle release notes specific to your operating system.
- Oracle installation guide specific to your operating system.
- For more information about Oracle Data Guard GUI, see the online help available with this graphical user interface. To access the online help topics, click Help on the menu bar in Data Guard GUI.

Refer to the following documentation for information about related products:

- *Oracle Database Concepts*
- *Oracle Net Services Administrator's Guide*
- Oracle Enterprise Manager product documentation set

Printed documentation is available for sale in the Oracle Store at

```
http://oraclestore.oracle.com/
```

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

```
http://otn.oracle.com/membership/
```

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

```
http://otn.oracle.com/documentation/
```

Conventions

This section describes the conventions used in the text and code examples of this document. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`*` [ , `*`precision`*` ])`
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`

|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}`
		`[COMPRESS | NOCOMPRESS]`

. . .	Horizontal ellipsis points indicate either:	
	- That we have omitted parts of the code that are not directly related to the example	`CREATE TABLE ... AS `*`subquery`*`;`
	- That you can repeat a portion of the code	`SELECT `*`col1`*`, `*`col2`*`, ... , `*`coln`*` FROM employees;`
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.

Convention	Meaning	Example
`UPPERCASE monospace (fixed-width font)`	Uppercase monospace typeface indicates elements supplied by the system.	You can back up the database by using the `BACKUP` command. Use the `DBMS_STATS.GENERATE_STATS` procedure.
`lowercase monospace (fixed-width font)`	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements.	Enter `sqlplus` to open SQL*Plus. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table.
`lowercase monospace (fixed-width font) italic`	Lowercase monospace italic font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* refers to the release you installed prior to upgrading.
`MixedCase monospace (fixed-width font)`	Mixed-case monospace typeface indicates a Data Guard broker database property. The mixed case helps you visually differentiate a Data Guard broker property from other system-supplied elements, which are always shown in uppercase typeface. Mixed-case monospace typeface can also indicate other programmatic elements. Enter these elements as shown.	The `StandbyFileManagement` property corresponds to the `STANDBY_FILE_MANAGEMENT` initialization parameter. The `JRepUtil` class implements these methods.

What’s New in Oracle Data Guard Broker?

This section describes the new features of Oracle Data Guard broker release 10.1 and provides pointers to additional information.

Oracle Database Release 10.1 New Features in Data Guard Broker

Oracle Data Guard release 10.1 provides several new features that enhance your ability to centrally control, manage, and monitor a broker configuration.

This release provides the following new features:

- **Real Application Clusters (RAC) database support**

  Data Guard can be configured with Oracle Real Application Clusters (RAC) databases, which provides for multiple instances of a single, shared database. It is now possible to configure and support RAC databases in a Data Guard broker configuration using the Data Guard GUI or the Data Guard (DGMGRL) command-line interface.

  Data Guard broker is tightly integrated with Oracle Cluster Ready Services (CRS), a framework for high availability (HA) in a RAC database. CRS manages individual instances to provide unattended high availability of a given clustered database. The broker manages individual databases (clustered or otherwise) in a Data Guard configuration to provide disaster recovery in the event that CRS is unable to maintain availability of the primary database. Together, broker and CRS maximize protection and availability of your data. See Section 1.2, "Benefits of Data Guard Broker" and see *Oracle Real Application Clusters Administrator's Guide* for information about CRS.

- **Real-time apply support**

When you enable this feature using the `RealTimeApply` property, log apply services recover redo data from standby redo log files in real time (at the same time the log files are being filled up) as opposed to when a log switch occurs. Because real-time application of standby redo log files generally keeps the standby database caught up, this feature provides a number of benefits including quicker switchover and failover and instant up-to-date results after you change a physical standby database from applying redo to read-only operations. See Section 3.5.1, "Managing Real-Time Apply".

- **Automatic parallel apply support**

  The `ParallelApply` property accepts a new value, `AUTO`, which is the broker default value. This value allows the standby database to automatically select the number of parallel processes used for log apply services. In Release 9.0.2, you could only specify an explicit number of parallel processes for the apply service. See Section 3.5.3, "Managing Parallel Apply in Physical Standby Databases".

- **Enhanced health check status report**

  A more comprehensive health check status report is available through the monitorable properties or the GUI's Verify page.

- **Logical standby databases now support standby redo log files, which are required when Data Guard is configured to run in either maximum protection mode or maximum availability mode.**

- **New and improved GUI**

  - The Data Guard graphical user interface (GUI) is now HTML-based and is more integrated with Oracle Enterprise Manager than in previous releases. Because of its seamless integration into the new Web-based Enterprise Manager for Oracle Database 10*g*, the new Data Guard GUI provides an easier way to create, manage, and monitor your Data Guard environment. See Chapter 5, "Data Guard Scenarios - Using Oracle Enterprise Manager".

  - Standby redo log files are now created automatically when you change to a protection mode that requires them. See Section 3.6.1, "Setting the Protection Mode for Your Configuration" and Section 5.4.3, "Changing the Database Protection Mode".

- **Oracle Data Guard broker simplifies management of databases.**

  Data Guard broker supports database objects (previously known as database resource objects and site objects) and their instances.

- **Single command mode for Data Guard**

To improve the scripting power of the command-line mode (DGMGRL), you can now execute a single command. In single command mode, DGMGRL executes one command and exits upon the completion of the command. The exit code is the result of the command. See Section 7.1.1, "DGMGRL Optional Parameters".

- **New properties include:**
  - `AlternateLocation` (configurable property)
  - `ApplyInstanceTimeout` (configurable property)
  - `HostName` (configurable property)
  - `InitialConnectIdentifier` (configurable property)
  - `LatestLog` (monitorable property)
  - `LocalListenerAddress` (configurable property)
  - `NetTimeout` (configurable property)
  - `PreferredApplyInstance` (configurable property)
  - `RealTimeApply` (configurable property)
  - `SidName` (configurable property)
  - `StandbyArchiveLocation` (configurable property)
  - `StatusReport` (monitorable property)
  - `TopWaitEvents` (monitorable property)

  See Chapter 8, "Database Properties".

- **New CLI commands include:**
  - `ADD DATABASE`
  - `DISABLE DATABASE`
  - `EDIT CONFIGURATION (`*`protection mode`*`)`
  - `EDIT DATABASE (`*`property`*`)`
  - `EDIT DATABASE (`*`rename`*`)`
  - `EDIT DATABASE (`*`state`*`)`
  - `EDIT INSTANCE (AUTO PFILE)`
  - `EDIT INSTANCE (`*`property`*`)`

    - `ENABLE DATABASE`
    - `REMOVE DATABASE`
    - `REMOVE INSTANCE`
    - `SHOW DATABASE`
    - `SHOW INSTANCE`

  See Chapter 7, "Data Guard Command-Line Interface Reference".

- **Support for shared server connections on open databases**

  The broker now supports shared server connections to open databases from either of the Broker clients (the DGMGRL CLI and Data Guard GUI in Oracle Enterprise Manager).

1

Oracle Data Guard Broker Concepts

This chapter describes the Oracle Data Guard broker, its architecture and components, and how it automates the creation, control, and monitoring of a Data Guard configuration.

The following sections introduce Data Guard broker terminology and concepts:

- Section 1.1, "Oracle Data Guard Overview"
- Section 1.2, "Benefits of Data Guard Broker"
- Section 1.3, "Data Guard Broker Management Model"
- Section 1.4, "Data Guard Broker Components"
- Section 1.5, "Data Guard Broker User Interfaces"
- Section 1.6, "Data Guard Monitor"
- Section 1.7, "Oracle Data Guard Installation, Upgrade, Downgrade, and First Use"

See *Oracle Data Guard Concepts and Administration* for the definition of a Data Guard configuration and for complete information about Oracle Data Guard concepts and terminology.

1.1 Oracle Data Guard Overview

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. Data Guard maintains these standby databases as transactionally consistent copies of the primary database. If the primary database becomes unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus minimizing the downtime associated with the outage. Data Guard can be used with traditional backup, recovery, and cluster techniques, as well as the Flashback Database feature to provide a high level of data protection and data availability.

1.1.1 Oracle Data Guard Configuration Overview

A Data Guard configuration consists of one primary database and up to nine standby databases. The databases in a Data Guard configuration are connected by Oracle Net and may be dispersed geographically. There are no restrictions on where the databases are located if they can communicate with each other. For example, you can have a standby database on the same system as the primary database, along with two standby databases on another system.

The Data Guard broker logically groups these primary and standby databases into a *broker configuration* that allows the broker to manage and monitor them together as an integrated unit. You can manage them using the broker's graphical user interface (GUI) that is integrated with Oracle Enterprise Manager or using a command-line interface (CLI) called DGMGRL.

1.1.2 Oracle Data Guard Broker Overview

The Oracle Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations. The following are some of the operations that the broker automates and simplifies:

- Automated creation of Data Guard configurations incorporating a primary database, a new or existing (physical or logical) standby database, log transport services, and log apply services, where any of the databases could be Real Application Clusters (RAC) databases.
- Adding up to 8 additional new or existing (physical or logical, RAC, or non-RAC) standby databases to each existing Data Guard configuration, for a total of one primary database, and from 1 to 9 standby databases in the same configuration.
- Managing an entire Data Guard configuration, including all databases, log transport services, and log apply services, through a client connection to any database in the configuration.
- Invoking switchover or failover with a single command to initiate and control complex role changes across all databases in the configuration.

- Monitoring the status of the entire configuration, capturing diagnostic information, reporting statistics such as the log apply rate and the redo generation rate, and detecting problems quickly with centralized monitoring, testing, and performance tools.

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

These interfaces simplify the configuration and management of a Data Guard configuration. Table 2–1 provides a comparison of configuration management using the broker's interfaces and using SQL*Plus.

1.2 Benefits of Data Guard Broker

The broker's interfaces improve usability and centralize management and monitoring of the Data Guard configuration. Available as a feature of the Enterprise Edition and Personal Edition of the Oracle database, the broker is also integrated with the Oracle database. These broker attributes result in the following benefits:

Disaster protection: By automating the tasks required to configure and monitor a Data Guard configuration, the broker enhances the high availability, data protection, and disaster protection capabilities that are inherent in Oracle Data Guard. Access is possible through a client to any system in the Data Guard configuration, eliminating any single point of failure. If the primary database fails, the broker streamlines the process for any one of the standby databases to replace the primary database and take over production processing. The *intersite* availability that Data Guard provides makes it easier to protect your data.

Higher availability and scalability with Real Application Clusters (RAC) Databases: While Oracle Data Guard broker enhances disaster protection by maintaining transactionally consistent copies of the primary database, Data Guard, configured with Oracle high availability solutions such as Real Application Clusters (RAC) databases, further enhances the availability and scalability of any given copy of that database. The *intrasite* high availability of a RAC database complements the *intersite* protection that is provided by Data Guard broker.

Consider that you have a cluster system hosting a primary RAC database comprised of multiple instances sharing access to that database. Further consider that an unplanned failure has occurred. From a Data Guard broker perspective, the primary database remains available as long as at least one instance of the clustered database continues to be available for transporting redo data to the standby databases. Oracle Cluster Ready Services (CRS) manages the availability of instances of a RAC database. CRS works to rapidly recover failed instances to keep the primary database available. If CRS is unable to recover a failed instance, the Data Guard broker continues to run automatically with one less instance. If the last instance of the primary database fails, the Data Guard broker provides way to fail over to a specified standby database.

The 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). The 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 the broker to properly restart the instances as necessary, then to resume service availability on the new primary database. The broker manages the underlying Data Guard configuration and its database roles while CRS manages service availability that depends upon those roles. Applications that rely upon CRS for managing service availability will see only a temporary suspension of service as the role change occurs within the Data Guard configuration.

Note that, while CRS enhances availability of a given copy of the RAC database, the Data Guard broker enhances availability of the data across multiple geographically dispersed locations, hence providing disaster protection. Together, broker and CRS provide a strong foundation for Oracle's high-availability architecture.

> **See Also:** *Oracle Real Application Clusters Administrator's Guide* for information about CRS

Automated creation of a Data Guard configuration: The broker helps you to logically define and create a Data Guard configuration consisting of a primary database and a (physical or logical, RAC or non-RAC) standby database. The broker automatically communicates between the databases in a Data Guard configuration using Oracle Net Services. The database can be local or remote, connected by a LAN or geographically dispersed over a WAN.

The Data Guard GUI provides a wizard that automates the complex tasks involved in creating a broker configuration, including:

- Adding an existing standby database, or a new standby database created from existing backups taken through the Data Guard GUI
- Configuring the standby control file, server parameter file, and datafiles

- Initializing communication with the standby databases
- Creating online redo log files

Although the CLI cannot automatically create a new standby database, the CLI can configure and monitor an existing standby database, including those created by the Data Guard GUI.

Easy configuration of additional standby databases: After you create a Data Guard configuration consisting of a primary and standby database, you can add up to eight new or existing, physical or logical standby databases to each Data Guard configuration. The Data Guard GUI provides an Add Standby Database wizard to guide you through the process of adding more databases. The GUI also makes all Oracle Net Services configuration changes necessary to support log transport services and log apply services across the configuration.

Simplified, centralized, and extended management: You can issue commands to:

- Simplify the management of all components of the configuration, including the primary and standby databases, log transport services, and log apply services.
- Coordinate database state transitions and update database properties dynamically with the broker recording the changes in a broker configuration file that includes profiles of all the databases in the configuration. The broker propagates the changes to all databases in the configuration and their server parameter files.
- Simplify the control of the configuration protection modes (to maximize protection, to maximize availability, or to maximize performance).
- Invoke the GUI’s verify operation to ensure that log transport services and log apply services are configured and functioning properly.

Automated switchover and failover operations: Only one command is required to initiate complex role changes for switchover or failover operations across all databases in the configuration. The broker automates switchover and failover to a specified standby database in the broker configuration. The GUI enables you to select a new primary database from a set of viable standby databases (enabled and online, with normal status). The CLI `SWITCHOVER` and `FAILOVER` commands only require you to specify the target standby database before automatically initiating and completing the many steps in switchover or failover operations across the multiple databases in the configuration.

Built-in monitoring and alert and control mechanisms: The broker provides built-in validation that monitors the *health* of all of the 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 the GUI and the CLI retrieve a complete configuration view of the progress of log transport services on the primary database and the progress of log apply services on the standby database. The GUI and the CLI also retrieve data specific to physical and logical standby databases.

The ability to monitor local and remote databases and respond to events is significantly enhanced by the broker's health check mechanism and the GUI's tight integration with the Oracle Enterprise Manager event management system.

Transparent to application: Use of the broker is possible for any database because the broker works transparently with applications; no application code changes are required to accommodate a configuration that you manage with the broker.

> **See Also:** *Oracle Data Guard Concepts and Administration* for a complete description of the discrete steps that comprise the creation of standby databases and the other monitoring and control operations that have been automated or simplified by the broker.

1.3 Data Guard Broker Management Model

The broker simplifies the management of a Data Guard environment by performing operations upon the following logical objects:

- Configuration of databases
- A single database

The broker supports one or more Data Guard configurations, each of which includes a profile for one primary database and for up to nine physical or logical, RAC or non-RAC standby databases. A supported broker configuration consists of:

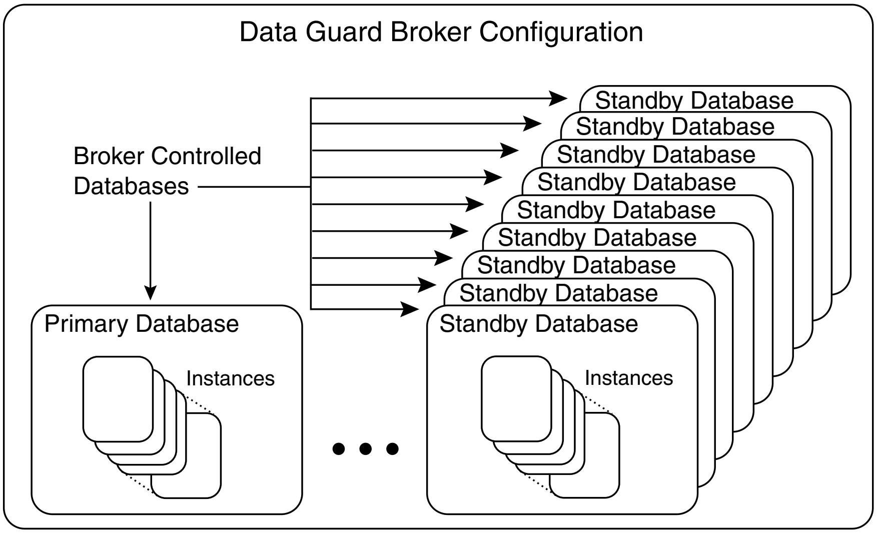
- A **configuration** object, which is a named collection of database profiles. A database **profile** is a description of a database object including its current state, current status, and properties. The configuration object profiles one primary database and up to nine standby databases that can include a mix of both physical and logical standby databases. The databases of a given configuration are typically distributed across multiple host systems.

- **Database** objects, corresponding to primary or standby databases. The broker uses a database object's profile to manage and control the state of a single database on a given system. The database object may be comprised of one or more instance objects if this is a RAC database.

  Instance objects. The broker treats a database as a collection of one or more named instances. The broker automatically discovers the instances and associates them with their database.

Figure 1–1 shows the relationship of these objects.

Figure 1–1 Relationship of Objects Managed by the Data Guard Broker



You can perform complex operations on a single database or on all databases in an entire configuration with a single mouse click or CLI command. You can enable or disable broker management of each database in a configuration one at a time. Or, if desired, enable or disable them all at the same time in a single step by enabling or disabling the configuration itself.

> **See also:** Chapter 2, Chapter 3, and Chapter 4 for more information about managing configuration and database objects

1.4 Data Guard Broker Components

The Oracle Data Guard broker consists of the following components:

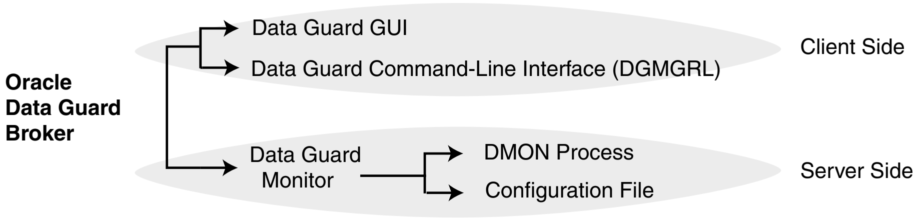
- Data Guard GUI
- Data Guard Command-Line Interface (DGMGRL)
- Data Guard Monitor

The Data Guard graphical user interface, tightly integrated with Oracle Enterprise Manager, and the Data Guard command-line interface are the broker client interfaces that help you define and manage a configuration consisting of a collection of primary and standby databases. Section 1.5 describes these interfaces in more detail.

The Data Guard monitor is the broker server-side component that is integrated with the Oracle database. Data Guard monitor is composed of the DMON process and broker configuration files that allow you to control the databases of that configuration, modify their behavior at runtime, monitor the overall health of the configuration, and provide notification of other operational characteristics. Section 1.6 describes the Data Guard monitor in more detail.

Figure 1–2 shows these components of the broker.

Figure 1–2 Oracle Data Guard Broker



1.5 Data Guard Broker User Interfaces

You can use either of the broker’s user interfaces to create a broker configuration and to control and monitor the configuration. The following sections describe the broker’s user interfaces:

- Data Guard GUI
- Data Guard Command-Line Interface (DGMGRL)

1.5.1 Data Guard GUI

The Data Guard GUI is a graphical user interface that works with the Data Guard monitor and Oracle Enterprise Manager to automate and simplify the management of a Data Guard configuration. Because it is integrated with Oracle Enterprise Manager, the Data Guard GUI enables you to manage your configuration using a familiar interface and event management system.

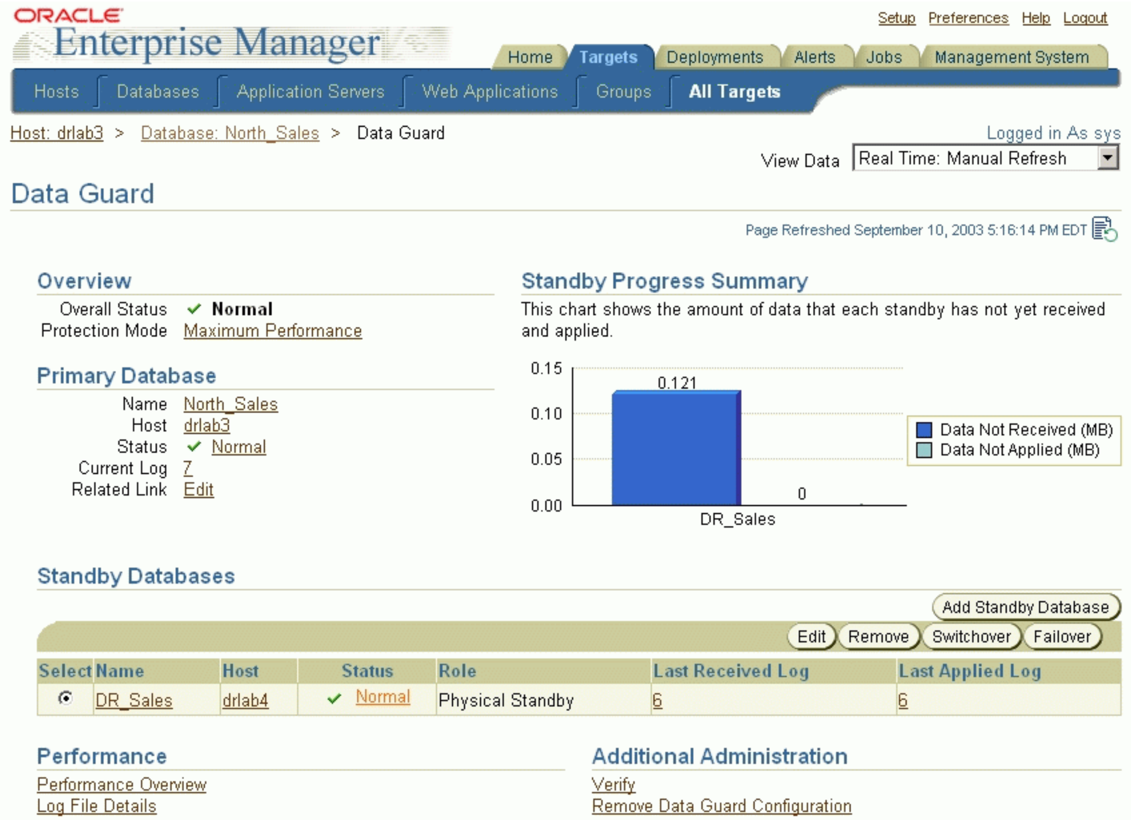
With the Data Guard GUI, the complex operations of creating and managing standby databases are simplified through wizards provided by the Data Guard GUI. The Data Guard GUI includes:

- An Add Standby Database wizard that helps you to create a broker configuration, if one does not already exist, having a primary database and a local or remote standby database. The wizard can create a physical or logical standby database (non-RAC) or import an existing physical or logical (RAC or non-RAC) standby database. If the wizard creates a physical or logical standby database, the wizard also automates the creation of the standby control file, server parameter file, online redo log files, and the standby datafiles.
- A switchover operation that helps you switch roles between the primary database and a standby database.
- A failover operation that changes one of the standby databases to the role of a primary database.
- Performance tools and graphs that help you monitor and tune log transport services and log apply services.
- Property pages that allow you to set database properties on any database and, if applicable, the settings are immediately propagated to all other databases and server parameter files in the configuration.
- Integration with Oracle Enterprise Manager to perform event reporting through e-mail.

In addition, it makes all Oracle Net Services configuration changes necessary to support log transport services and log apply services.

Figure 1–3 shows the overview page from the Data Guard GUI in Oracle Enterprise Manager.

Figure 1–3 Data Guard GUI (in Oracle Enterprise Manager) - Overview Page


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Host: drlab3 > Database: North_Sales > Data Guard
Logged in As sys
0.15
0.10
0.05
0.00
0.121
0
DR_Sales
🟦 Data Not Received (MB)
🟩 Data Not Applied (MB)
Add Standby Database
Edit Remove Switchover Failover

View Data: Real Time: Manual Refresh

Data Guard

Page Refreshed September 10, 2003 5:16:14 PM EDT

Overview

Overall Status	✓ Normal
Protection Mode	Maximum Performance

Primary Database

Name	North_Sales
Host	drlab3
Status	✓ Normal
Current Log	7
Related Link	Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	DR_Sales	drlab4	✓ Normal	Physical Standby	6	6

Performance

Performance Overview
Log File Details

Additional Administration

Verify
Remove Data Guard Configuration

> **See Also:** The Data Guard GUI online help

1.5.2 Data Guard Command-Line Interface (DGMGRL)

The Data Guard command-line interface (CLI) enables you to control and monitor a Data Guard configuration from the CLI prompt or within scripts. You can perform most of the activities required to manage and monitor the databases in the configuration using the CLI.

The following example lists the available commands:

```
DGMGRL> HELP

The following commands are available:
```

```
add            Add a standby database into the broker configuration
connect        Connect to an Oracle instance
create         Create a broker configuration
disable        Disable broker control of a configuration or database
edit           Edit a configuration, database or instance
enable         Enable broker control of a configuration or database
exit           Exit the program
failover       Change a standby database to be the primary database
help           Display description and syntax for a given command
quit           Exit the program
rem            Comment to be ignored by DGMGRL
remove         Remove a configuration, database or instance
show           Display information of a configuration, database or instance
shutdown       Shut down a currently running Oracle instance
startup        Start an Oracle database instance
switchover     Switch roles between the primary database and a standby database

Use "help <command>" to see syntax for individual commands.
```

See Also: Chapter 7 for complete reference information for the Data Guard command-line interface

1.6 Data Guard Monitor

The configuration, control, and monitoring functions of the broker are implemented by server-side software and configuration files that are maintained for each database that the broker manages. The software is called the Data Guard monitor.

The following sections describe how the Data Guard monitor interacts with the Oracle database and with remote Data Guard monitors to manage the broker configuration.

1.6.1 Data Guard Monitor (DMON) Process

The Data Guard monitor process (DMON) is an Oracle background process that runs for every database instance that is managed by the broker. When you start the Data Guard broker, a DMON process is created.

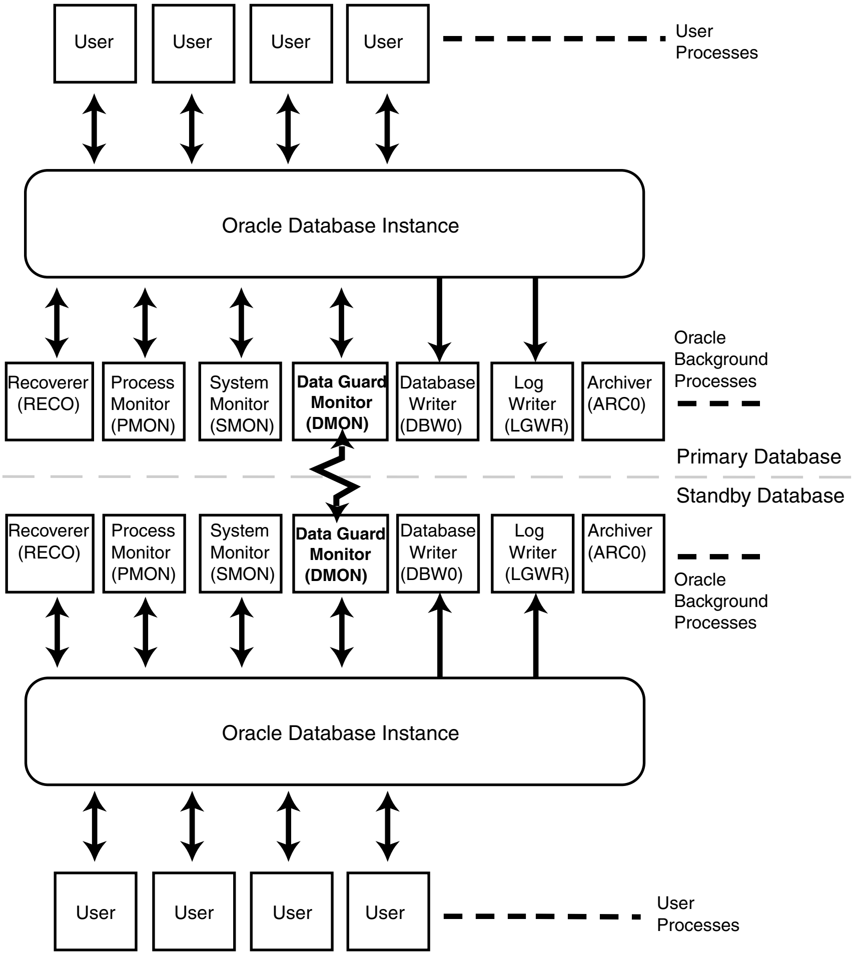
See Also: Section 2.3 for information on starting the broker.

When you use the Data Guard GUI or the CLI to manage a database, the DMON process is the server-side component that interacts with the local database and the DMON processes of the other databases to perform the requested function. The DMON process is also responsible for monitoring the health of the broker configuration and for ensuring that every database has a consistent description of the configuration.

> **See Also:** *Oracle Database Concepts* for more information about the memory structures and processes that are used with an Oracle database

Figure 1–4 shows the broker's DMON process as one of several background processes that constitute an instance of the Oracle database. Figure 1–4 shows multiple databases, each having its own DMON process. This distributes the broker across all of the databases of the broker configuration.

Figure 1–4 Databases With Distributed Broker (DMON) Processes



The zigzag arrow in the center of Figure 1–4 represents the two-way Oracle Net Services communication channel that exists between the DMON processes of two databases in the same broker configuration.

This two-way communication channel is used to pass requests between databases and to monitor the health of all of the databases in the broker configuration.

When creating a new Data Guard configuration or adding a new standby database into an existing configuration, the DMON process uses an initial connect identifier to connect to the database to collect necessary information about the database. This initial connect identifier is supplied by the user if the CLI is used, or constructed automatically if the GUI is used.

After the initial connection, the DMON process constructs connect descriptors for communication with other DMON processes on other databases, using the address value from the `LOCAL_LISTENER` initialization parameter from those databases. The DMON processes automatically manage the connections to each other. If a database is a RAC database, then as long as one instance in the database is running and the DMON process is started on the instance, that DMON process is able to establish two-way communications with other DMON processes on other databases to manage the database as part of the Data Guard configuration.

1.6.2 Configuration Management

The broker's DMON process persistently maintains profiles about all database objects in the broker configuration in a binary **configuration file.** A copy of this file is maintained by the DMON process for each of the databases that belong to the broker configuration. Each database's copy of the file is shared by all instances of the database if it is a RAC database. Changes to this file are made by the DMON process for all copies.

This configuration file contains profiles that describe the states and properties of the databases in the configuration. For example, the file records the databases that are part of the configuration, the roles and properties of each of the databases, and the state of each database in the configuration.

The configuration data is managed transparently by the DMON process to ensure that the configuration information is kept consistent across all of the databases. The broker uses the data in the configuration file to configure and start the databases, control each database's behavior, and provide information to the CLI and the Data Guard GUI. (See Section 3.3 for more information.)

Whenever you add databases to a broker configuration, or make a change to an existing database's properties, each DMON process records the new information in its copy of the configuration file.

1.6.3 Database Property Management

Associated with each database are various properties that the DMON process uses to control the database's behavior. The properties are recorded in the configuration file as a part of the database's object profile that is stored there. Many database properties are used to control database initialization parameters related to the Data Guard environment.

To ensure that the broker can update the values of parameters in both the database itself and in the configuration file, you must use a server parameter file to control static and dynamic initialization parameters. The use of a server parameter file gives the broker a mechanism that allows it to reconcile property values selected by the database administrator (DBA) when using the broker with any related initialization parameter values recorded in the server parameter file.

When you set values for database properties in the broker configuration, the broker records the change in the configuration file and propagates the change to all of the databases in the Data Guard configuration.

Note: The broker supports both the default and nondefault server parameter file filenames. If you use a nondefault server parameter filename, the initialization parameter file must include the complete filename and location of the server parameter file. If this is a RAC database, there must be one nondefault server parameter file for all instances.

See Also: Section 3.3.2 for more information.

1.7 Oracle Data Guard Installation, Upgrade, Downgrade, and First Use

Oracle Data Guard, the Data Guard monitor, and the Data Guard command-line interface (DGMGRL) are included with the Enterprise Edition or Personal Edition of the Oracle database software.

The Data Guard graphical user interface (GUI) is included with the Oracle Enterprise Manager software.

Note: In Oracle9*i*, the Data Guard GUI was called Data Guard Manager and was usable either through the Oracle Enterprise Manager Console or by invoking it directly from the command line. In Oracle Database 10*g*, it is tightly integrated with Oracle Enterprise Manager and is referred to as the Data Guard GUI.

1.7.1 Installation

To use the Data Guard monitor and the CLI, you must install the Oracle Enterprise Edition or Personal Edition database on each location where you plan to manage broker configurations.

The Oracle Data Guard graphical user interface (GUI), which you can use to manage broker configurations, is installed with the Oracle Enterprise Manager software.

1.7.2 Upgrade from Release 9.0.*n* to Release 10.1

If you are currently running an Oracle Data Guard release 9.0.*n* configuration, you must upgrade to Oracle Database release 10.1 and re-create the broker configuration, as follows:

1. Delete (remove) the release 9.0.*n* broker configuration using the same release of either Data Guard Manager or the CLI. For example, the CLI `REMOVE CONFIGURATION` command can be used.
2. Upgrade the database software to Oracle Database release 10.1. See the Oracle installation documentation that is appropriate for your operating system.
3. If you are using Oracle Enterprise Manager and Data Guard Manager release 9.0.*n*, you must upgrade to Oracle Enterprise Manager release 10.1 to manage a broker configuration running Oracle Data Guard release 10.1:
   - Data Guard Manager release 9.0.*n* is not compatible with Oracle Data Guard release 10.1.
   - Data Guard GUI release 10.1 is not compatible with Oracle Data Guard release 9.0.*n*. You will receive an error message stating that the Oracle database is too old.
4. If you are using the CLI release 9.0.*n*, you must upgrade to Data Guard command-line interface release 10.1:
   - The CLI release 9.0.*n* is not compatible with Oracle Data Guard release 10.1.

     Note: Existing Oracle9*i* command-line scripts are supported in Oracle Database 10*g* for non-RAC databases. See Appendix A for information about deprecated commands.

   - The CLI release 10.1 is not compatible with Oracle Data Guard release 9.0.*n*.

Note: Oracle Database 10*g* command-line scripts are not supported in Oracle9*i*.

5. Invoke the Data Guard GUI or the CLI, and re-create the broker configuration.

   See Also: See *Oracle Database Upgrade Guide* if you are upgrading from Oracle8*i* Data Guard to Oracle Data Guard

1.7.3 Upgrade from Release 9.2.0 to Release 10.1

If you are currently running an Oracle Data Guard release 9.2.0 configuration, you must upgrade to Oracle release 10.1, and re-create the broker configuration, as follows:

1. Delete (remove) the release 9.2.0 broker configuration using Data Guard Manager or the CLI release 9.2.0. For example, the CLI `REMOVE CONFIGURATION` command can be used.
2. If using the CLI, clear the `LOG_ARCHIVE_DEST_`*n* initialization parameter settings by using the `ALTER SYSTEM SET LOG_ARCHIVE_DEST_`*n*`=" "` SQL*Plus command.
3. Upgrade the database software to Oracle release 10.1. See the Oracle Database installation documentation that is appropriate for your operating system.
4. If you are using Oracle Enterprise Manager and Data Guard Manager release 9.2.0, you must upgrade to Oracle Enterprise Manager release 10.1 to manage a broker configuration running Oracle Data Guard release 10.1. Data Guard Manager release 9.2.0 is not compatible with Oracle Data Guard release 10.1.
5. If you are using the CLI release 9.2.0, you must upgrade to Data Guard command-line interface release 10.1:
   - The CLI release 9.2.0 is not compatible with Oracle Data Guard release 10.1.

     Note: Existing Oracle9*i* command-line scripts are supported in Oracle Database 10*g* for non-RAC databases. See Appendix A for information about deprecated commands.

   - The CLI release 10.1 is not compatible with Oracle Data Guard release 9.2.0.

Note: Oracle Database 10*g* command-line scripts are not supported in Oracle9*i*.

---

6. Invoke the Data Guard GUI or the CLI, and re-create the broker configuration.

1.7.4 Downgrade from Release 10.1

If you have upgraded to release 10.1 and want to downgrade to your prior release, you must downgrade the database release and re-create the broker configuration as follows:

1. Delete (remove) the release 10.1 broker configuration using the Data Guard GUI or the CLI release 10.1. For example, the CLI `REMOVE CONFIGURATION` command can be used.
2. Downgrade the database software to your prior Oracle release. See the Oracle database documentation that is appropriate for your operating system.
3. If you are downgrading to Oracle release 9.2.0 and you were using the Data Guard GUI, you may continue to use the Data Guard GUI in Oracle Enterprise Manager release 10.1 to manage your 9.2.0 broker configuration. You may also downgrade to the Data Guard Manager by re-installing Oracle Enterprise Manager release 9.2.0.
4. If you are downgrading to Oracle release 9.0.*n* and you were using the Data Guard GUI, you must downgrade to the Data Guard Manager by re-installing Oracle Enterprise Manager release 9.0.*n*. The Oracle Database 10*g* Data Guard GUI is not compatible with Oracle Data Guard release 9.0.*n*.
5. Invoke the Data Guard Manager or the CLI and re-create the broker configuration.

1.7.5 Prerequisites for First Use

The following conditions must be true before you can use the broker:

- The primary and standby databases must be running Oracle 10.1 and each can be installed in either a single-instance or multi-instance environment. The database must be licensed for Oracle Enterprise Edition or Personal Edition.
- You must use a server parameter file (SPFILE) to ensure the broker can persistently reconcile values between broker properties and any related initialization parameter values. See Section 3.3.2 for more information.

- The value of the DG_BROKER_START parameter must be set to TRUE. See Section 2.3 for more information. (Data Guard GUI sets this parameter automatically.)
- If any of the databases in the configuration is a RAC database, you must set up the DG_BROKER_CONFIG_FILE*n* initialization parameters for that database such that they point to the same shared files for all instances of that database. The default values for these parameters will not work. The shared files could be files on a cluster file system, if available, or on raw devices.

  See Also: See Section 1.6.2 for information about the configuration file. See Section 2.2 for details about setting up the broker configuration file. See Section 2.2.1 for details about sizing the raw devices.

- Oracle Net Services network files must be set up on the primary database and on the standby database if you configure an existing standby database into the broker configuration. Otherwise, the Data Guard GUI automatically sets up the network files when it creates a standby database.
- The LOCAL_LISTENER initialization parameter on each instance that is part of a Data Guard broker configuration must resolve to a listener address that is reachable by all members of the configuration. See the *Oracle Database Reference* and the *Oracle Net Services Administrator's Guide* for additional information.
- To enable the Data Guard broker's CLI to restart instances during the course of broker operations, a service with a specific name must be statically registered with the local listener of each instance. The value for the GLOBAL_DBNAME attribute must be set to a concatenation of *db_unique_name*_DGMGRL.*db_domain*. For example, in the listener.ora file:

```
LISTENER = (DESCRIPTION =
     (ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=host_name)
     (PORT=port_num))))
SID_LIST_LISTENER=(SID_LIST=(SID_DESC=(SID_NAME=sid_name)
     (GLOBAL_DBNAME=db_unique_name_DGMGRL.db_domain)
     (ORACLE_HOME=oracle_home)))
```

- Any database, including instances of the database, managed by the broker must be mounted.
- If any of the databases in the configuration is a RAC database, the START_OPTIONS for that database must be set to MOUNT in the Oracle Cluster Repository (OCR) using SRVCTL as follows:

```
SRVCTL ADD DATABASE -d <db_unique_name> -o <$ORACLE_HOME> -s MOUNT
or
SRVCTL MODIFY DATABASE -d <db_unique_name> -o <$ORACLE_HOME> -s MOUNT
```

- The primary database must be opened in ARCHIVELOG mode.
- You must set the COMPATIBLE initialization parameter to 9.2.0 or higher for both the primary and standby databases.

> **See Also:** Section 2.3 for more information about preparing and starting the Oracle Data Guard broker. See *Oracle Data Guard Concepts and Administration* for more information about setting up the network files.

2

Managing Broker Configurations

This chapter contains the following sections:

2.1 Configuration Support

The broker enables you to logically define a Data Guard configuration, consisting of a primary database and physical and logical standby databases. With the broker, you define a **broker configuration** that is a *logical* grouping of the databases, including log transport services and log apply services. At the DBA's discretion, the broker controls the logical objects in the configuration, modifies their behavior at runtime, monitors the overall health of the configuration, and reports any health and other operational characteristics up through the Oracle Enterprise Management notification mechanisms if you are using the Data Guard GUI, or through `SHOW` commands if you are using the CLI.

The broker supports Data Guard configurations consisting of a primary database, and up to nine standby databases that are either local to, or, remote from, the primary database, and any of those databases can be an Oracle Real Application Clusters (RAC) database.

A supported Data Guard configuration contains the following components:

- A (RAC or non-RAC) primary database
- From one to nine physical or logical (RAC or non-RAC) standby databases
- Physical systems that host the primary and standby databases
- Oracle Net Services network configuration that defines a connection between the databases
- Standby (archived redo log files) destination parameters and configuration properties
- Log transport services that transmit the redo data from the primary database to the standby databases
- Log apply services that apply the archived redo log files or standby redo log files to the standby databases as they arrive from the primary database

The standby database is updated with redo data that is transmitted automatically from the primary database by log transport services. The archived redo log file and standby redo log file contain all of the database changes except for *unrecoverable* or *unlogged* changes. On the standby database, log apply services apply the archived redo log files to stay synchronized with the primary database. Thus, the standby database can take over operations if the primary database becomes unusable.

The broker's DMON process configures and maintains the broker configuration as a group of objects that you can manage and monitor as a single unit. Thus, when you enter a command that affects multiple databases, the DMON process:

- Carries out your request on the primary database
- Coordinates with the DMON process for each of the other databases, as required for your request
- Updates its local configuration file
- Communicates with the DMON process for each of the other databases to update their copies of the configuration file

The DMON process enables you to configure, monitor, and control the databases and the configuration together as a unit. If the configuration is disabled, broker management of all of the databases in the configuration is also disabled. If you later request the configuration to be enabled, broker management is enabled for each database in the configuration.

Figure 2–1 shows a two-database broker configuration with the Data Guard monitor (DMON) process running at each location.

Figure 2–1 Oracle Data Guard Broker Configuration

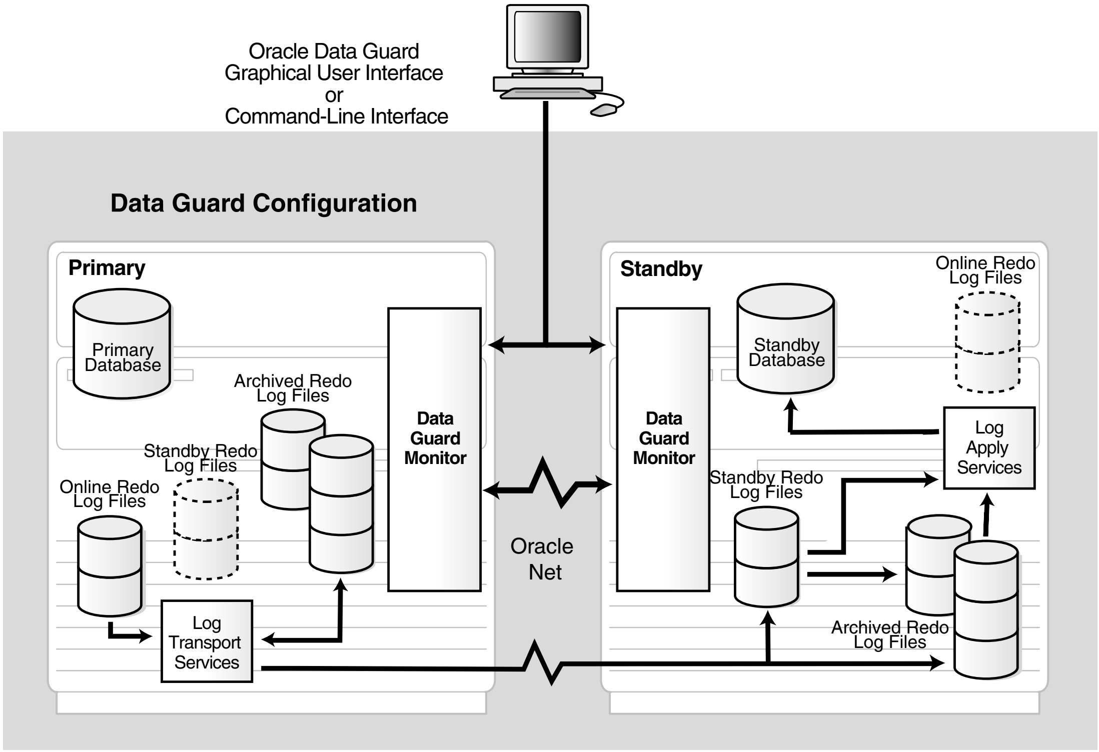


Table 2–1 provides a comparison of configuration management using the broker's interfaces and using SQL*Plus.

Table 2–1 Configuration Management With and Without the Broker

	With the Broker	Without the Broker
General	Provides primary and standby database management as one unified configuration.	You must manage the primary and standby databases separately.
Standby Database Creation	Provides the Data Guard GUI wizards that automate and simplify the steps required to create a configuration with an Oracle database on each site, including creating the standby control file, online redo log files, datafiles, and server parameter files.	You must manually: ■ Copy the database files to the standby database. ■ Create a control file on the standby database. ■ Create server parameter or initialization parameter files on the standby database.
Configuration and Management	Enables you to configure and manage multiple databases from a single location and automatically unifies all of the databases in the broker configuration.	You must manually: ■ Set up log transport services and log apply services on each database in the configuration. ■ Manage the primary database and standby databases individually.
Control	■ Automatically set up log transport services and log apply services. Simplify management of these services, especially within a RAC database environment. ■ Automates switchover and failover. ■ Automates CRS service and instance management over database role transitions. ■ Provides mouse-driven database state changes and a unified presentation of configuration and database status. ■ Provides mouse-driven property changes.	You must manually: ■ Use many separate SQL*Plus statements to manage the database. ■ Coordinate sequences of multiple commands across multiple database sites to execute operations. ■ Coordinate sequences of multiple commands to disable CRS management of services and instances, to execute role transitions, then to reenable CRS management of instances and services that are to be available after the role transitions.
Monitoring	■ Provides continuous monitoring of the configuration health, database health, and other runtime parameters. ■ Provides a unified updated status and detailed reports. ■ Provides an integrated tie-in to Oracle Enterprise Manager events.	You must manually: ■ Monitor the status and runtime parameters using fixed views on each database—there is no unified view of status for all of the databases in the configuration. ■ Provide a custom method for monitoring Oracle Enterprise Manager events.

2.2 Setting Up the Broker Configuration Files

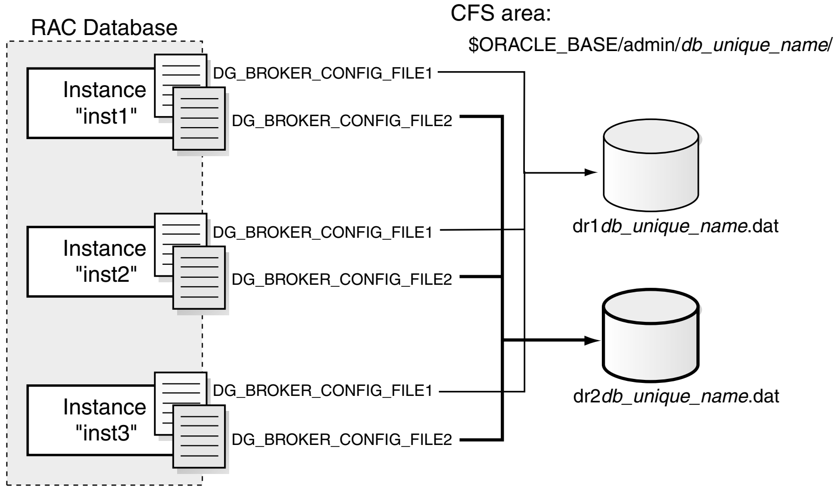
Two copies of the configuration file are maintained for each database so as to always have a record of the last known *valid* state of the configuration. When the broker is started for the first time, the configuration files are automatically created and named using a default path name and filename that is operating-system specific. You can override this default path name and filename by setting the following initialization parameters for that database:

```
DG_BROKER_CONFIG_FILE1
DG_BROKER_CONFIG_FILE2
```

If the broker is managing a RAC database, the value of `DG_BROKER_CONFIG_FILE1` and the value of `DG_BROKER_CONFIG_FILE2` for each of the instances must point to the same set of physical files. In other words, all instances of the database must reference the same set of configuration files. If cluster file system (CFS) is available, and the configuration files reside there, the `DG_BROKER_CONFIG_FILE`*n* parameters on all of the instances must be set to these files including the path to the CFS area. Figure 2–2 shows the set up for the broker configuration files on CFS. In this scenario, the parameters and value for all instances would be:

```
DG_BROKER_CONFIG_FILE1=$ORACLE_BASE/admin/db_unique_name/dr1db_unique_name.dat
DG_BROKER_CONFIG_FILE2=$ORACLE_BASE/admin/db_unique_name/dr2db_unique_name.dat
```

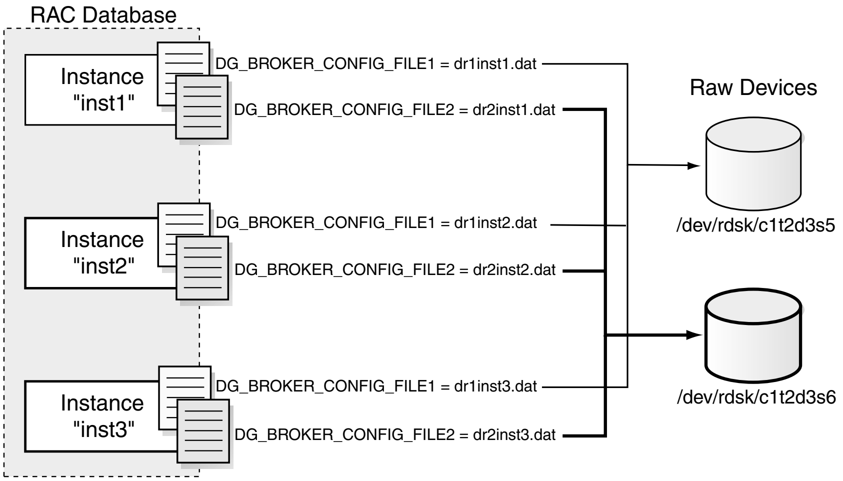
Figure 2–2 Broker Configuration Setup in a CFS Area



If CFS is not available, the files must be on raw devices. In this case, the parameter values on each of the instances must point to the raw devices. Figure 2–3 shows the set up for the broker configuration files on raw devices. On a UNIX system, you would set this up similar to the following:

```
%ln -s /dev/rdsk/c1t2d3s5 dr1instn.dat
```

Figure 2–3 Broker Configuration Setup With Raw Device


RAC Database
Instance "inst1"
DG_BROKER_CONFIG_FILE1 = dr1inst1.dat
DG_BROKER_CONFIG_FILE2 = dr2inst1.dat
Instance "inst2"
DG_BROKER_CONFIG_FILE1 = dr1inst2.dat
DG_BROKER_CONFIG_FILE2 = dr2inst2.dat
Instance "inst3"
DG_BROKER_CONFIG_FILE1 = dr1inst3.dat
DG_BROKER_CONFIG_FILE2 = dr2inst3.dat
Raw Devices
/dev/rdsk/c1t2d3s5
/dev/rdsk/c1t2d3s6

You can change the configuration filenames dynamically by issuing the ALTER SYSTEM SQL statement. However, you cannot alter these parameters when the DMON process is running. To change the names of these configuration files for a given database, perform the following steps:

1. Disable the broker configuration using the CLI DISABLE command. See Section 2.5.

2. Stop the Data Guard broker DMON process using the following SQL statement:

   ```
   SQL> ALTER SYSTEM SET DG_BROKER_START=FALSE;
   ```

3. Change the configuration filenames for the database:

   ```
   SQL> ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1=filespec1
   SQL> ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2=filespec2
   ```

Note: If the broker is managing a RAC database, the value of DG_BROKER_CONFIG_FILE1 and the value of DG_BROKER_CONFIG_FILE2 for each of the instances must point to the same set of physical files.

---

4. If not on raw devices, rename the existing files to *filespec1* and *filespec2*, respectively, at the operating system level to avoid losing the existing broker configuration information.
5. Restart the Data Guard broker DMON process, as follows:

   ```
   SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE;
   ```

6. Enable the broker configuration using the CLI ENABLE command or the Enable operation in the Data Guard GUI.

2.2.1 Sizing for Raw Devices

If the broker configuration files need to be on raw devices, set up two additional raw devices of 1MB each. Set up the value of the DG_BROKER_CONFIG_FILE1 and DG_BROKER_CONFIG_FILE2 parameters to point to the raw devices. A 1MB configuration file will accommodate 10 databases with a total of 45 instances between them.

You may need a larger device if the number of instances for this configuration exceeds 45 instances. You will need 15KB for each additional instance.

2.3 Starting the Data Guard Broker

After setting up the configuration files, the DG_BROKER_START initialization parameter must be set to TRUE for each database to start the Data Guard monitor (DMON) processes.

By default, the DG_BROKER_START initialization parameter is set to FALSE. However, you can set the value in the following ways:

- If you are using the Data Guard GUI, it automatically sets the DG_BROKER_START initialization parameter to TRUE for new standby databases that it creates.
- If you are using the CLI, you must explicitly set the DG_BROKER_START initialization parameter to TRUE; otherwise, the DMON process will not start.

You can set the DG_BROKER_START initialization parameter with the following SQL statement:

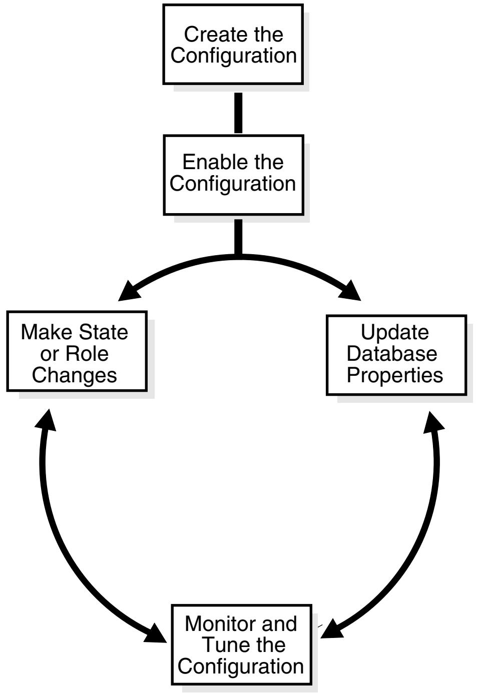
```
SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE;

System altered.

SQL> SHOW PARAMETER DG_BROKER_START

NAME               TYPE      VALUE
------------------------------------
dg_broker_start    boolean   TRUE
```

Whether you use the Data Guard GUI or the CLI, set the DG_BROKER_START=TRUE initialization parameter in the server parameter file on each primary and standby database. Doing so ensures that the DMON processes will start automatically the next time you start any instance of the database.

2.4 Management Cycle of a Broker Configuration

The broker helps you to create a new configuration or manage an existing configuration. Figure 2–4 shows the life cycle of a broker configuration.

Figure 2–4 Life Cycle of a Broker Configuration and Its Databases


Create the Configuration
Enable the Configuration
Make State or Role Changes
Update Database Properties
Monitor and Tune the Configuration

Create the Broker Configuration

When using the Data Guard GUI, the Add Standby Database wizard can either add an existing (RAC or non-RAC) standby database into the configuration or create a new (non-RAC only) standby database and add it to the configuration. The standby database can be either a physical or logical database.

When using the CLI, the primary database and a standby database must already exist. You construct the standby database from backups of the primary database control files and datafiles, and then prepare it for recovery.

> **See Also:** Chapter 5 and Chapter 6 which describe the preparation requirements if you are using the Data Guard GUI or the CLI, respectively

Enable the Broker Configuration

A Data Guard configuration must be enabled to be managed or monitored by the broker. Conversely, you disable a configuration if you no longer want to manage it with the broker. When you disable a configuration, broker management of all of its databases is also disabled.

> **Note:** You can enable or disable the configuration using the CLI. You cannot disable the configuration using the Data Guard GUI. You can enable the configuration using the Data Guard GUI in the event that it was previously disabled using the CLI.

A broker configuration, when first created using the Data Guard GUI, is automatically enabled as soon as the Add Standby Database wizard completes.

A broker configuration, when first created using the CLI, is in a disabled condition. This means its constituent databases are not yet under active control of the Data Guard monitor. When you finish configuring the databases into a broker configuration with the CLI, you must enable the configuration to allow the Data Guard monitor (DMON) process to manage the configuration.

You can enable:

- The entire configuration, including all of its databases
- An individual standby database

You can easily disable a database if a problem occurs and you can no longer function properly in the broker configuration.

You may also want to disable a configuration temporarily, and then change some properties in the broker configuration without affecting the actual database properties. The changed properties will take effect when the configuration is enabled again for management by the broker.

Make Role Changes Within the Broker Configuration, As Needed

At any time, you can issue a single command to change the roles of the databases in the configuration. If some event renders the primary database unusable, you can fail over one of the standby databases to become the new primary database.

In addition, planned downtime for maintenance can be reduced because you can quickly switch over production processing from the current primary database to a standby database, and then switch back again after the planned maintenance.

See Also: Chapter 4 for more information about role changes

Make State Changes to the Databases, As Needed

The Data Guard broker transitions the databases into an online state, by default, the first time that you enable the database.

At any time, you can issue a single command through the Data Guard GUI or the CLI to change the state of the database. For example, you could bring the primary database into a `LOG-TRANSPORT-OFF` state to temporarily stop archiving log files to the standby database. Then, you issue another command to return the database to a full online state (that is, online and archiving log files to the standby databases).

See Also: Chapter 3 for more information about database state changes

Update Database Properties, As Needed

The Data Guard broker enables you to set database properties, some of which correspond to database initialization parameters. You can change these properties to dynamically control such things as log transport, file management, log apply, and to support the overall configuration protection mode. The broker records the changes in the broker configuration file for each database in the Data Guard configuration and propagates the changes to the related initialization parameters in the server parameter files, if needed.

See Also: Chapter 3 and Chapter 8 for complete information about database properties

Set Data Protection Levels, As Needed

The Data Guard broker enables you to set the data protection level for the configuration. You can configure the protection mode to maximize data protection, maximize availability, or maximize performance.

See Also: Section 3.6 for information about managing data protection modes

Monitor the Configuration

You can check the health of the configuration, display and update the properties of the databases, and set Oracle Enterprise Manager events.

The Data Guard GUI also provides a dynamic performance page that automatically and dynamically refreshes chart data and status at specified intervals. The performance chart shows a graphical summary of how far behind and how much redo data is being generated and applied.

See Also: Chapter 5 and Chapter 6 for scenarios that show examples using the Data Guard GUI and the CLI, respectively

2.5 Enable and Disable Operations

A key concept of management with the broker is the notion of enabling and disabling broker management of the databases in a broker configuration. The enable and disable operations are defined for databases that were incorporated into a broker configuration; you cannot perform these broker operations on the physical components of a Data Guard configuration, nor on databases that are not part of the broker configuration. This is because when you enable or disable a database in the broker configuration, you are effectively enabling or disabling the ability of the Data Guard monitor (DMON) process to:

- Manage and monitor the specified database.
- Manage the profile information in the broker configuration file for each database.

However, disabling a broker configuration does not affect current services and operations in the actual Data Guard configuration. For example, when you disable a broker configuration, log transport services and log apply services in the Data Guard configuration continue to function unchanged, but you can no longer manage them through the broker interfaces.

In addition, disabling a database *does not* remove or delete its profile from the broker configuration file. You can reenable your ability to manage with the broker using the CLI `ENABLE CONFIGURATION` or `ENABLE DATABASE` commands, or the `Enable` option in the Data Guard GUI.

Note: You can enable or disable the configuration using the CLI. You cannot disable the configuration using the Data Guard GUI. You can enable the configuration using the Data Guard GIUI in the event that it was previously disabled using the CLI.

Caution: If you disable broker management of a standby database in the broker configuration, that standby database cannot be used by the broker as a failover target in the event of loss of the primary database.

Disabling broker management of the configuration may be useful to do even though you are removing the broker's ability to monitor and control the databases. For example, it may be advantageous to disable a configuration temporarily in order to change one or more properties in the broker configuration all at the same time. When you change properties in a disabled configuration, it does not affect the actual database properties underneath because the changes are not applied to the running database until you reenable the configuration. For example, you might want to change the overall configuration protection mode and the log transport services properties on a disabled configuration so that all changes are applied to the configuration at the same time upon the next enable operation.

Note: For Oracle9*i* users, the configuration object is in either an online or offline state. For Oracle Database 10*g* users, the configuration object is always in an online state.

See Also: Section 3.6.2, "How Broker Operations Affect Protection Modes"

2.6 Configuration Status

A configuration status reveals the overall health of the configuration. Status of the configuration is acquired from the status of all of its databases.

The following list describes the possible status modes for a configuration:

- Normal

  The configuration, including all of the databases configured in it, is operating as specified by the user without any warnings or errors.

- Warning

  One or more of the databases in the configuration is not operating as specified by the user. To obtain more information, use the CLI `SHOW` command or the

Data Guard GUI display to locate each database and examine its error status to reveal the source of the problem.

- Error

  One or more of the databases in the configuration failed or may no longer be operating as specified by the user. To obtain more information, use the CLI `SHOW` command or the Data Guard GUI display to locate each database and examine its error status to reveal the source of the problem.

- Unknown/Disabled

  Broker management of the configuration is disabled. The Data Guard monitor is not monitoring the status of the databases in the configuration.

3

Managing Databases

This chapter describes managing the states and properties that are specific to the database. This chapter contains the following sections:

3.1 Database Objects

The broker manages database objects. A database object profiles and corresponds to a primary or standby database. The broker uses this object's profile to manage and monitor the state of a single database.

The broker distinguishes between a physical and a logical standby database. These databases are configured with a profile having states and properties that are appropriate for their standby types.

3.2 Database States

When a configuration is enabled, its databases can be in one of several states. Table 3–1 describes all of the primary and standby database states.

Table 3–1 Database States and Descriptions

Database Role	State Name	Description
Primary	`ONLINE`	The primary database is open for read/write access and log transport services are archiving online redo log files to the standby databases. If this is a RAC database, all started instances are open in read/write mode and have log transport services running. This is the default state for a primary database when it is enabled for the first time.
Primary	`LOG-TRANSPORT-OFF`	The primary database is open for read/write access, but log transport services are not transmitting redo data to the standby databases. If this is a RAC database, all started instances are open in read/write mode and log transport services are not running on any instances.
Physical standby	`ONLINE`	The physical standby database is mounted and log apply services are started. The standby database is not open for read-only queries. If the standby database is a RAC database, the broker starts log apply services on exactly one standby instance, called the **apply instance**. If this instance fails, the broker automatically chooses another started instance. This new instance then becomes the apply instance. This is the default state for a physical standby database when it is enabled for the first time.
Physical standby	`LOG-APPLY-OFF`	The physical standby database is mounted, but log apply services are stopped. The standby database is not open for read-only queries. If this is a RAC database, there is no instance running log apply services until you change the database state to `ONLINE`.

Table 3–1 (Cont.) Database States and Descriptions

Database Role	State Name	Description
Physical standby	`READ-ONLY`	The physical standby database is open for read-only queries, and log apply services are stopped. If this is a RAC database, one or more instances will be open in read-only mode. Log apply services are not running on any instance.
Logical standby	`ONLINE`	The logical standby database is open for read-only queries and log apply services are started. The logical standby database guard is on. If this is a RAC database, log apply services are running on one instance, the apply instance. If this instance fails, the broker automatically chooses another started instance. This new instance becomes the apply instance. This is the default state for a logical standby database when it is enabled for the first time.
Logical standby	`LOG-APPLY-OFF`	The logical standby database is open for read-only queries, and log apply services are not running. The logical standby database guard is on. If this is a RAC database, there is no instance running log apply services until you change the database state to `ONLINE`.

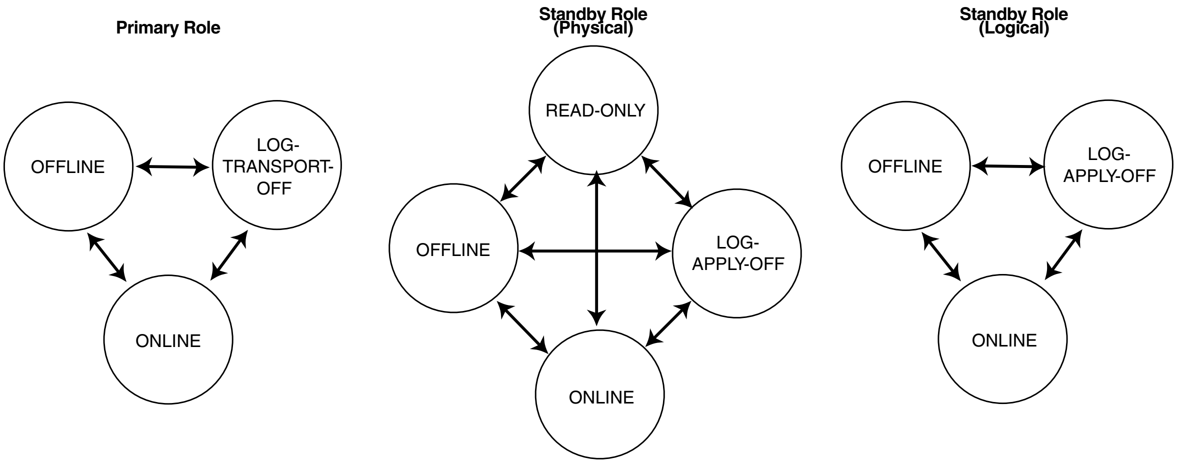
Table 3–1 (Cont.) Database States and Descriptions

Database Role	State Name	Description
All	OFFLINE	When you set the state of a standby database to OFFLINE, the broker automatically shuts down the database, turns off log transport services to this database, and leaves the database as disabled in the broker configuration. The broker will not manage this database until you restart the database. When you set the state of a primary database to OFFLINE, the broker automatically shuts down the database. The primary database is not available, and the broker is no longer managing the entire Data Guard configuration. You need to restart the database to resume broker control. **Note:** In order for the broker to once again manage a standby database that is shutdown, the primary database must be running and the broker must be managing the Data Guard configuration when the standby database is restarted. If this is a RAC database, all started instances are shutdown. **Caution:** Before setting the state to OFFLINE, you should carefully consider whether or not the interruption in access to data and computing resources is necessary.

3.2.1 Database State Transitions

Figure 3–1 graphically shows the transition of the states that were described in Table 3–1. The double arrows indicate that you can transition from one state to any other state.

Figure 3–1 Database State Transition Diagrams



With the CLI, you can use the EDIT DATABASE command to explicitly change the state of a database. For example, the EDIT DATABASE command in the following example changes the state of the North_Sales database to LOG-TRANSPORT-OFF.

```
DGMGRL> EDIT DATABASE 'North_Sales' SET STATE='LOG-TRANSPORT-OFF';
Succeeded.
```

> **See Also:** Chapter 7 for complete information about the EDIT DATABASE command. See Chapter 5 for examples of performing state transitions using the Data Guard GUI.

The following sections detail the transition of the states in which the primary and standby databases can be.

Primary database state transitions

For the primary database, when transitioning from any state to the ONLINE state, the broker sets up log transport services to all broker-managed standby databases using the log transport-related properties of the standby databases (see Section 3.4 for the list of all log transport-related properties). Log transport services setup is done by setting the LOG_ARCHIVE_DEST_*n* and LOG_ARCHIVE_DEST_STATE_*n* initialization parameters on the primary database, and the LOG_ARCHIVE_CONFIG initialization parameter on all databases (primary or standby). If necessary, the broker also sets up the data protection mode of the database to match the protection mode recorded in the broker configuration file, and opens the database for read and write access. Finally, the broker switches a log for each thread to initiate log transport services.

When transitioning from any state to the LOG-TRANSPORT-OFF state, the broker turns off log transport services to all broker-managed standby databases by resetting the LOG_ARCHIVE_DEST_STATE_*n* initialization parameter. Transmission of redo data to all broker-managed standby databases is stopped. Log files continue to be archived at the primary database.

When transitioning to the OFFLINE state, the broker shuts down the primary database. The primary database is not available, and the broker is no longer managing the configuration. To transition out of the OFFLINE state, you need to start up the database in the mounted mode. The broker will restore the primary database to the ONLINE state.

Note: Before setting the state to OFFLINE, you should carefully consider whether or not the interruption in access to data and computing resources is necessary.

If the primary database is a RAC database, the broker configures log transport services on all primary instances with the exact same settings.

See Also: Section 3.4 for more details on managing log transport services

Physical standby database state transitions

For a physical standby database, when transitioning from any state to the ONLINE state, the broker starts log apply services with options specified by the log apply-related properties (see Section 3.5 for the property list). If the standby database is a RAC database, the broker starts log apply services on one standby instance, called the **apply instance**.

When transitioning to the LOG-APPLY-OFF state, the broker stops log apply services if the database is in the ONLINE state, or closes the database if the database is in the READ-ONLY state.

When transitioning to the READ-ONLY state, the broker stops log apply services if it is running, and opens the database for read-only access. If the standby is a RAC database, all currently active instances will be open READ-ONLY.

Note: Before you transition a physical standby database from a READ-ONLY state to any other state, it is recommended that you first close all open sessions to the standby database. If some of the sessions are not closed, the broker will automatically shut down all pending user sessions during the state transition.

When transitioning to the OFFLINE state, the broker shuts down the standby database. The standby database is not available, the broker stops log transport services to this database, and the broker stops managing this database. To transition out of the OFFLINE state, you need to start up the database in the mounted mode. The broker restores the standby database to the state it was in before the OFFLINE state. In order for the broker to once again manage a standby database that is shutdown, the primary database must be running and the broker must be managing the Data Guard configuration when the standby database is restarted.

See Also: Section 3.5 for more details on managing log apply services

Logical standby database state transitions

For a logical standby database, when transitioning from any state to the ONLINE state, the broker opens the database if it is not yet opened, turns on the guard, and starts log apply services with options specified by the log apply-related properties. If the logical standby database is a RAC database, the broker starts log apply services on one standby instance, the apply instance.

When transitioning to the LOG-APPLY-OFF state, the broker stops log apply services.

When transitioning to the OFFLINE state, the broker shuts down the logical standby database. The logical standby database is not available, the broker stops log transport services to this database, and the broker stops managing this database. To transition out of the OFFLINE state, you need to start up the database in the mounted mode. The broker restores the standby database to the state it was in before the OFFLINE state. In order for the broker to once again manage a standby database that is shutdown, the primary database must be running and the broker must be managing the Data Guard configuration when the standby database is restarted.

See Also: Section 3.5 for more details on managing log apply services

3.3 Database Properties

There are two types of properties: monitorable and configurable. Both monitorable and configurable properties can be further defined into those properties that have database scope and those having instance scope.

- Monitorable property values can be viewed only when the associated object is enabled.

  Monitorable properties allow you to view information related to database objects, but you cannot change the values of these properties.

- Configurable property values can be viewed and dynamically updated.

  Configurable properties affect the operation or configuration of the broker. You can change the value of these properties using the Data Guard CLI or the Data Guard GUI. You can edit properties if the configuration and its databases are enabled, disabled, online, or offline. However, if the database is disabled, the new property value will not take effect until you enable the configuration or database, as appropriate.

See Also: Chapter 8 for a detailed list of all database properties

To see these properties, you might use the CLI SHOW command or Edit Properties page in the GUI. The following example uses the SHOW DATABASE VERBOSE command to display information about the North_Sales database.

```
DGMGRL> SHOW DATABASE VERBOSE 'North_Sales';

Database
  Name:            North_Sales
  Role:            PRIMARY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    sales1

  Properties:
    InitialConnectIdentifier        = 'North_Sales.foo.com'
    LogXptMode                      = 'ARCH'
    Dependency                      = ''
    DelayMins                       = '0'
    Binding                         = 'OPTIONAL'
    MaxFailure                      = '0'
    ReopenSecs                      = '300'
    AsyncBlocks                     = '2048'
    NetTimeout                      = '30'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '120'
    RealTimeApply                   = 'OFF'
    ApplyNoDelay                    = 'NO'
    ApplyNext                       = '0'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '5'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = 'dbs/s2t, dbs/t'
    LogFileNameConvert              = 'dbs/s2t, dbs/t'
    StatusReport                    = '(monitor)'
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    HostName                        = 'north.foo.com'
    SidName                         = 'sales1'
    LocalListenerAddress            = '(ADDRESS=(PROTOCOL=TCP)(HOST=north.foo.com)(PORT=1514))'
    StandbyArchiveLocation          = '/archfs/arch/'
    AlternateLocation               = ''
    LogArchiveTrace                 = '4095'
    LogArchiveFormat                = 'r_%t_%s_%r.arc'
    LatestLog                       = '(monitor)'
    TopWaitEvents                   = '(monitor)'

Current status for "North_Sales":
SUCCESS
```

See Also: Chapter 7 for complete information about the Data Guard command-line interface

3.3.1 Monitorable (Read-Only) Properties

Monitorable properties allow you to view information related to the database, but you cannot change the values of these properties. These properties can be very helpful when you are trying to diagnose problems in the broker configuration. For example, view the `InconsistentLogXptProps` property to determine where there is a discrepancy for log transport services properties whose values are inconsistent between the broker configuration file and the actual value currently used by the database.

You can view all of the monitorable properties using the CLI SHOW command. For example:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'InconsistentLogXptProps';
INCONSISTENT LOG TRANSPORT PROPERTIES

INSTANCE_NAME    STANDBY_NAME    PROPERTY_NAME    MEMORY_VALUE    BROKER_VALUE
sales1          DR_Sales       DelayMins        30               -1
```

The Data Guard GUI displays the information obtained from these properties on the Edit Properties page in the GUI.

3.3.2 Configurable (Changeable) Database Properties

Configurable properties affect the operation or configuration of the database. When you use the CLI or the Data Guard GUI to create a primary database object and import existing standby databases into a new broker configuration, the property values are initially imported from the database settings.

You can update many property values when the database is either disabled or enabled. When a new database is added into the configuration, the broker connects to the database and imports initial values for the database properties from the current database settings. For example:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'ArchiveLagTarget';
  ArchiveLagTarget = '0'

DGMGRL> EDIT DATABASE 'North_Sales' SET PROPERTY 'ArchiveLagTarget'=1200;
  Property "ArchiveLagTarget" updated.

DGMGRL> SHOW DATABASE 'North_Sales' 'ArchiveLagTarget';
  ArchiveLagTarget = '1200'
```

When the configuration is enabled, the broker keeps the database property values in the broker configuration file consistent with the values being used in the database. For those that are related to initialization parameter properties, the broker maintains the consistency among the value in the broker configuration file, the current database value, and the initialization parameter value in the server parameter file, as follows:

- For dynamic parameters, the broker keeps the value of the database parameter consistent in the system global area (SGA) for the instance, in the broker configuration file, and in the server parameter file.
- For static parameters and properties, the database parameter value in the system global area (SGA) for the instances may temporarily differ from what is in the broker configuration file and in the server parameter file. Typically, the database value becomes the same as the server parameter file value and the broker configuration file value the next time the database instance is stopped and restarted.

Even when the configuration is disabled, you can update database property values through the broker. The broker retains the property settings (without validating the values) and updates the database initialization parameters in the server parameter file and the in-memory settings the next time you enable the broker configuration.

> **Note:** Even though you can change a property value when the configuration is disabled, the change does not take effect on the database unless the configuration is enabled. Also note that some property values can only be changed in the disabled state.

3.4 Managing Log Transport Services

You can manage log transport services through the following set of log transport-related configurable properties:

- `AlternateLocation`
- `AsyncBlocks`
- `Binding`
- `Dependency`
- `LogShipping`
- `LogXptMode`
- `MaxFailure`
- `NetTimeout`
- `ReopenSecs`
- `StandbyArchiveLocation`

Each standby database has a set of these properties. By setting these properties of a standby database, you tell the broker how to set up log transport services to this standby database. The actual log transport setup, such as setting the LOG_ARCHIVE_DEST_*n* initialization parameter, is carried out by the broker on the primary database (except for the StandbyArchiveLocation property). If the actual setting involves changing the LOG_ARCHIVE_DEST_*n* initialization parameter attributes, the broker forces a log switch on each thread after the setting so that the new setting is adopted immediately by the primary database's LGWR and ARCH processes.

You may also preset these properties on the primary database in preparation for it to be switched over to a standby database.

3.4.1 Managing Log Transport Services for Data Protection Modes

Section 3.6 describes how the broker handles data protection modes. As a part of the overall configuration protection mode, you must ensure that log transport services are also properly set up for the data protection mode that you choose.

You use the LogXptMode property to set the SYNC, ASYNC, or ARCH mode for log transport services. See Table 3–2 for additional information about protection modes and log transport modes.

The values for the LogXptMode property are described in the following list:

SYNC

Configures log transport services for this standby database using the LGWR, SYNC, and AFFIRM attributes of the LOG_ARCHIVE_DEST_*n* initialization parameter. This mode, along with standby redo log files, is required for the maximum protection or maximum availability protection modes. This log transport mode enables the highest grade of data protection to the primary database, but also incurs the highest performance impact.

ASYNC

Configures log transport services for this standby database using the LGWR, ASYNC, and NOAFFIRM attributes of the LOG_ARCHIVE_DEST_*n* initialization parameter. This mode, along with standby redo log files, enables a moderate grade of protection to the primary database, and lower performance impact.

ARCH

Configures log transport services for this standby database using the ARCH attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter. Standby redo log files are not required. This mode enables the lowest grade of protection to the primary database, and the lowest performance impact.

Note: When you change the protection mode, the Data Guard GUI automatically sets up standby redo log files on one or more standby databases in your configuration, and on the primary database in preparation for switchover.

3.4.2 Turning On and Off Log Transport Services

Turn log transport services on and off by setting the state of the primary database. Setting the primary database state to `ONLINE` turns on log transport services to the standby databases, and setting the primary database state to `LOG-TRANSPORT-OFF` turns off log transport services to all the standby database.

Note: Oracle does not recommend turning off log transport services to all standby databases. This increases the risk of data loss if the primary database fails.

Turn log transport services on and off to an individual standby database using the `LogShipping` property on the standby database. The `LogShipping` property accepts values `ON` and `OFF`. If you set the `LogShipping` property to `OFF` for a standby database, log transport services to this standby database are turned off, while log transport services to other databases are not affected. You can set `LogShipping` to `ON` to turn back on log transport services to the standby database.

The relationship between setting the primary database state and setting the `LogShipping` property is as follows:

- If the primary database state is set to `LOG-TRANSPORT-OFF`, log transport services to all the standby databases are turned off regardless of the individual `LogShipping` property values of the individual standby databases.
- If the primary database state is set to `ONLINE`, log transport services to each standby database are determined by the `LogShipping` property of that database.

Example 3–1 and Example 3–2 show how to turn off log transport services in two different scenarios.

Example 3–1 Turn Off Log Transport Services to All Standby Databases

```
DGMGRL> EDIT DATABASE 'North_Sales' SET STATE="LOG-TRANSPORT-OFF";
Succeeded.
DGMGRL> SHOW DATABASE 'North_Sales';

Database
  Name:             North_Sales
  Role:             PRIMARY
  Enabled:          YES
  Intended State:   LOG-TRANSPORT-OFF
  Instance(s):
    dgr

Current status for "North_Sales":
SUCCESS
```

Example 3–2 Turn Off Log Transport Services to a Specific Standby Database

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'LogShipping'='OFF';
Property "LogShipping" updated.

DGMGRL> SHOW DATABASE 'DR_Sales' 'LogShipping';
  LogShipping = 'OFF'
```

3.4.3 Managing Standby Locations to Archive the Online Redo Log Files From the Primary Database

You can set up locations on the standby database to store the archived redo log files to be used by log apply services on the standby database. This is done by setting the `StandbyArchiveLocation` and `AlternateLocation` properties on the standby database.

`StandbyArchiveLocation` specifies a standby location where the archived redo log files will be stored. The broker ensures that the archived redo log files are stored in the same specified location whether or not the standby database has standby redo log files. The broker only manages the location on the standby database to store archived redo log files received from the primary database. For archived redo log files generated locally when the database is either the primary database or a logical standby database, you need to set up local destinations directly through the `LOG_ARCHIVE_DEST_`*`n`* initialization parameter. The broker allows the value of `StandbyArchiveLocation` to be the same as the location you set up for locally generated logs, in which case the broker sets up the `VALID_FOR` attribute of the destination appropriately so that it can be used for both the archived redo log files received from the primary database and archived redo log files generated locally.

Note: On a logical standby database, Oracle recommends the LOCATION attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter for the local destination be different from the value of either the StandbyArchiveLocation or AlternateLocation property.

The broker does not manage the local destinations for locally-generated archived redo log files, but the broker does ensure that the location specified by StandbyArchiveLocation is not used to store locally-generated archived redo log files. (This is done by setting the VALID_FOR attribute of the destination to be (STANDBY_ROLE,STANDBY_LOGFILE).)

You can also set up an alternate location to store archived redo log files on the standby using the AlternateLocation property on the standby database. This is useful for avoiding disk capacity problems or disk errors when archiving the online redo log files on the standby database. AlternateLocation specifies a standby location where the archived redo log files will be stored if the location specified by the StandbyArchiveLocation fails. The broker sets up the alternate location properly using the ALTERNATE attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter.

Note: If the flash recovery area is set up on the standby database and you have configured it to store archived redo log files from the primary database, the broker no longer manages the standby locations. In this case, the broker disallows the setting of the StandbyArchiveLocation and AlternateLocation properties.

3.4.4 Setting a Dependent Standby Database

You can use the Dependency property to set up a standby database whose incoming archived redo log files depends on another standby database or on the primary database. This is useful when two standby databases are on the same system (for example, one physical and one logical standby database). This is also useful if the standby database is on the same system as the primary database, in which case one standby database can share the log files with the other database on the same system, and there is no need to archive the log files separately to this standby database.

If you want standby database B to depend on another database A (either a standby database or the primary database), you can set the Dependency property of standby database B to be the database object name (same as its DB_UNIQUE_NAME initialization parameter) of database A. The broker sets up log transport services properly through the DEPENDENCY attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more details on log transport dependency.

3.4.5 Other Log Transport Settings

You can use properties AsyncBlocks, Binding, MaxFailure, NetTimeout, and ReopenSecs to tune the performance of log transport services and to set up log transport services failure policies. These properties correspond to the LOG_ARCHIVE_DEST_*n* initialization parameter attributes. See Chapter 8 for the detailed explanations of these properties.

When creating a new standby database, the broker sets these properties with the default values. When importing an existing standby database, the broker imports existing settings corresponding to these properties if the broker finds a match between the service string of a current destination and the initial connect identifier supplied by you when you add the standby database using the command-line interface DGMGRL.

For most cases, the default values or the imported values should be sufficient for normal operations. If for some reason you need to change these property values, you can use the Data Guard broker command-line interface (DGMGRL) to set up these properties. The Data Guard GUI does not provide an interface for these properties.

3.4.6 Managing Connections to the Standby Databases for Log Transport Services

The broker automatically manages the Oracle Net connections used by log transport services to the standby databases. Log transport services managed by the broker do not rely on any Oracle Net Services naming method.

The broker constructs the connect descriptor to the standby apply instance and uses it as the value of the SERVICE attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter. The broker uses the following information to construct the connect descriptor:

- The LOCAL_LISTENER initialization parameter of the standby apply instance
- The DB_UNIQUE_NAME initialization parameter of the standby database
- The INSTANCE_NAME parameter of the standby apply instance from V$INSTANCE

On the standby database, the broker registers a special service name <DB_UNIQUE_NAME>_XPT with the standby listeners for log transport services. On the primary database, the broker extracts the listener address from the LOCAL_LISTENER initialization parameter, and uses the service name <DB_UNIQUE_NAME>_XPT and the value of INSTANCE_NAME to construct the connect descriptor so that log transport services transmit archived redo log files to the standby apply instance.

Note: You need to ensure that the LOCAL_LISTENER initialization parameter of all standby instances must resolve to an address that is reachable by all members of the configuration, and that the DB_UNIQUE_NAME initialization parameter of the standby database matches the name that you provide when you add the standby database into the broker configuration.

3.4.7 Log Transport Services in a RAC Database Environment

If your database is a RAC database, the broker sets up log transport services in the following manner:

- If the primary database is a RAC database, the broker ensures that log transport services are identical on each of the primary database instances. Each instance has the same remote destinations, and for each remote destination, all instances are set up the same in terms of log transport mode, performance related settings, and so on. If an instance has different settings, the broker raises a health check warning on that particular instance.
- If the standby database is a RAC database, only the apply instance receives log files from the primary database. If the primary database is also a RAC database, all instances of the primary database transmit log files to the same apply instance on the standby database. If the apply service is moved to a different standby instance either by the user or by the broker, the broker resets log transport services on the primary database to transmit log files to the new apply instance. See Section 3.5 for more details.

Settings relative to log transport services are saved in the broker configuration file as properties. When you update a log transport-related property on a standby database, the corresponding change is also made automatically by the broker to the `LOG_ARCHIVE_DEST_`*n* initialization parameter on all of the primary database instances. If a new instance comes up on the primary database, the broker sets up log transport services for the new instance using the log transport-related properties of all the standby databases currently being managed by the broker. After the new instance is opened for activity, all archived redo log files generated on this instance will begin to transmit to the standby databases.

> **See also:** *Oracle Data Guard Concepts and Administration* for additional information about the `LOG_ARCHIVE_DEST_`*n* initialization parameter

3.5 Managing Log Apply Services

You can manage log apply services on a physical or logical standby database through the following log apply-related configurable properties:

- Properties common to physical and logical standby databases
  - `ApplyNoDelay`
  - `ApplyInstanceTimeout`
  - `DelayMins`
  - `PreferredApplyInstance`
  - `RealTimeApply`
- Properties specific to physical standby databases
  - `ApplyNext`
  - `ApplyParallel`
- Properties specific to logical standby databases
  - `LsbyASkipTxnCfgPr`
  - `LsbyDSkipTxnCfgPr`
  - `LsbyASkipCfgPr`
  - `LsbyDSkipCfgPr`
  - `LsbyASkipErrorCfgPr`
  - `LsbyDSkipErrorCfgPr`
  - `LsbyMaxEventsRecorded`

  - `LsbyTxnConsistency`
  - `LsbyRecordSkipErrors`
  - `LsbyRecordSkipDdl`
  - `LsbyRecordAppliedDdl`
  - `LsbyMaxSga`
  - `LsbyMaxServers`

3.5.1 Managing Real-Time Apply

Data Guard log apply services can recover from standby redo log files in real time (as the log files are being filled), without requiring them to be archived at the standby database. You can turn the real-time apply feature of log apply services on and off using the `RealTimeApply` property. Setting `RealTimeApply` property to `ON` on the standby database turns on the feature and setting it to `OFF` turns off the feature.

The real-time apply feature requires the standby database to have standby redo log files configured. It applies the redo directly from standby redo log files as they are being received, instead of waiting for the log files to be archived and then applying them from the archived redo log files. This allows the standby database to be much more closely synchronized with the primary database. If the standby redo log files are not present on the standby database, the real-time apply feature is automatically turned off. When the standby database is in the real-time apply mode, any apply delay setting is automatically ignored.

Because real-time application of standby redo log files generally keeps the standby database caught up, this feature provides a number of benefits including:

- Quicker switchover and failover operations
- Having physical standby databases be instantly up-to-date after you change from Redo Apply to read-only operations

See *Oracle Data Guard Concepts and Administration* for details on the real-time apply feature.

3.5.2 Managing Delayed Apply

You can set up log apply services in a delayed apply mode. This allows the standby database to lag behind the primary database, and if a user error (for example, dropping a table) occurs during this window of time, the standby database will still contain the correct data that can be transmitted back to the primary database to repair the data.

You can set the delay time window using the `DelayMins` property which specifies, in number of minutes, how long log apply services on the standby database needs to wait before applying a log file received from the primary database. Note that only log apply services is delayed. Log transport services are not delayed and thus the primary database data is still well protected by the standby database.

If you want log apply services to ignore the delay and apply the archived redo log files immediately when they become available, you can use the `ApplyNoDelay` property. Setting `ApplyNoDelay` to `YES` overrides any delay setting in `DelayMins`, and log apply services immediately start applying all available log files. If you want to use the delay setting again, set the `ApplyNoDelay` property to `NO`. Note that it is not enough to set `DelayMins` to zero to force log apply services to start immediately applying log files. When you set `DelayMins` to zero, all the log files transmitted to the standby database (after the `DelayMins` is set) will be applied on the standby database in their turn without any additional delay, but for the log files already accumulated on the standby database while the delay setting is on, log apply services still respect the delay setting and wait until after the delay period to apply those log files.

For a physical standby database, while log apply services is in the delayed apply mode, if you want log apply services to apply a few log files immediately and then go back to the delayed mode again, you can use the `ApplyNext` property to specify the number of log files you want to apply immediately, temporarily overriding the delay. You cannot use the `ApplyNext` property on a logical standby database, because the SQL apply mechanism of a logical standby database is different. A logical standby database applies transaction by transaction rather than log file by log file as in the physical standby database.

3.5.3 Managing Parallel Apply in Physical Standby Databases

For a physical standby database, you can configure the number of parallel processes for log apply services to tune its performance by using the `ApplyParallel` property. The default value of `ApplyParallel` is `AUTO`, which means log apply services automatically choose the number of parallel processes based on the number of CPUs in the system. This should be sufficient in most cases. If you want to manually configure the parallel apply setting, you can set the `ApplyParallel` property either to `NO`, which means no parallel apply processes, or a positive integer to specify the number of parallel processes you want.

Note: The ApplyParallel property is not displayed on the Edit Properties page of the GUI.

See Also: *Oracle Database Backup and Recovery Advanced User's Guide*

3.5.4 Allocating Resources to SQL Apply in Logical Standby Databases

You can control how much SGA memory is available for the SQL apply service on a logical standby database. This can be set using the LsbyMaxSga property.

To control the number of parallel query servers used by the SQL apply service, you can use the LsbyMaxServers property.

Users can control the trade off between SQL apply performance and transaction consistency level. For a higher transaction consistency level, the SQL apply service will have a higher performance impact. To control the transaction consistency level, use the LsbyTxnConsistency property.

See Also: Section 8.2.30 and Section 8.2.34 for additional information

Changing any property pertaining to the SQL apply service will result in restarting the SQL apply service if the current database state is ONLINE (for example, SQL apply is currently running). If the current database state is LOG-APPLY-OFF, the property changes will take effect the next time the database state is changed to ONLINE.

3.5.5 Managing SQL Apply Filtering in Logical Standby Databases

One of the benefits of a logical standby database is to allow control of what to apply and what not to apply. This is done by setting up SQL apply filters. The granularity of the filtering ranges from a specific transaction to database objects to a particular class of SQL statement on a particular schema.

To add a SQL apply filter in Data Guard broker, use the LsbyASkip* properties (for example, LsbyASkipTxnCfgPr or LsbyASkipCfgPr). To delete a previously added SQL apply filter, use the LsbyDSkip* properties (for example, LsbyDSkipTxnCfgPr or LsbyDSkipCfgPr).

See Also: Chapter 8 for information on these properties

Changing these properties results in restarting the SQL apply service if the current database state is ONLINE. If the current database state is LOG-APPLY-OFF, the property changes take effect the next time the database state is changed to ONLINE.

3.5.6 Managing SQL Apply Error Handling in Logical Standby Databases

You can fine-tune SQL apply to handle apply errors on a specified set of SQL statements on particular schemas. When such SQL apply errors are encountered, Data Guard can either skip the error to continue the SQL apply operation or call a specified stored procedure at the time when the error is encountered.

To add this error handling capability, use the LsbyASkipErrorCfgPr property. To delete a previously added error handling specification, use the LsbyDSkipErrorCfgPr property.

Changing these properties results in restarting the SQL apply service if the current database state is ONLINE. If the current database state is LOG-APPLY-OFF, the property changes take effect the next time the database state is changed to ONLINE.

3.5.7 Managing the DBA_LOGSTDBY_EVENTS Table in Logical Standby Databases

On a logical standby database, the DBA_LOGSTDBY_EVENTS table records important events that happen to SQL apply. Because every logical standby database might have a different interest in the set of events to be recorded in this table, Data Guard provides a means to control the event recording. From the Data Guard broker, you can use the LsbyRecord* properties (for example, LsbyRecordSkipDdl or LsbyRecordSkipErrors) to control recording of a particular set of events. The value of these properties are either TRUE or FALSE, indicating the turning on or off of the event recording.

Changing these properties results in restarting the SQL apply service if the current database state is ONLINE. If the current database state is LOG-APPLY-OFF, the property changes take effect the next time the database state is changed to ONLINE.

3.5.8 Log Apply Services in a RAC Database Environment

If a standby database is a RAC database, only one instance of the RAC database can have log apply services running at any time. This instance is called the **apply instance**. The broker manages the selection of the apply instance. If the apply instance fails, the broker automatically moves log apply services to a different instance; this is called **apply instance failover**. In order for the apply instance to continuously apply archived redo log files received from the primary database, the broker also manages log transport services such that the archived redo log files generated on the primary database are always transmitted to the apply instance.

3.5.8.1 Selecting the Apply Instance

If you have no preference which instance is to be the apply instance in a RAC standby database, the broker randomly picks an apply instance. If you want to select a particular instance as the apply instance, there are two methods to do this.

- The first method is to pick an apply instance before there is an apply instance running in the RAC standby database. To do so, set the value of the `PreferredApplyInstance` property to the name of the instance (SID) you prefer to be the apply instance. The broker starts log apply services on the instance specified by the `PreferredApplyInstance` property when no apply instance is yet selected in the RAC standby database. This could be the case before you enable the standby database for the first time, or if the apply instance just failed and the broker is about to do an apply instance failover, or if the RAC database is currently the primary and you want to specify its apply instance in preparation for a switchover. Once the apply instance is selected and, as long as the apply instance is still running, the broker disregards the value of the `PreferredApplyInstance` property even if you change it.
- The second method is to change the apply instance when the apply instance is already selected and is running. To change the apply instance, issue the CLI `SET STATE` command to set the standby database state to `ONLINE`, with a specific apply instance argument. The `SET STATE` command will update the `PreferredApplyInstance` property to the new apply instance value, and then move log apply services to the new instance. For example, use the CLI to show the available instances for the standby database, then issue the following command to move log apply services to the new instance:

```
DGMGRL> SHOW DATABASE 'DR_Sales' ;
Database
  Name:             DR_Sales
  Role:             PHYSICAL STANDBY
  Enabled:          YES
  Intended State:   ONLINE
  Instance(s):
    dr_sales1 (apply instance)
    dr_sales2

Current status for "DR_Sales":
SUCCESS
```

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET STATE='ONLINE' WITH APPLY
INSTANCE="dr_sales2';
Succeeded.
DGMGRL> SHOW DATABASE 'DR_Sales' 'PreferredApplyInstance';
  PreferredApplyInstance = 'dr_sales2'

DGMGRL> SHOW DATABASE 'DR_Sales' ;
Database
  Name:            DR_Sales
  Role:            PHYSICAL STANDBY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    dr_sales1
    dr_sales2 (apply instance)

Current status for "DR_Sales":
SUCCESS
```

Ensure that the new apply instance is running when the command is issued. Otherwise, the apply instance remains the same.

Once the apply instance is selected, the broker keeps apply instance information in the broker configuration file so that even if the standby database is shut down and restarted, the broker still selects the same instance to start log apply services. The apply instance remains unchanged until changed by the user or it fails for any reason and the broker decides to do an apply instance failover.

3.5.8.2 Apply Instance Failover

When the apply instance fails, not only does log apply services stop applying log files to the standby database, but log transport services stop transmitting redo data to the standby database because the apply instance is not available to receive and store archived redo log files locally to the standby database. To tolerate a failure of the apply instance, the broker leverages the availability of the RAC standby database by automatically failing over log apply services to a different standby instance. The apply instance failover capability provided by the broker enhances data protection.

To set up apply instance failover, set the `ApplyInstanceTimeout` property to specify the time period that the broker will wait after detecting an apply instance failure and before initiating an apply instance failover. To select an appropriate timeout value, you need to consider:

- If there is another mechanism in the cluster that will try to recover the failed apply instance.
- How long the primary database can tolerate not transmitting redo data to the standby database.
- The overhead associated with moving the log apply services to a different instance. The overhead may include retransmitting, from the primary database, all log files accumulated on the failed apply instance that have not been applied if those log files are not saved in a shared file system that can be accessed from other standby instances.

The broker default value of the `ApplyInstanceTimeout` property is 120 seconds.

After the broker initiates an apply instance failover, the broker selects a new apply instance according to the following rule: if the `PreferredApplyInstance` property indicates an instance that is currently running, select it as the new apply instance; else pick a random instance that is currently running to be the new apply instance. The broker also resets log transport services so that the primary database starts transmitting redo data to the new apply instance.

If you do not want to use the apply instance failover feature, you can turn it off by setting `ApplyInstanceTimeout` to zero. However, we highly recommend leaving the apply instance failover feature on to provide added protection to your primary database.

3.6 Managing Data Protection Modes

The broker can simplify the process of setting up your configuration for any of the different grades of data protection: maximum protection, maximum availability, maximum performance.

This section contains the following topics to help you configure the proper protection for your configuration:

- Section 3.6.1, "Setting the Protection Mode for Your Configuration"
- Section 3.6.2, "How Broker Operations Affect Protection Modes"

3.6.1 Setting the Protection Mode for Your Configuration

To set the protection mode, perform the following steps:

Step 1 Determine which data protection mode you want to use.

Each data protection mode provides a different balance of data protection, data availability, and database performance. To select the data protection mode that meets the needs of your business, carefully consider your data protection requirements and the performance expectations of your users.

Maximum Protection This protection mode guarantees that no data loss will occur if the primary database fails. To provide this level of protection, the redo data needed to recover each transaction must be written to both the local online redo log and to the standby redo log on at least one standby database before the transaction commits. To ensure that data loss cannot occur, the primary database shuts down if a fault prevents it from writing its redo stream to at least one remote standby redo log.

Maximum Availability This protection mode provides the highest level of data protection that is possible without compromising the availability of the primary database. Like maximum protection mode, a transaction will not commit until the redo needed to recover that transaction is written to the local online redo log and to at least one remote standby redo log. Unlike maximum protection mode, the primary database does not shut down if a fault prevents it from writing its redo stream to a remote standby redo log. Instead, the primary database operates in maximum performance mode until the fault is corrected, and all gaps in redo log files are resolved. When all gaps are resolved, the primary database automatically resumes operating in maximum availability mode.

This mode guarantees that no data loss will occur if the primary database fails, but only if a second fault does not prevent a complete set of redo data from being sent from the primary database to at least one standby database.

Maximum Performance This protection mode (the default) 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 primary database's redo data stream is also written to at least one standby database, but the 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 impact on primary database performance.

The maximum protection and maximum availability modes require that a standby redo log is configured on at least one standby database in the configuration. All three protection modes require that specific log transport attributes be specified on the LOG_ARCHIVE_DEST_*n* initialization parameter to send redo data to at least one standby database. See *Oracle Data Guard Concepts and Administration* for complete information about the data protection modes.

Step 2 Set up standby redo log (SRL) files, if needed.

If the data protection mode that you need requires a standby database to use the SYNC or ASYNC log transport mode, you need to add standby redo log files to at least one standby database. (Note that the maximum performance mode does not require standby redo log files.)

The Data Guard GUI automatically prompts you to select one or more standby databases in the configuration and sets up standby redo log (SRL) files on them and on the primary database in preparation for a future switchover. See Section 5.4.3 for additional information about changing the database protection mode.

Step 3 Set the LogXptMode property, if necessary.

If the data protection mode requires that you change the log transport mode used by any of the standby databases, change the setting of the LogXptMode database property appropriately on each standby database. See Section 3.4 for more information about setting the log transport mode. Table 3–2 shows the protection modes, the corresponding log transport mode, and indicates whether or not SRLs are needed.

The Data Guard GUI automatically specifies the correct log transport mode on the primary database in preparation for a future switchover. See Section 5.4.3 for additional information about changing the database protection mode.

Table 3–2 Data Guard Protection Modes and Requirements

Protection Mode	Log Transport Mode	Standby Redo Log Files Needed?
MAXPROTECTION	SYNC	Yes
MAXAVAILABILITY	SYNC	Yes
MAXPERFORMANCE	ASYNC or ARCH	Yes for ASYNC

Step 4 Set the protection mode.

Select the protection mode using the CLI or the Data Guard GUI. See Section 6.5 for additional information about setting the configuration protection mode.

With the CLI:

1. If you plan to set the protection mode to either the `MAXPROTECTION` or `MAXAVAILABILITY`, ensure that standby redo log files are configured on the standby database. Do this also for the primary database or another standby database in the configuration to ensure that it can support the chosen protection mode after a switchover.
2. Use the `EDIT DATABASE (property)` command and specify the standby database whose log transport mode should be changed to correspond to the protection mode you plan to set. For example, if you plan to set the overall Data Guard configuration to the `MAXAVAILABILITY` mode, you must use the `EDIT DATABASE` command to set the `SYNC` mode for log transport services. For example:

   ```
   DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'LogXptMode'='SYNC';
   ```

   Do this also for the primary database or another standby database in the configuration to ensure that it can support the chosen protection mode after a switchover.
3. Use the `EDIT CONFIGURATION SET PROTECTION MODE AS` *`protection-mode`* command to set the overall configuration protection mode. For example:

   ```
   DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MAXAVAILABILITY;
   ```

With the Data Guard GUI:

1. From the Data Guard GUI overview page, click the link to the right of the Protection Mode label.
2. Select Maximum Protection, Maximum Availability, or Maximum Performance and click Continue.
3. If prompted, log in to the database with SYSDBA privileges and click Login.
4. Select one or more standby databases to support the protection mode that you selected. If standby redo log files are needed, verify the names of the log files. Click OK.
5. From the Confirmation page, click Yes.

After you upgrade the protection mode using either the CLI or the Data Guard GUI, the primary database will be restarted automatically. The primary database need not be restarted following a downgrade of the protection mode.

3.6.2 How Broker Operations Affect Protection Modes

This section describes how operations such as switchover, failover, disabling, or enabling the Data Guard configuration can have an effect on the configuration's protection mode and log transport services. This section contains the following sections:

- Upgrading or Downgrading the Current Protection Mode
- Switchover Operations
- Failover Operations
- Disable and Enable Operations
- Requirements When Removing a Database from the Configuration
- Requirements On Other Operations

3.6.2.1 Upgrading or Downgrading the Current Protection Mode

When you upgrade the current Data Guard protection mode (for example, you might want to upgrade from the maximum performance mode to the maximum availability mode), the broker shuts down and restarts the primary database. When you downgrade the current Data Guard protection mode, the database does not need to be restarted. Follow these recommendations when upgrading or downgrading the Data Guard protection mode:

- When upgrading the protection mode, upgrade the log transport mode before you upgrade the overall protection mode. (The Data Guard GUI does this for you. See Section 5.4.3 for information.) At the time when you change the protection mode or reset the log transport mode of a standby database, the broker verifies that there is at least one standby database in the configuration that can support the requested grade of protection. If not, then the broker does not change the protection mode and returns an error.
- When downgrading the protection mode, downgrade the protection mode first and then change the log transport mode (if necessary). The broker will not allow changing the log transport mode if doing so invalidates the current overall protection mode.

If you upgrade the protection mode from the maximum performance mode to the maximum protection mode, the broker ensures that there is at least one standby database using standby redo log files, and whose log transport mode is set to SYNC. If there are no standby databases in the configuration that meet these requirements, the request to upgrade the protection mode is rejected with an error.

3.6.2.2 Switchover Operations

A switchover does not change the overall Data Guard protection mode. The protection mode remains the same as it was before the switchover.

This requires that there be a standby database that is properly configured to support the current protection mode once the switchover completes. This can be either another standby database in the configuration or the current primary database that will become a standby database after the switchover completes.

Before you invoke a switchover, if necessary, you can add standby redo log files and set the log transport mode on the current primary database, or on another standby database in the configuration, to the SYNC, ASYNC, or ARCH mode that is required to support the Data Guard protection mode. Then, when the switchover begins, the broker verifies the presence of standby redo log files and the log transport mode setting on each standby database and on the current primary database. If the verification is successful, the switchover continues; otherwise, the switchover fails, and the database roles and the broker configuration files remain unchanged.

3.6.2.3 Failover Operations

After a failover, the Data Guard protection mode is always downgraded to maximum performance mode. You can upgrade the protection mode later, if necessary. The log transport modes of the standby databases remain unchanged.

3.6.2.4 Disable and Enable Operations

When you disable broker management of a standby database, the broker checks to see if the overall protection mode can still be satisfied by any of the remaining standby databases. If not, the broker rejects the disable operation. Otherwise, the broker allows the disable operation to proceed.

Caution: If you disable broker management of a standby database in the broker configuration, that standby database cannot be used by the broker as a failover target in the event of loss of the primary database.

After a standby database is successfully disabled, you can change the log transport mode for that database and the broker will record the change in the broker configuration file. The change will not affect the overall protection mode because it is guaranteed that at least one of the enabled standby databases already satisfies the overall protection mode requirement.

You can disable the entire configuration regardless of the protection mode. This is because you may want to use the broker only to set up a Data Guard configuration, and then disable it from the broker's control and use other interfaces (for example, using SQL*Plus and SQL statements) for management.

If the entire configuration is disabled, you can change any broker settings, including the log transport modes of the standby databases and the protection mode of the configuration. The broker saves the changes in the broker configuration file, but the changes will not be made to the database itself.

When enabling broker management of the entire configuration, the broker first checks to see if the protection mode will be satisfied by the log transport modes of the standby databases that will be enabled. If not, the enable operation fails and the configuration remains disabled. Otherwise, the enable operation successfully enables the configuration, and the broker enables the database using the settings saved in the broker configuration file.

3.6.2.5 Requirements When Removing a Database from the Configuration

When removing a standby database from the broker configuration, the broker checks to see if the protection mode will still be satisfied. The broker quits the operation if removing the database compromises the protection mode. If you want to remove the entire configuration, the broker always allows the operation.

3.6.2.6 Requirements On Other Operations

Some operations that take place in a broker configuration, especially operations related to log transport services, can affect the overall protection mode. These operations include:

- Setting the standby database to the offline state
- Stopping log transport services on the primary database
- Stopping log transport services to individual standby databases

Before any of these operations can proceed, the broker checks to see if the protection mode will be supported by the log transport mode settings on the standby databases after the operation completes. If not, the broker quits the operation and returns an error.

3.7 Database Status

Database status reveals the health of the database. In general, the broker checks the health of a database by verifying if the actual database state and settings match with those described in the broker configuration file. This is done by checking if any component of the Data Guard configuration is functioning incorrectly (for example, if log transport services have an error), and by checking if other required database settings are correctly set (for example, if the server parameter files are available and if the `ARCHIVELOG` mode is turned on). The following is a detailed list of what is being checked by the broker on a primary database and a standby database.

On a primary database, the health check includes checking if the:

- Database is in the state specified by the user, as recorded in the broker configuration file.
- Database is in the correct data protection mode.
- Database is using a server parameter file.
- Database is in the `ARCHIVELOG` mode.
- Logical standby guard is turned off.
- Supplemental logging is turned on when there is a logical standby database in the configuration.
- Log transport services have any errors.
- Database settings match those specified by the broker configurable properties.
- Log transport settings match those specified by the log transport-related properties of the standby databases.

On a standby database, the health check includes checking if the:

- Database is in the state specified by the user, as recorded in the broker configuration file.
- Database is using a server parameter file.
- Database settings match those specified by the broker configurable properties.
- Logical standby guard is turned on when the database is a logical standby database.

The following monitorable properties can be used to query the database status:

- `StatusReport`
- `LogXptStatus`
- `InconsistentProperties`
- `InconsistentLogXptProps`

Note: These properties are directly accessed through the broker command-line interface (DGMGRL). The Data Guard GUI rearranges the values of these properties and presents them more simply in the GUI.

The `StatusReport` property provides a list of all health check problems the broker detected during a health check. This is usually the first property you use to check the database status. In the following example, you see three items reported by the `StatusReport` property.

```
DGMGRL> SHOW DATABASE North_Sales 'StatusReport';
STATUS REPORT
       INSTANCE_NAME   SEVERITY ERROR_TEXT
               sales1      ERROR ORA-16737: The log transport service for
standby "reportdb2" has an error.
               sales2      ERROR ORA-16737: The log transport service for
standby "reportdb2" has an error.
               sales2      WARNING ORA-16715: Log transport related property
MaxFailure of standby "reportdb2" is inconsistent.
```

To further check the details about the database status, you can use the `LogXptStatus`, `InconsistentProperties`, and `InconsistentLogXptProps` properties. `LogXptStatus` lists all log transport errors detected on all instances of the primary database. `InconsistentProperties` lists all properties that have inconsistent values between the broker configuration file and the database settings. `InconsistentLogXptProps` lists all log transport-related properties of standby databases that have inconsistent values between the broker configuration file and the log transport settings. For example, the output of `StatusReport` (in the previous example) shows two problems: some log transport services errors and an inconsistent log transport-related property. Issue the following queries to obtain further details about the problems.

```
DGMGRL> SHOW DATABASE North_Sales 'LogXptStatus';
LOG TRANSPORT STATUS
PRIMARY_INSTANCE_NAME STANDBY_DATABASE_NAME               STATUS
               sales1            reportdb2 ORA-12514: TNS:listener could not
resolve SERVICE_NAME given in connect descriptor
               sales2            reportdb2 ORA-12514: TNS:listener could not
resolve SERVICE_NAME given in connect descriptor

DGMGRL> SHOW DATABASE North_Sales 'InconsistentLogXptProps';
INCONSISTENT LOG TRANSPORT PROPERTIES
```

```
  INSTANCE_NAME        STANDBY_NAME       PROPERTY_NAME        MEMORY_VALUE
BROKER_VALUE
      sales2         reportdb2        MaxFailure                9
0
```

See Also: Chapter 8 for detailed information about properties

4 Role Management

This chapter describes how the broker manages databases during switchover and failover. This chapter contains the following sections:

- Section 4.1, "Managing Switchover Operations"
- Section 4.2, "Managing Failover Operations"

When the primary database (or all instances of the RAC primary database) fails, such as when a system or software failure occurs, you may need to transition one of its corresponding standby databases to take over the primary role by performing a failover. Even in the absence of such a failure, you may have reason (for example, planned hardware maintenance) to perform a switchover to direct one of the standby databases to assume the role of being the primary database, while the former primary database assumes the role of being a standby database.

Without the broker, failover and switchover are manual processes that can be automated only by using script-based solutions. For example, if a physical standby database is in read-only mode (log apply services are offline) when a failure occurs on the primary database, you must change the standby database to Redo Apply mode, apply archived redo log files that have not yet been applied to the standby database, and fail over the standby database to the primary role.

The broker simplifies the switchover or failover by allowing you to invoke them through a single command and then coordinating role transitions on all databases in the configuration.

4.1 Managing Switchover Operations

You can switch a database from the primary role to the standby role, as well as from standby to primary, without resetting the online redo log files of the associated new primary database. This is known as a database switchover, because the standby database that you specify becomes the primary database, and the original primary database becomes a standby database. There is no loss of application data, the data is consistent between the original, and the new primary database after the switchover completes.

Whenever possible, you should switch over to a physical standby database:

- If the switchover transitions a physical standby database to the primary role, then the original primary database will be switched to a physical standby role. The online redo log files are continuously archived from the new primary database to all standby databases in the configuration.
- If the switchover transitions a logical standby database to the primary role, then the original primary database will be switched to a logical standby role. If there are physical standby databases in the configuration not involved in the switchover, they will not be able to serve as standby databases to the new primary database, because a new incompatible online redo log stream has started.

Warning: Switching over to a logical standby database results in the physical standby databases in the broker configuration being disabled by the broker. These are no longer viable as standby databases. Section 4.2.5 describes how to restore their viability as standby databases.

- If the switchover transitions a physical standby database to the primary role, then both the primary database and the target standby database will be restarted after the switchover completes.
- If the switchover transitions a logical standby database to the primary role, neither the primary database nor the logical standby database needs to be restarted after the switchover completes.

4.1.1 Before You Perform a Switchover Operation

Consider the following points before you begin a switchover:

- When you start a switchover, the broker verifies that at least one standby database (including the primary database that is about to be transitioned to the standby role) is configured to support the overall protection mode (maximum protection, maximum availability, or maximum performance).

- You should prepare the primary database in advance for its possible future role as a standby database in the context of the overall protection mode (see Section 3.6). The preparation includes:
  - Setting up standby redo log files on the primary database if you intend to use `SYNC` or `ASYNC` log transport mode to the database after the switchover.
  - Preset log transport services related properties, such as `LogXptMode`, `NetTimeout`, `StandbyArchiveLocation`, and `AlternateLocation`. For more details about managing log transport services using configurable properties, see Section 3.4.
  - Preset log apply services related properties, such as `RealTimeApply` and `ApplyParallel`. For more details about managing log apply services using configurable properties, see Section 3.5.

  Note that the broker does not use these properties to set up log transport and log apply services until you actually switch over the primary database to the standby role. Thus, the validity of the values of these properties is not verified until after the switchover. Once you set these properties, their values persist through role changes during switchover and failover.

- After a switchover completes, the overall Data Guard protection mode (maximum protection, maximum availability, or maximum performance) remains at the same protection level it was before the switchover. Also, the log transport mode (`SYNC`, `ASYNC`, or `ARCH`) of other standby databases not involved in the switchover does not change after a switchover. Log apply services for all other standby databases not involved in the switchover automatically begin applying archived redo log files from the new primary database.
- If there are both logical and physical standby databases in the configuration and the switchover occurs to a logical standby database, you need to re-create all physical standby databases, as described in Section 4.2.5.

4.1.2 Starting a Switchover Operation

The act of switching roles should be a well-planned activity. The primary and standby databases involved in the switchover should have as small a transactional lag as possible. Oracle highly recommends that you consider performing a full, consistent backup of the primary database before starting the switchover. (*Oracle Data Guard Concepts and Administration* provides detailed information about setting up the databases in preparation of a switchover.)

To start a switchover using the Data Guard GUI, select the standby database that you want to change to the primary role and click **Switchover**. When using the CLI, you need to issue only one `SWITCHOVER` command to specify the name of the standby database that you want to change into the primary role.

The broker controls the rest of the switchover, as described in Section 4.1.3.

4.1.3 How the Broker Performs a Switchover Operation

Once you start the switchover, the broker:

1. Verifies that the primary and the target standby databases are in the following states:

   a. The primary database must be enabled and in the `ONLINE` state.

   b. The participating standby database must be enabled and in the `ONLINE` state.

   The broker allows the switchover to proceed as long as there are no errors for the primary database and the standby database that you selected to participate in the switchover operation. Errors occurring for any other standby databases not involved in the switchover will not impede the switchover.

2. Shuts down all instances except one.

   If the primary database is a RAC database, the broker will keep only one instance running and shut down all other instances before it continues the switchover. If the standby database you want to switch to the primary role is a RAC database, the broker will shut down all instances except the apply instance before it continues the switchover. If those other instances cannot be shut down, the switchover fails. In this case, you must manually shut down those other instances and issue the switchover command again. It is also important that you do not start any new instances during the switchover.

3. Switches roles between the primary and standby databases.

   The broker first converts the original primary database to run in the standby role. Then, the broker transitions the target standby database to the primary role. If any errors occur during either conversion, the broker stops the switchover. See Section 9.4 for more information.

4. Updates the broker configuration file to record the change in roles.

   Because the configuration file profiles all database objects in the configuration, this ensures that each database will run in the correct role should it be restarted later for any reason.

5. Restarts the new standby database if the switchover operation occurs with a physical standby database, and log apply services begin applying archived redo log files transmitted from the new primary database. If this is a RAC database, the broker restarts the instances that it shut down prior to the switchover.

6. Restarts the new primary database if it was a physical standby database, opens it in read/write mode, and starts log transport services transmitting redo data to the archived redo log files for the standby databases, including to the former primary database. If the switchover occurs to a logical standby database, there is no need to restart any databases. If this is a RAC database and a restart was necessary, the broker restarts the instances that it shut down prior to the switchover.

The broker verifies the state and status of the databases to ensure that the switchover successfully transitioned the databases to their new role correctly. Standby databases not involved in the switchover and not disabled by the broker after the switchover will continue operating in the state they were in before the switchover. For example, if a physical standby database was in read-only mode, it will remain in that mode after switchover completes. Log apply services for all other standby databases not involved in the switchover automatically begin applying archived redo log files from the new primary database.

4.2 Managing Failover Operations

Database failover transitions one of the standby databases to the role of primary database. You should perform a failover only when a catastrophic failure occurs on the primary database, and there is no possibility of recovering the primary database in a timely manner. The failed primary database is discarded, and the target standby database assumes the primary role.

The broker supports two methods of failover:

- Complete failover (`FAILOVER TO` *`database-name`*`;`)

  This is the recommended and default failover option that automatically recovers the maximum amount of data for the protection mode of the original primary database application data and attempts to bring along any standby databases not involved in the failover to continue serving as standby databases to the new primary database:

  – After failover to a physical standby database, the original primary database must be re-created to act as a standby database for the new primary database. In addition, some standby databases may be disabled by the broker during the failover if the broker detects that they have applied log sequences beyond the new primary database. Any database that was disabled by the broker must be re-created using the steps described in Section 4.2.5.

  – After failover to a logical standby database, the original primary database and any physical standby databases in the configuration must be re-created to act as a standby database for the new primary database. Additionally, if there is a gap in the log sequence, the logical standby databases not involved in the failover cannot finish applying all of the redo data that the target logical standby database applied before the failover. This results in the logical standby database being disabled. Any database that was disabled by the broker must be re-created using the steps described in Section 4.2.5.

- Immediate failover (`FAILOVER TO` *`database-name`* `IMMEDIATE;`)

  > **Caution:** Do not perform an immediate failover to a standby database except in an emergency.

  A consequence of the immediate failover is that you must re-create the original primary database and all other standby databases not involved in the failover before they can serve as standby databases to the new primary database. Section 4.2.5 describes how to do this. Another consequence is that there may be lost application data.

Depending on the destination attributes of log transport services, the result of a complete failover may provide no data loss or minimal data loss. An immediate failover may result in data loss. Always try to perform a complete failover. Only when a complete failover is unsuccessful should you perform an immediate failover.

> **Note:** After a failover (complete or immediate), the overall Data Guard protection mode is always reset to the maximum performance mode. The log transport mode (`SYNC`, `ASYNC`, or `ARCH`) of the other standby databases not involved in the failover does not change. You can subsequently upgrade the protection mode as described in Section 3.6.1.

If the standby database you want to fail over to the primary role is a RAC database, the broker will shut down all instances except the apply instance before it continues the failover. If the broker cannot shut down the instances, the failover fails. In this case, you must manually shut down all instances except the apply instance and issue the `FAILOVER` command again. It is also important that you do not start any new instances during the failover. The broker restarts instances that it shut down prior to the failover.

4.2.1 Considerations When Selecting the Failover Target

There are many considerations when selecting a standby database to be the next primary database in a failover. Each production site might have unique requirements. This section discusses various considerations of which you need to be aware. Data Guard broker does not enforce a particular choice of standby database as the next primary database in the failover. You will need to make the decision and should be able to get the information you need from Data Guard broker to assist with this decision.

If you have a configuration that only has one type of standby database (either all physical or all logical standby databases), your choice of the target standby database for the failover should be decided based on the following criteria:

- Which standby database has the most data archived to it. Failover to this standby database will result in the least amount of (or no) data loss.
- Which standby database has the least amount of online redo log files to be applied before it can become the primary. (This can be done with the CLI by connecting to one of the standby databases in the configuration and reading the value of the `RecvQEntries` property on different standby databases. Using the Data Guard GUI, you can view the value of the `LastAppliedLog` column for each standby database in the Standby Databases section of the overview page.) This will affect the downtime of your primary database. The longer it takes to failover to the target standby, the longer the downtime would be.
- Logistically, which standby database is a more desirable database to assume the role of primary.

If you have a configuration that has both physical and logical standby databases, you may want to consider choosing one of your physical standby databases as the failover target. Failing over to a logical standby database has the side effect of invalidating all your physical standby databases. Therefore, it is always better to choose a physical standby database as the failover target. This will avoid the need to re-create other physical standby databases not involved in the failover after the failover to a logical standby database completes.

However, there may be exceptions to this rule when one of the previous criteria is an important consideration for your failover. For example, if your physical standby database is lagging behind your logical standby database (the physical standby database was in READ-ONLY state) and your business requires minimum downtime, you may want to consider selecting your logical standby database as the failover target.

Once you decide which standby database is your target, you can use Data Guard broker to execute the failover.

4.2.2 Starting a Failover Operation

To start a failover using the Data Guard GUI, select the standby database that you want to change to the primary role and click **Failover**. When using the CLI, you issue a FAILOVER command that specifies the name of the standby database that you want to change into the primary role.

After the failover, the overall protection mode of the new configuration (maximum protection, maximum availability, or maximum performance) is reset to the maximum performance mode.

The broker controls the failover steps described in Section 4.2.3. However, the failover results in eliminating the previous primary database as an active participant in the configuration. Depending on whether or not there are other valid standby databases in the configuration, you may need to perform the additional recovery steps described in Section 4.2.5 to maintain a viable disaster recovery solution in the event of another disaster.

4.2.3 How the Broker Performs a Complete Failover Operation

After determining that there is no possibility of recovering the primary database in a timely manner, ensure that the primary database is shut down and then begin the failover operation.

Once you start a complete failover, the broker:

1. Checks to see if the primary database is still available and, if so, issues a warning message asking whether you want to continue with the failover operation. If you choose to continue, the primary database is shutdown.
2. Verifies that the target standby database is enabled. If the database is not enabled, you will not be able to perform a failover to this database. If the target is a RAC standby database, the broker shuts down all instances except the apply instance.
3. Waits for the target standby database to finish applying any remaining archived redo log files before stopping log apply services on it.

4. Updates the broker configuration file to record the change in role.

   If a standby database not involved in the failover is not disabled by the broker during this failover, it will remain in the state it was in before the failover. For example, if a physical standby database was operating in read-only mode, it will remain in read-only mode.

   > **Note:** Standby databases not involved in the failover may be disabled by the broker during a failover, and they must be re-created in the configuration before they can serve as standby databases to the new primary database. A failover to a logical standby database will result in all physical standby databases being disabled by the broker, and it may result in all logical standby databases being disabled by the broker. See Section 4.2.5 for additional information.

5. Transitions the target standby database into the primary role, opens the new primary database in read/write mode, determines whether or not any standby databases that did not participate in the failover operation have applied log sequences beyond the new primary database and thus need to be re-created, and starts log transport services to begin transmitting redo data to all standby databases not involved in the failover and not required to be re-created.
6. If the target is a RAC standby database, the broker restarts instances that it shut down prior to the failover.

The broker allows the failover to proceed as long as there are no errors for the standby database that you selected to participate in the failover. Errors occurring for any standby databases not involved in the failover will not stop the failover. If you initiated a complete failover and it fails, you might need to restart it as an immediate failover.

4.2.4 How the Broker Performs an Immediate Failover Operation

After determining that there is no possibility of recovering the primary database in a timely manner, ensure that the primary database is shut down and then begin the failover operation.

Once you start an immediate failover, the broker:

1. Verifies that the target standby database is enabled. If the standby database is not enabled for management by the broker, then the failover cannot occur.

2. Stops log apply services on the standby database immediately, without waiting for log apply services to finish applying the available archived redo log files. Note that this may result in some data loss.
3. Updates the broker configuration file to record the change in role.
4. Transitions the target standby database into the primary role, opens the new primary database in read/write mode, and starts log transport services.

   Because an immediate failover starts a new incompatible redo branch from the new primary database, all the standby databases in the configuration are disabled by the broker and must be re-created. See Section 4.2.5.

The broker allows the failover to proceed as long as there are no errors for the standby database that you selected to participate in the failover.

4.2.5 Re-creating a Viable Disaster Recovery Solution After Failover

You must perform recovery steps after the failover completes:

- After a failover operation completes, the original, failed primary database is disabled by the broker until such time as the database can be re-created as a standby to the new primary database.
- After a complete failover finishes, any of the standby databases not involved in the failover that are determined to be unviable as a standby for the new primary database will be disabled by the broker.

  For instance, this could happen if a standby database not involved in the failover finds that it has applied more log files than the new primary database itself has applied. This standby database must be re-created before it can serve as a standby for the new primary database.
- After a complete failover to a logical standby database finishes, the broker disables all of the physical standby databases in the configuration. They must be re-created before they can serve as standby to the new primary database.
- After an immediate failover completes, the new primary database starts a new incompatible redo branch, causing all the standby databases in the configuration, regardless of their type, to be disabled. They must be re-created before they can serve as standby to the new primary database.

A database that has been disabled by the broker can be brought back to broker operation by:

- Re-creating the database, either through flashback instantiation or by creating the database as described in *Oracle Data Guard Concepts and Administration*.

- Enabling broker management of the re-created standby database. For example, use the CLI `ENABLE DATABASE` command.

Note: Databases can be flashback instantiated only if the Flashback Database feature was enabled and flashback logs are available. If failover was to a physical standby database, all databases can be flashback instantiated. If failover was to a logical standby database, the old primary database and any other logical standby databases can be flashback instantiated. The physical standby databases will need to be re-created as described in *Oracle Data Guard Concepts and Administration*.

The newly re-created standby database will begin serving as standby to the new primary database.

5 Data Guard Scenarios - Using Oracle Enterprise Manager

This chapter provides several scenarios that show how to use the Oracle Data Guard graphical user interface (GUI) to create, manage, and monitor a Data Guard configuration.

This chapter contains the following scenarios:

5.1 Scenario 1: Starting the Data Guard GUI

Start the Data Guard GUI through the Oracle Enterprise Manager Grid Control using the following steps:

1. Click the **Targets** tab to go to the Targets page.
2. Click **Databases** to go to the Databases page.

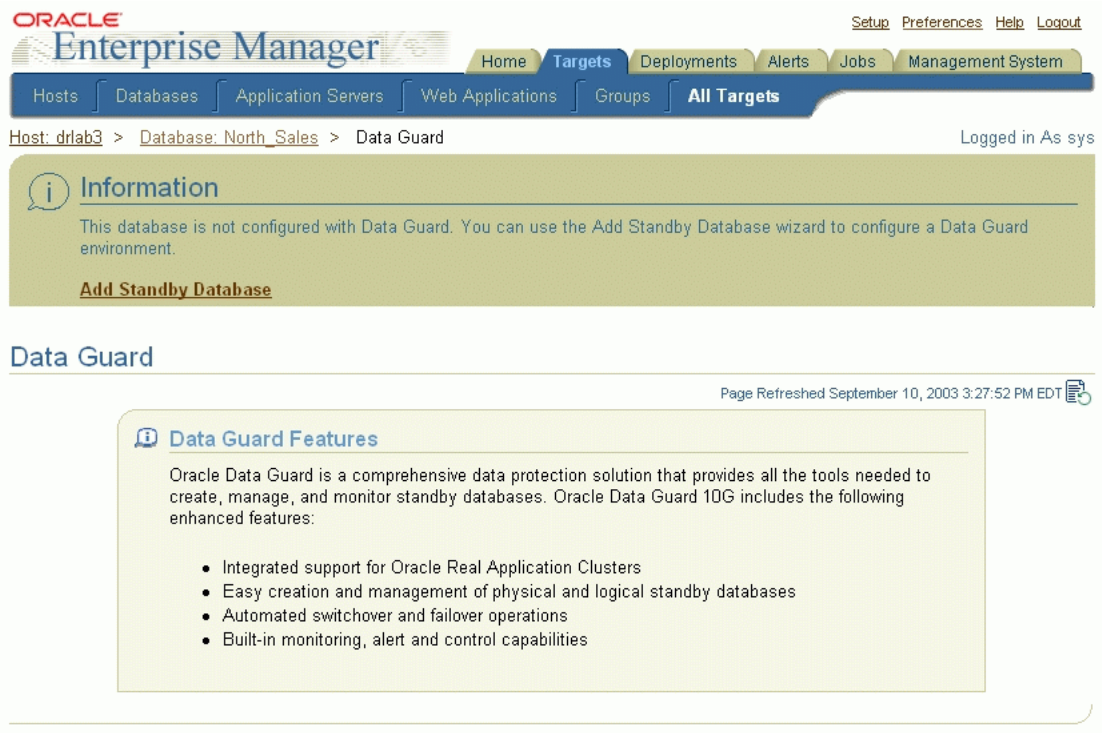
3. On the Databases page, you see a listing of all discovered databases including the primary database (in this scenario the primary database is called North_Sales). Click the primary database to go to the primary database home page.
4. Click **Administration**.
5. Under the High Availability section, click **Data Guard** and log in.

> **Note:** If you log in as a user with SYSDBA privileges, you will have access to all Data Guard functionality, including all monitoring and management features. If you log in as a non-SYSDBA user, you will have access to monitoring functions only; features such as standby creation, switchover, and failover will not be available.

If the primary database is not already in a broker configuration, clicking Data Guard in Step 5 will go to the page shown in Figure 5–1. From this page you can create a broker configuration and add a standby database.

See Also: Section 5.2, "Scenario 2: Creating a Configuration or Adding an Additional Standby Database"

Figure 5–1 Create a Broker Configuration



Information

This database is not configured with Data Guard. You can use the Add Standby Database wizard to configure a Data Guard environment.

Add Standby Database

Data Guard

Page Refreshed September 10, 2003 3:27:52 PM EDT

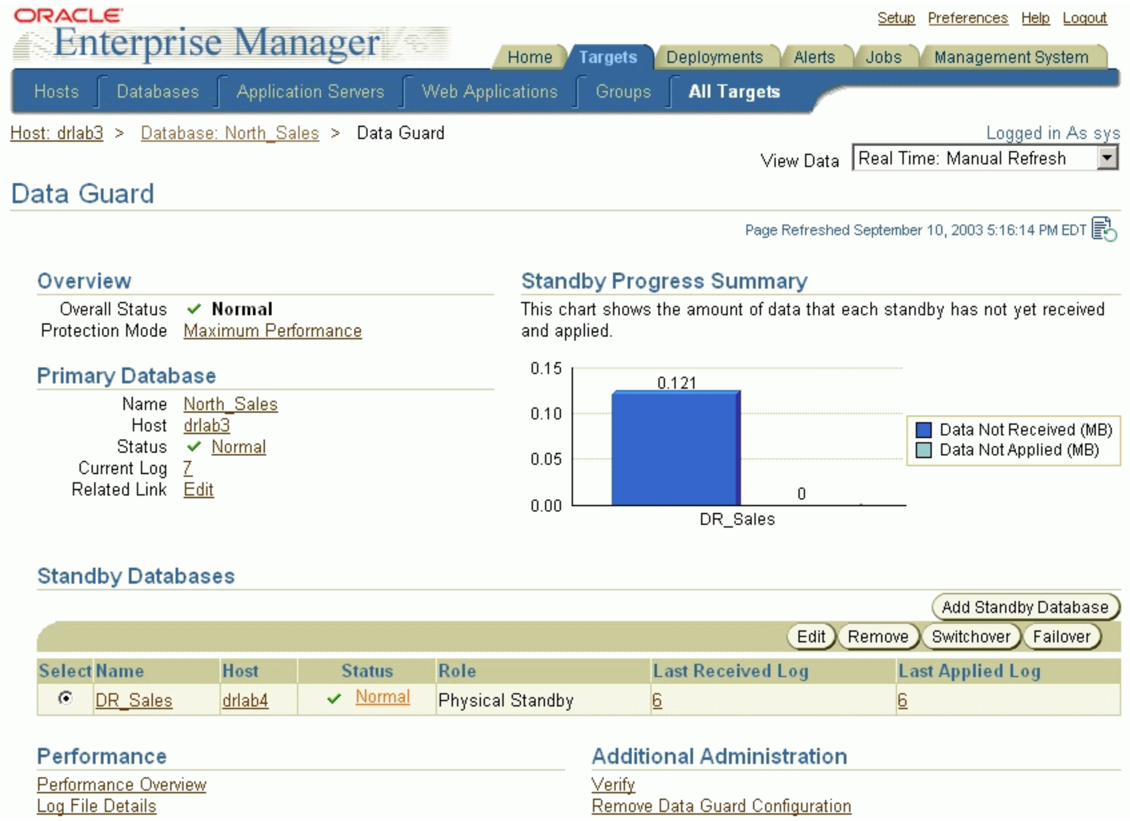
Data Guard Features

Oracle Data Guard is a comprehensive data protection solution that provides all the tools needed to create, manage, and monitor standby databases. Oracle Data Guard 10G includes the following enhanced features:

- Integrated support for Oracle Real Application Clusters
- Easy creation and management of physical and logical standby databases
- Automated switchover and failover operations
- Built-in monitoring, alert and control capabilities

If the primary database is already part of a broker configuration, clicking Data Guard in Step 5 will go to the Data Guard Overview page as shown in Figure 5–2.

Figure 5–2 Data Guard Overview Page


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Host: drlab3 > Database: North_Sales > Data Guard
Logged in As sys
Add Standby Database
Edit Remove Switchover Failover
0.15
0.10
0.05
0.00
0.121
0
DR_Sales
🟦 Data Not Received (MB)
▨ Data Not Applied (MB)

View Data Real Time: Manual Refresh

Data Guard

Page Refreshed September 10, 2003 5:16:14 PM EDT

Overview

Overall Status ✓ **Normal**
Protection Mode Maximum Performance

Primary Database

Name North_Sales
Host drlab3
Status ✓ Normal
Current Log 7
Related Link Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	DR_Sales	drlab4	✓ Normal	Physical Standby	6	6

Performance

Performance Overview
Log File Details

Additional Administration

Verify
Remove Data Guard Configuration

On the Data Guard Overview page, you can:

- Edit the protection mode—click **Protection Mode** in the Data Guard Overview section to view the current protection mode setting and, if necessary, change the protection mode.
- View a summary of standby progress—this chart shows the amount of data that each standby database has not yet received and applied.
- Retrieve information about the primary database—click **Name**, **Host**, **Status**, **Current Log**, or **Related Link (Edit)** to view pertinent information about the primary database.

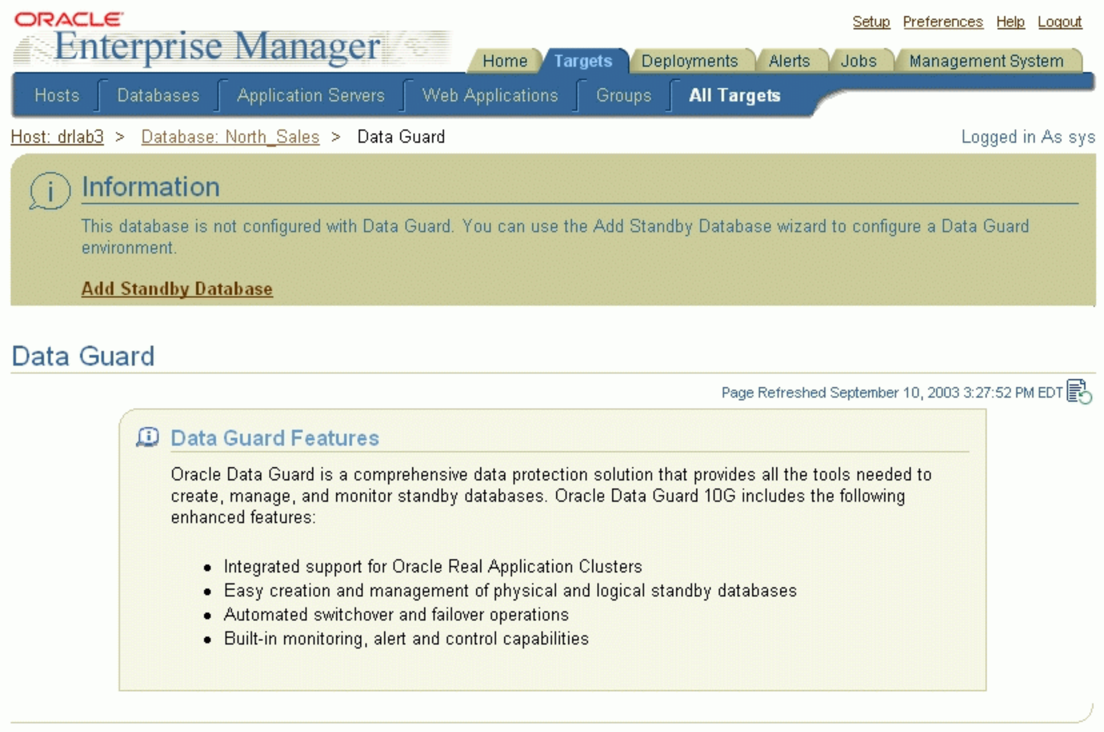
If you click **Status** or **Edit**, you are taken to the Edit Properties page where you can view and change the current state or properties of the primary database.

- View or change information on the standby databases:
  - Add a standby database to the broker configuration—click **Add Standby Database** to add a physical or logical standby database.
  - Change the state or properties—click **Edit** to go to the Edit Properties page to view and change the current state or properties of the standby database.
  - Discontinue Data Guard broker control—click **Remove** to remove the selected standby database from Data Guard broker control.
  - Switch the role from standby to primary—click **Switchover** to switch the database roles from standby to primary.
  - Transition the standby database to the role of primary database— click **Failover** when a catastrophic failure occurs on the primary database, and there is no possibility of recovering the primary database in a timely manner. The target standby database assumes the primary role, and the failed primary database is disabled by the broker. See Section 4.2.5 for steps to re-create the failed primary database as a standby database to the new primary database.
- View performance information for the configuration and status of online redo log files for each standby database:
  - Click **Performance Overview** to view information about data archived and applied, a summary of standby progress, and a summary of log services.
  - Click **Log File Details** to view the status of online redo log files that were generated on the primary database but not received by the standby, and for online redo log files that were received but not applied to the standby database
- Perform additional administrative activities:
  - Click **Verify** to check the protection mode and properties, confirm that standby redo log files are present, and verify log switch.
  - Click **Remove Data Guard Configuration** to remove the broker configuration from Data Guard broker control.
- Search for information about any Data Guard GUI or Enterprise Manager topic—Click **Help** to display the Welcome to Oracle Data Guard GUI help topic. Once in the help system, you can use the Contents page or Search page to locate help topics of interest.

5.2 Scenario 2: Creating a Configuration or Adding an Additional Standby Database

Creating a broker configuration is the first thing you must do before you can manage and monitor the databases. The Data Guard GUI provides the Add Standby Database wizard to create a broker configuration that includes a primary database and one standby database. (You can use the Add Standby Database wizard later to add more databases. See Section 5.3.) To start the Add Standby Database wizard, click **Add Standby Database** as shown in Figure 5–3.

Figure 5–3 Create a Configuration



Information

This database is not configured with Data Guard. You can use the Add Standby Database wizard to configure a Data Guard environment.

Add Standby Database

Data Guard

Page Refreshed September 10, 2003 3:27:52 PM EDT

Data Guard Features

Oracle Data Guard is a comprehensive data protection solution that provides all the tools needed to create, manage, and monitor standby databases. Oracle Data Guard 10G includes the following enhanced features:

- Integrated support for Oracle Real Application Clusters
- Easy creation and management of physical and logical standby databases
- Automated switchover and failover operations
- Built-in monitoring, alert and control capabilities

After clicking **Add Standby Database**, select one of the three available options:

- Create a new physical standby database (single-instance only)
- Create a new logical standby database (single-instance only)

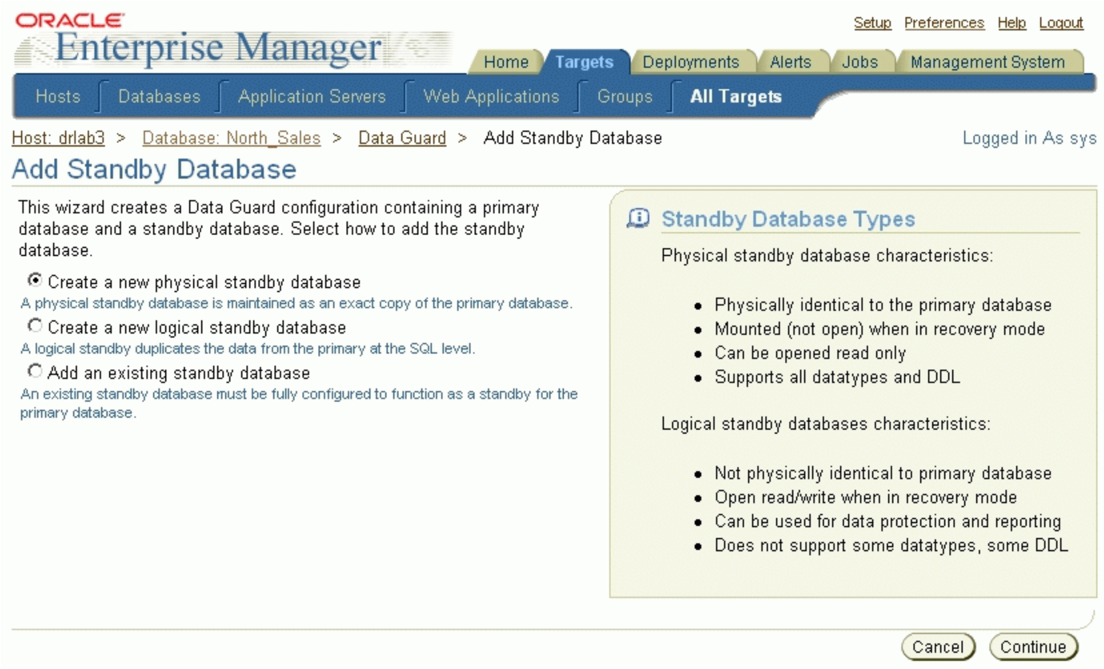
- Add an existing (physical or logical) standby database (single-instance or RAC)

You will need to connect to the primary database using SYSDBA credentials, if you haven't already.

> **See Also:** *Oracle Data Guard Concepts and Administration* for the steps about how to create a RAC standby database

Figure 5–4 show the introductory page of the create configuration process.

Figure 5–4 Add Standby Database Wizard - Introductory Page



Add Standby Database

This wizard creates a Data Guard configuration containing a primary database and a standby database. Select how to add the standby database.

● Create a new physical standby database
A physical standby database is maintained as an exact copy of the primary database.

○ Create a new logical standby database
A logical standby duplicates the data from the primary at the SQL level.

○ Add an existing standby database
An existing standby database must be fully configured to function as a standby for the primary database.

Standby Database Types

Physical standby database characteristics:

- Physically identical to the primary database
- Mounted (not open) when in recovery mode
- Can be opened read only
- Supports all datatypes and DDL

Logical standby databases characteristics:

- Not physically identical to primary database
- Open read/write when in recovery mode
- Can be used for data protection and reporting
- Does not support some datatypes, some DDL

****

The Add Standby Database wizard takes you through the following steps:

1. Determine the backup type.
2. Set up the backup options.

3. Select the Oracle home in which to create the standby database.
4. Set up the location for standby database files.
5. Provide standby database configuration parameters.
6. Review the information before clicking Finish.

The following steps create a broker configuration and create a new physical standby database. It shows how the wizard takes you through additional steps to select the Oracle home for the database and to copy datafiles to the standby database.

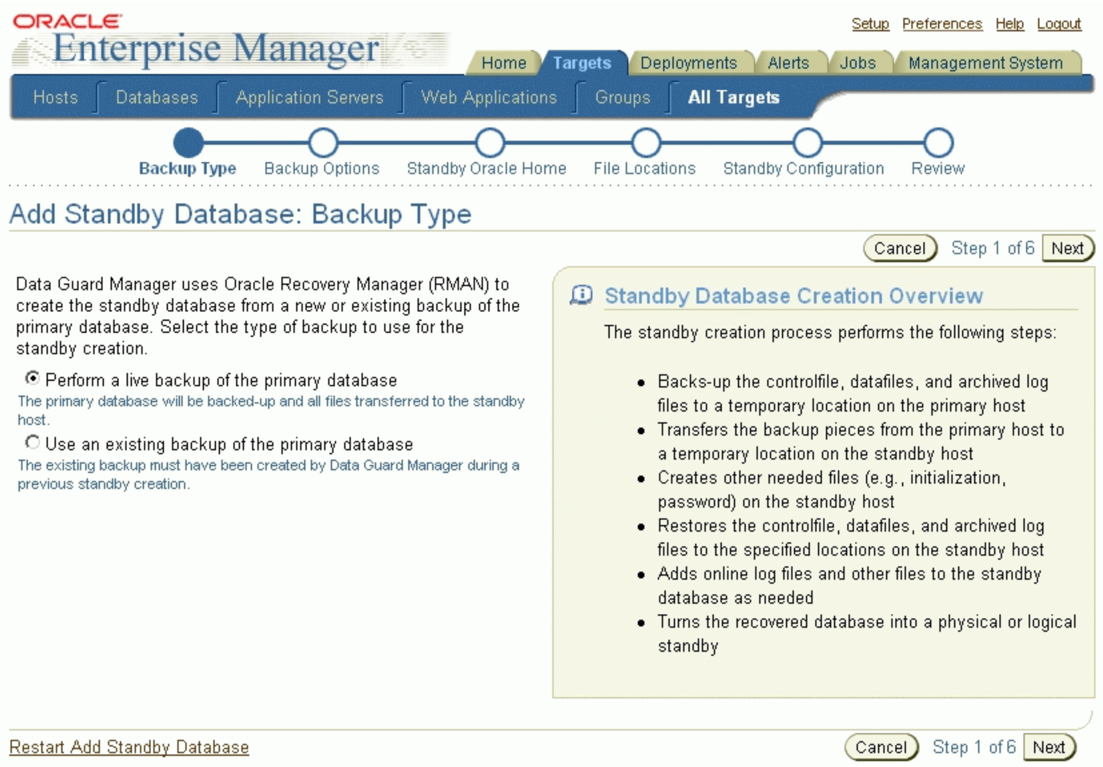
Step 1 Determine the backup type.

The Data Guard GUI uses Oracle Recovery Manager (RMAN) to create a single-instance standby database from a new or existing backup of the primary database. You can select one of two backup operations to use for the standby database creation:

- Perform a live backup of the primary database
- Use an existing backup of the primary database

Figure 5–5 shows Step 1 of 6 of the configuration process if you are adding a physical standby database.

Figure 5–5 Add Standby Database Wizard - Backup Type (Physical Standby Database)


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Backup Type
Backup Options
Standby Oracle Home
File Locations
Standby Configuration
Review
Cancel Step 1 of 6 Next

Add Standby Database: Backup Type

Data Guard Manager uses Oracle Recovery Manager (RMAN) to create the standby database from a new or existing backup of the primary database. Select the type of backup to use for the standby creation.

● Perform a live backup of the primary database
The primary database will be backed-up and all files transferred to the standby host.

○ Use an existing backup of the primary database
The existing backup must have been created by Data Guard Manager during a previous standby creation.

Standby Database Creation Overview

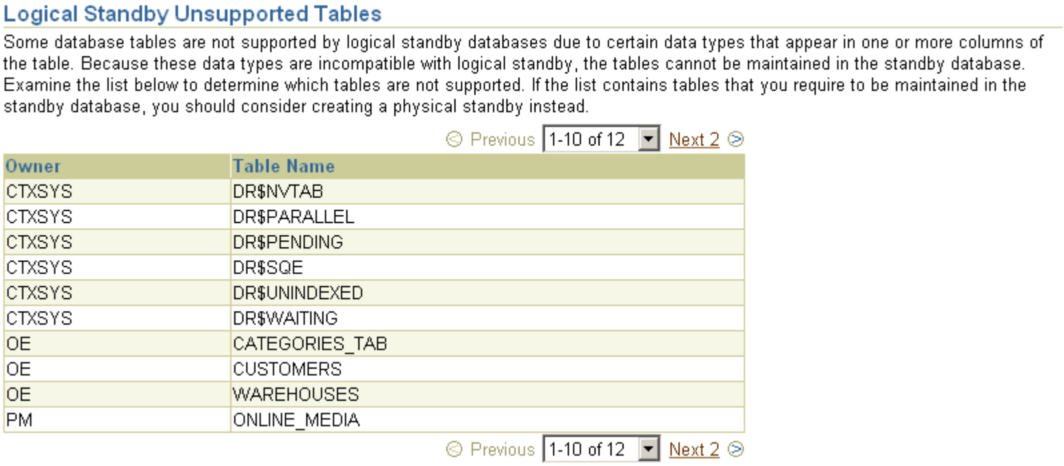
The standby creation process performs the following steps:

- Backs-up the controlfile, datafiles, and archived log files to a temporary location on the primary host
- Transfers the backup pieces from the primary host to a temporary location on the standby host
- Creates other needed files (e.g., initialization, password) on the standby host
- Restores the controlfile, datafiles, and archived log files to the specified locations on the standby host
- Adds online log files and other files to the standby database as needed
- Turns the recovered database into a physical or logical standby

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

Figure 5–6 shows additional screen content that appears if you are adding a logical standby database.

Figure 5–6 Add Standby Database Wizard - Backup Type (Logical Standby Database)



Logical Standby Unsupported Tables

Some database tables are not supported by logical standby databases due to certain data types that appear in one or more columns of the table. Because these data types are incompatible with logical standby, the tables cannot be maintained in the standby database. Examine the list below to determine which tables are not supported. If the list contains tables that you require to be maintained in the standby database, you should consider creating a physical standby instead.

Owner	Table Name
CTXSYS	DR$NVTAB
CTXSYS	DR$PARALLEL
CTXSYS	DR$PENDING
CTXSYS	DR$SQE
CTXSYS	DR$UNINDEXED
CTXSYS	DR$WAITING
OE	CATEGORIES_TAB
OE	CUSTOMERS
OE	WAREHOUSES
PM	ONLINE_MEDIA

Step 2 Set up the backup options.

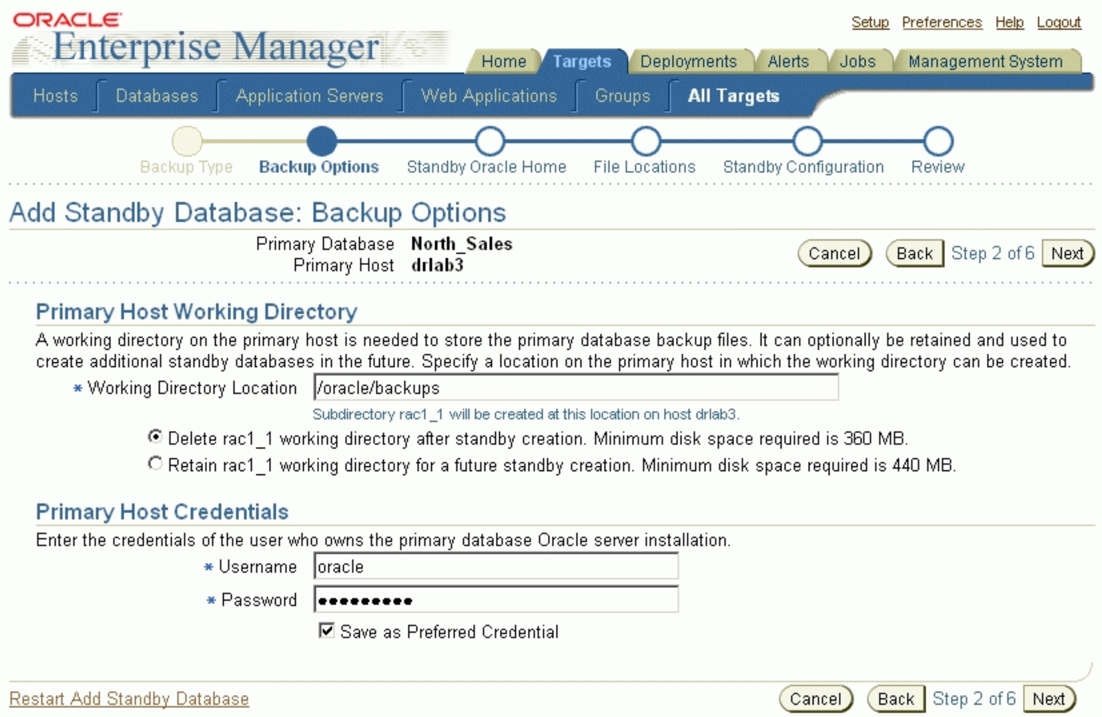
A working directory is needed to store the primary database backup files. It can optionally be retained and used to create additional standby databases in the future. Specify a location on the primary host in which the working directory can be created.

Primary host credentials are required for this step. Enter the credentials of the owner of the primary database Oracle server installation. These credentials can be saved by checking the box marked **Save as Preferred Credential**.

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

Figure 5–7 shows Step 2 of 6 of the configuration process.

Figure 5–7 Add Standby Database Wizard - Backup Options


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Backup Type
Backup Options
Standby Oracle Home
File Locations
Standby Configuration
Review

Add Standby Database: Backup Options

Primary Database **North_Sales**
Primary Host **drlab3**

Cancel Back Step 2 of 6 Next

Primary Host Working Directory

A working directory on the primary host is needed to store the primary database backup files. It can optionally be retained and used to create additional standby databases in the future. Specify a location on the primary host in which the working directory can be created.

* Working Directory Location /oracle/backups

Subdirectory rac1_1 will be created at this location on host drlab3.

● Delete rac1_1 working directory after standby creation. Minimum disk space required is 360 MB.
○ Retain rac1_1 working directory for a future standby creation. Minimum disk space required is 440 MB.

Primary Host Credentials

Enter the credentials of the user who owns the primary database Oracle server installation.

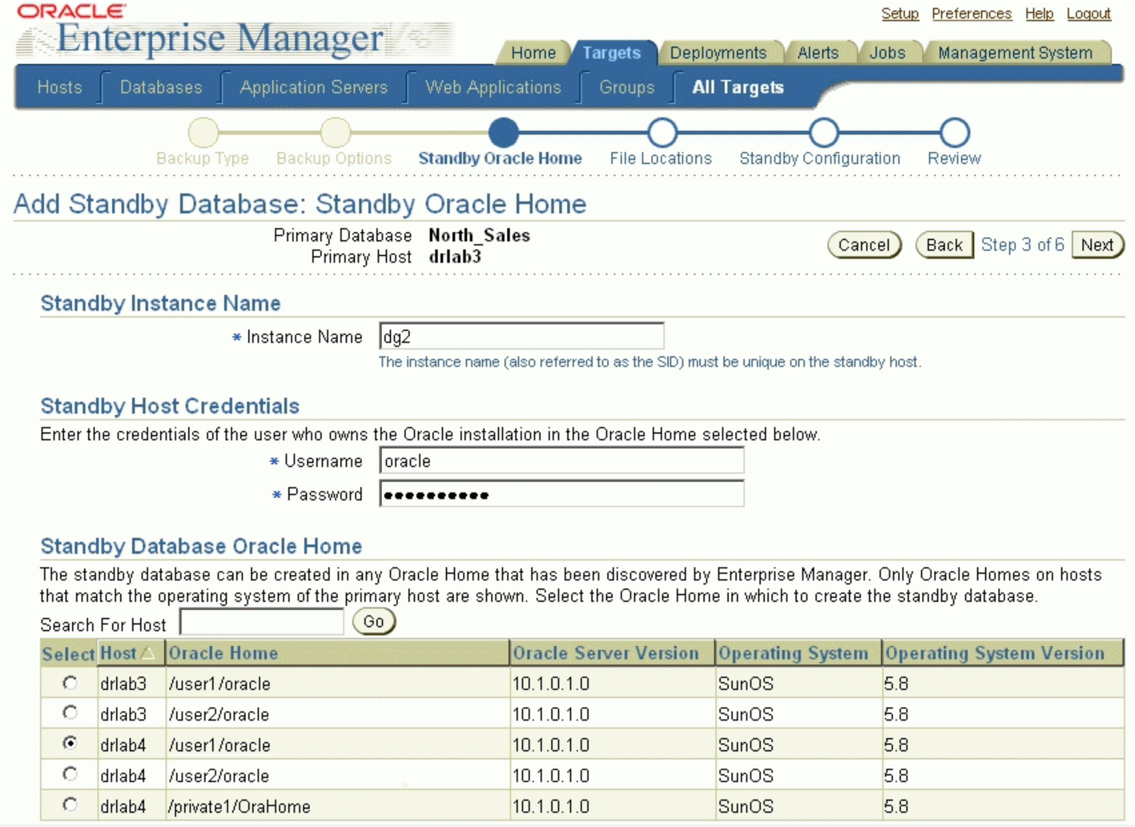
* Username oracle
* Password ••••••••••
☑ Save as Preferred Credential

Step 3 Select the Oracle home in which to create the standby database.

The standby database can be created in any Oracle home that was discovered by Oracle Enterprise Manager. Only Oracle homes on hosts that match the operating system of the primary host are shown. You must select a discovered Oracle home and provide a unique instance name for the standby database. Standby host credentials are required to continue.

Figure 5–8 shows Step 3 of 6 of the configuration process.

Figure 5–8 Add Standby Database Wizard - Standby Oracle Home



Add Standby Database: Standby Oracle Home

Primary Database North_Sales
Primary Host drlab3

Standby Instance Name

* Instance Name: dg2

The instance name (also referred to as the SID) must be unique on the standby host.

Standby Host Credentials

Enter the credentials of the user who owns the Oracle installation in the Oracle Home selected below.

* Username: oracle

* Password: •••••••••••

Standby Database Oracle Home

The standby database can be created in any Oracle Home that has been discovered by Enterprise Manager. Only Oracle Homes on hosts that match the operating system of the primary host are shown. Select the Oracle Home in which to create the standby database.

Search For Host

Select	Host	Oracle Home	Oracle Server Version	Operating System	Operating System Version
○	drlab3	/user1/oracle	10.1.0.1.0	SunOS	5.8
○	drlab3	/user2/oracle	10.1.0.1.0	SunOS	5.8
●	drlab4	/user1/oracle	10.1.0.1.0	SunOS	5.8
○	drlab4	/user2/oracle	10.1.0.1.0	SunOS	5.8
○	drlab4	/private1/OraHome	10.1.0.1.0	SunOS	5.8

Step 4 Set up the location for standby database files.

Part of the create broker configuration process involves making the datafiles for the primary database available to the standby host. You have the option of customizing the location for the standby database files. Standby host credentials are required to continue. The following list describes your options:

- Specify the backup file access method

  Choose the method by which you want to make the primary database backup files accessible to the standby host. The two options are:

  – Transfer files from the primary host working directory to a standby host working directory

  - Directly access the primary host working directory location from the standby host using a network path name

- Specify the standby database file location

  Choose the locations for the standby database files. You have two options:

  - Convert to Oracle OFA (Optimal Flexible Architecture)
  - Keep file names and locations the same as the primary database

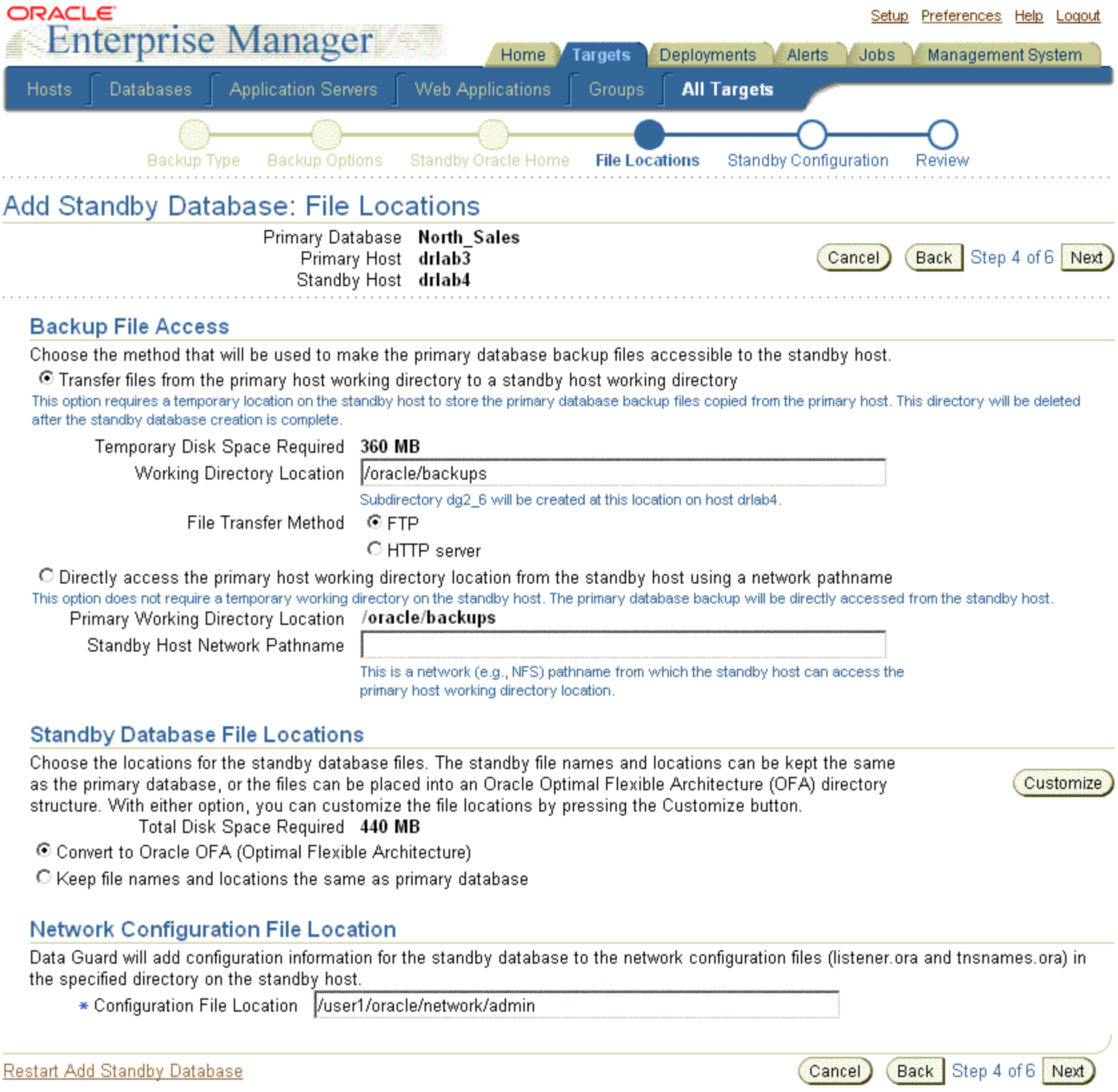
- Specify the network configuration file location

  Data Guard will add configuration information for the standby database to the network configuration files (listener.ora and tnsnames.ora) in the specified directory on the standby host.

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

Figure 5–9 shows Step 4 of 6 of the configuration process.

Figure 5–9 Add Standby Database Wizard - File Locations



Add Standby Database: File Locations

Primary Database **North_Sales**
Primary Host **drlab3**
Standby Host **drlab4**

Backup File Access

Choose the method that will be used to make the primary database backup files accessible to the standby host.

● Transfer files from the primary host working directory to a standby host working directory

This option requires a temporary location on the standby host to store the primary database backup files copied from the primary host. This directory will be deleted after the standby database creation is complete.

Temporary Disk Space Required	**360 MB**
Working Directory Location	/oracle/backups Subdirectory dg2_6 will be created at this location on host drlab4.
File Transfer Method	● FTP ○ HTTP server

○ Directly access the primary host working directory location from the standby host using a network pathname

This option does not require a temporary working directory on the standby host. The primary database backup will be directly accessed from the standby host.

Primary Working Directory Location	**/oracle/backups**
Standby Host Network Pathname	This is a network (e.g., NFS) pathname from which the standby host can access the primary host working directory location.

Standby Database File Locations

Choose the locations for the standby database files. The standby file names and locations can be kept the same as the primary database, or the files can be placed into an Oracle Optimal Flexible Architecture (OFA) directory structure. With either option, you can customize the file locations by pressing the Customize button.

Total Disk Space Required **440 MB**

● Convert to Oracle OFA (Optimal Flexible Architecture)

○ Keep file names and locations the same as primary database

Network Configuration File Location

Data Guard will add configuration information for the standby database to the network configuration files (listener.ora and tnsnames.ora) in the specified directory on the standby host.

* Configuration File Location /user1/oracle/network/admin

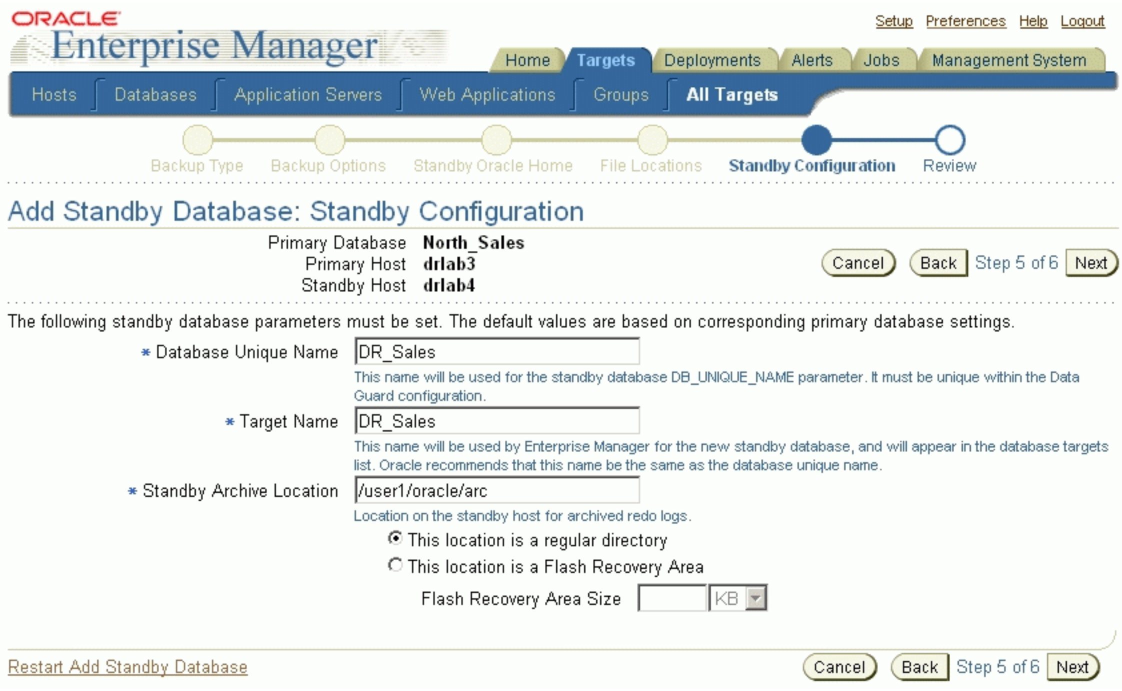
Step 5 Provide standby database configuration parameters.

Standby database configuration parameters must be set. These parameters include the instance name (SID), database unique name, target name, and standby archive location. The standby archive location can be a regular directory or a flash recovery area. The default values are based on corresponding primary database settings.

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

Figure 5–10 shows Step 5 of 6 of the configuration process.

Figure 5–10 Add Standby Database Wizard - Standby Configuration



ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Backup Type Backup Options Standby Oracle Home File Locations Standby Configuration Review

Add Standby Database: Standby Configuration

Primary Database **North_Sales**
Primary Host **drlab3**
Standby Host **drlab4**

Cancel Back Step 5 of 6 Next

The following standby database parameters must be set. The default values are based on corresponding primary database settings.

* Database Unique Name: DR_Sales
This name will be used for the standby database DB_UNIQUE_NAME parameter. It must be unique within the Data Guard configuration.

* Target Name: DR_Sales
This name will be used by Enterprise Manager for the new standby database, and will appear in the database targets list. Oracle recommends that this name be the same as the database unique name.

* Standby Archive Location: /user1/oracle/arc
Location on the standby host for archived redo logs.

● This location is a regular directory
○ This location is a Flash Recovery Area

Flash Recovery Area Size: KB

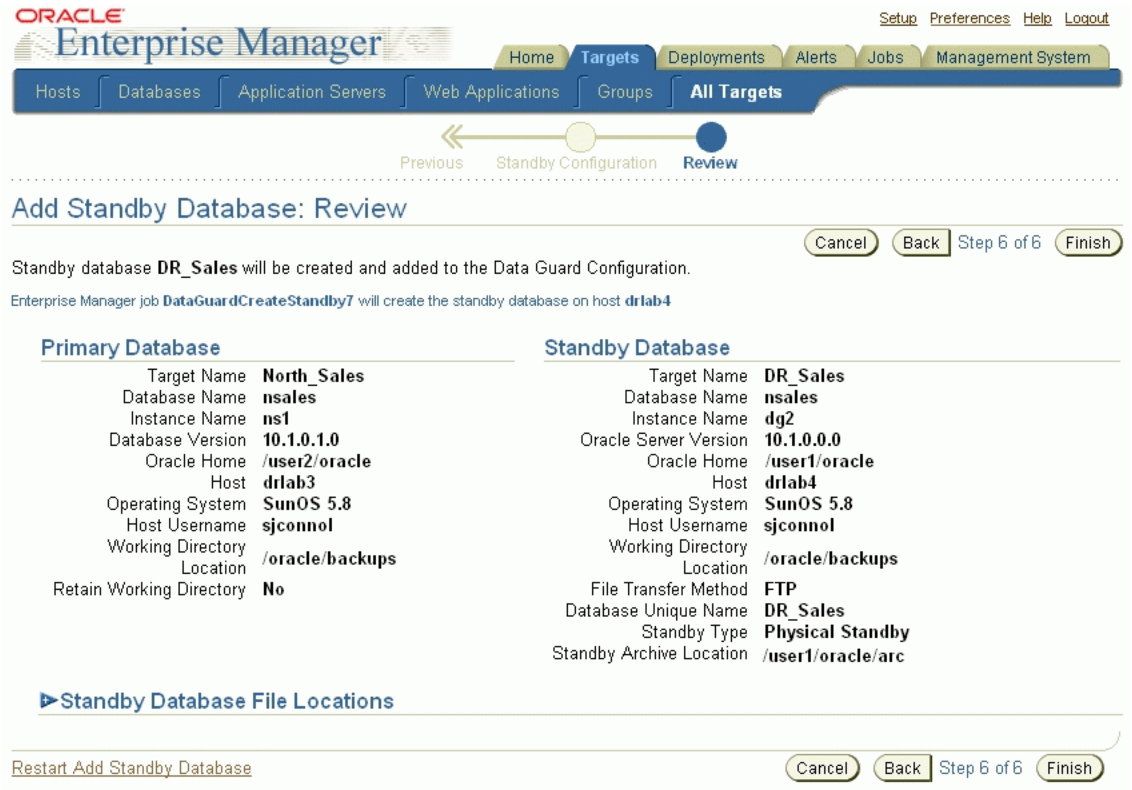
Step 6 Review the information before clicking Finish.

The Add Standby Database wizard allows one last review of the data you input for the configuration and standby database. Click **Finish** when you are certain all of the information is correct.

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

Figure 5–11 shows Step 6 of 6 of the configuration process.

Figure 5–11 Add Standby Database Wizard - Review



Add Standby Database: Review

Standby database **DR_Sales** will be created and added to the Data Guard Configuration.

Enterprise Manager job **DataGuardCreateStandby7** will create the standby database on host **drlab4**

Primary Database

Target Name	North_Sales
Database Name	nsales
Instance Name	ns1
Database Version	10.1.0.1.0
Oracle Home	/user2/oracle
Host	drlab3
Operating System	SunOS 5.8
Host Username	sjconnol
Working Directory Location	/oracle/backups
Retain Working Directory	No

Standby Database

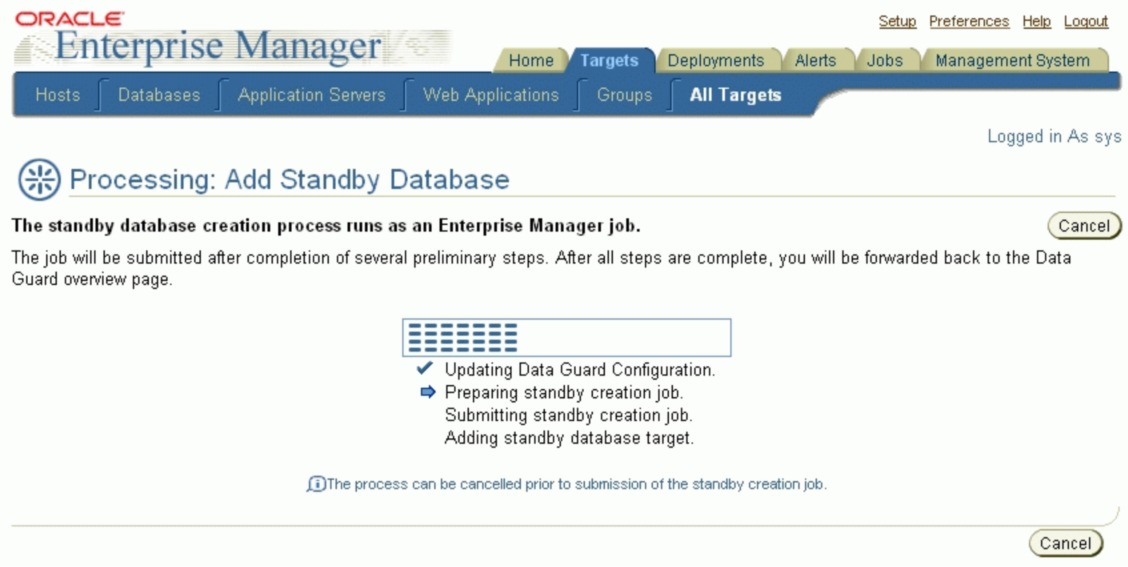
Target Name	DR_Sales
Database Name	nsales
Instance Name	dg2
Oracle Server Version	10.1.0.0.0
Oracle Home	/user1/oracle
Host	drlab4
Operating System	SunOS 5.8
Host Username	sjconnol
Working Directory Location	/oracle/backups
File Transfer Method	FTP
Database Unique Name	DR_Sales
Standby Type	Physical Standby
Standby Archive Location	/user1/oracle/arc

▶Standby Database File Locations

By clicking **Standby Database File Locations**, you can see additional information about all the standby database file locations.

Once you click **Finish**, the standby database creation process runs as an Oracle Enterprise Manager job. You can cancel the standby creation at any point before the job submission. Figure 5–12 shows the standby database creation process.

Figure 5–12 Add Standby Database Wizard - Processing



Processing: Add Standby Database

The standby database creation process runs as an Enterprise Manager job.

The job will be submitted after completion of several preliminary steps. After all steps are complete, you will be forwarded back to the Data Guard overview page.

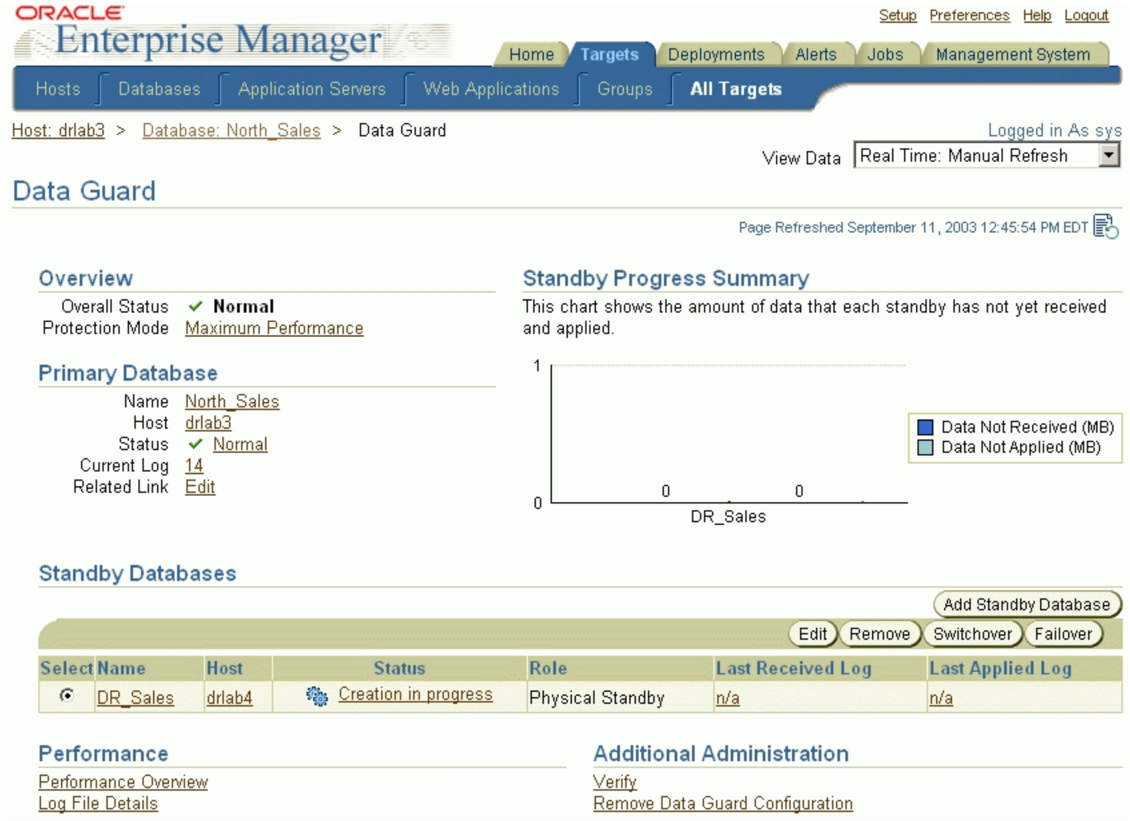
✓ Updating Data Guard Configuration.
⇨ Preparing standby creation job.
Submitting standby creation job.
Adding standby database target.

The process can be cancelled prior to submission of the standby creation job.

After the job is submitted, you will be returned to the Data Guard Overview page. In the Status column of the Standby Databases table, you will see *Creation in progress* listed. If you click that link, you can monitor the progress of the standby database creation. Figure 5–13 shows the Data Guard Overview page with this link.

To add additional standby databases after the initial creation of the configuration, click **Add Standby Database** to run the Add Standby Database wizard again.

Figure 5–13 Data Guard Overview Page - Creation in Progress


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
1
0
0
0
DR_Sales
Add Standby Database
Edit Remove Switchover Failover

Host: drlab3 > Database: North_Sales > Data Guard

Logged in As sys

View Data: Real Time: Manual Refresh

Data Guard

Page Refreshed September 11, 2003 12:45:54 PM EDT

Overview

Overall Status ✓ Normal
Protection Mode Maximum Performance

Primary Database

Name North_Sales
Host drlab3
Status ✓ Normal
Current Log 14
Related Link Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

🟦 Data Not Received (MB)
▨ Data Not Applied (MB)

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	DR_Sales	drlab4	Creation in progress	Physical Standby	n/a	n/a

Performance

Performance Overview
Log File Details

Additional Administration

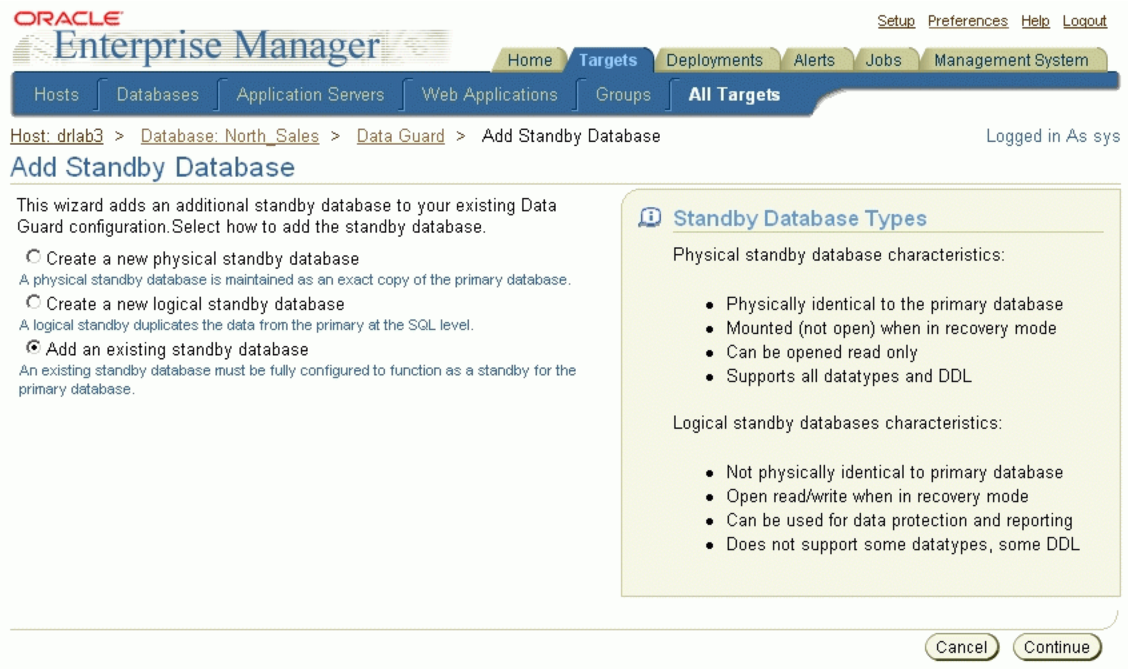
Verify
Remove Data Guard Configuration

5.3 Scenario 3: Adding an Existing RAC Standby Database

Although the Add Standby Database wizard will not create a RAC standby database, you can add an existing RAC standby database into a Data Guard configuration. Click **Add Standby Database** to run the Add Standby Database wizard and select **Add an existing standby database**.

Figure 5–14 shows the Add Standby Database introductory page.

Figure 5–14 Adding an Existing RAC Standby Database to the Data Guard Configuration


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Host: drlab3 > Database: North_Sales > Data Guard > Add Standby Database
Logged in As sys

Add Standby Database

This wizard adds an additional standby database to your existing Data Guard configuration. Select how to add the standby database.

○ Create a new physical standby database
A physical standby database is maintained as an exact copy of the primary database.

○ Create a new logical standby database
A logical standby duplicates the data from the primary at the SQL level.

● Add an existing standby database
An existing standby database must be fully configured to function as a standby for the primary database.

Standby Database Types

Physical standby database characteristics:

- Physically identical to the primary database
- Mounted (not open) when in recovery mode
- Can be opened read only
- Supports all datatypes and DDL

Logical standby databases characteristics:

- Not physically identical to primary database
- Open read/write when in recovery mode
- Can be used for data protection and reporting
- Does not support some datatypes, some DDL

The Add Standby Database wizard takes you through the following steps:

1. Select an existing standby database.
2. Set the standby archive location setting.
3. Review the configuration settings.

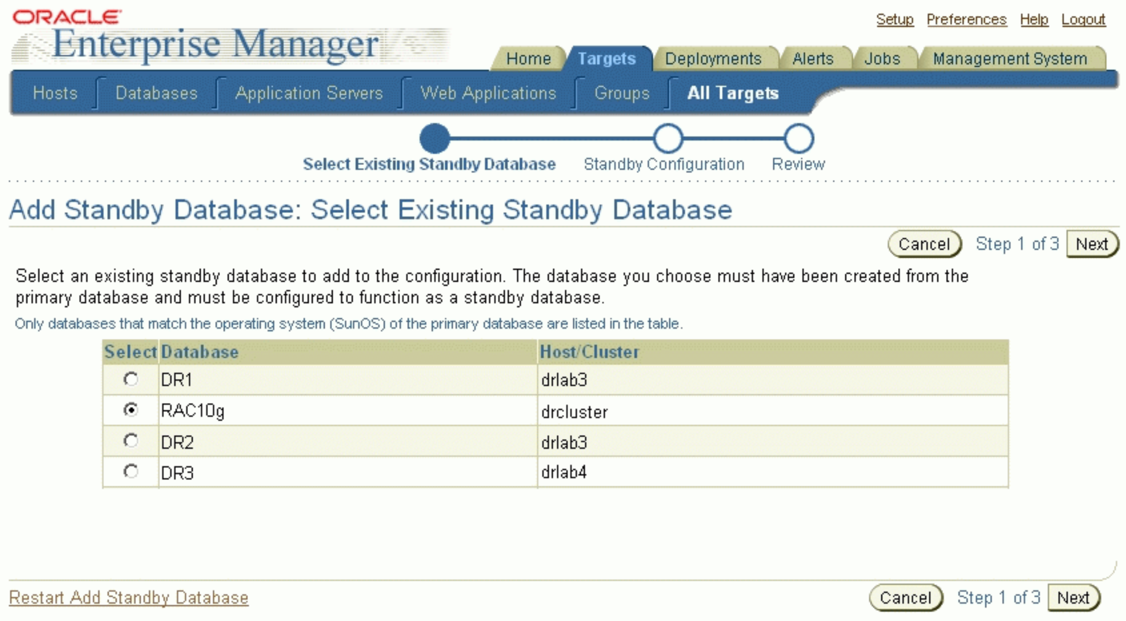
Step 1 Select an existing standby database.

In this step, select the RAC standby database you want to add to the configuration. All discovered databases in your environment (both RAC and non-RAC databases) will be shown in the list. In the example shown in Figure 5–15 (Step 1 of 3), one of the databases in the list (RAC10*g*) is a RAC standby database.

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

If you wish to continue, click **Next**.

Figure 5–15 Select an Existing Standby Database



Add Standby Database: Select Existing Standby Database

Select an existing standby database to add to the configuration. The database you choose must have been created from the primary database and must be configured to function as a standby database.

Only databases that match the operating system (SunOS) of the primary database are listed in the table.

Select	Database	Host/Cluster
○	DR1	drlab3
●	RAC10g	drcluster
○	DR2	drlab3
○	DR3	drlab4

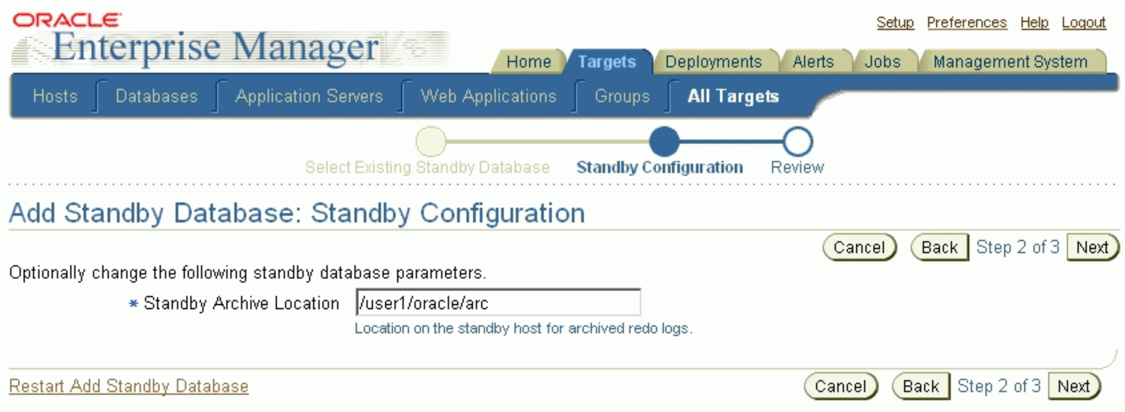
Step 2 Set the standby archive location setting.

You can optionally change the Standby Archive Location setting of the existing standby cluster database, as shown in Figure 5–16 (Step 2 of 3).

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

If you wish to continue, click **Next**.

Figure 5–16 Add Standby Database: Standby Configuration



Add Standby Database: Standby Configuration

Optionally change the following standby database parameters.

* Standby Archive Location /user1/oracle/arc

Location on the standby host for archived redo logs.

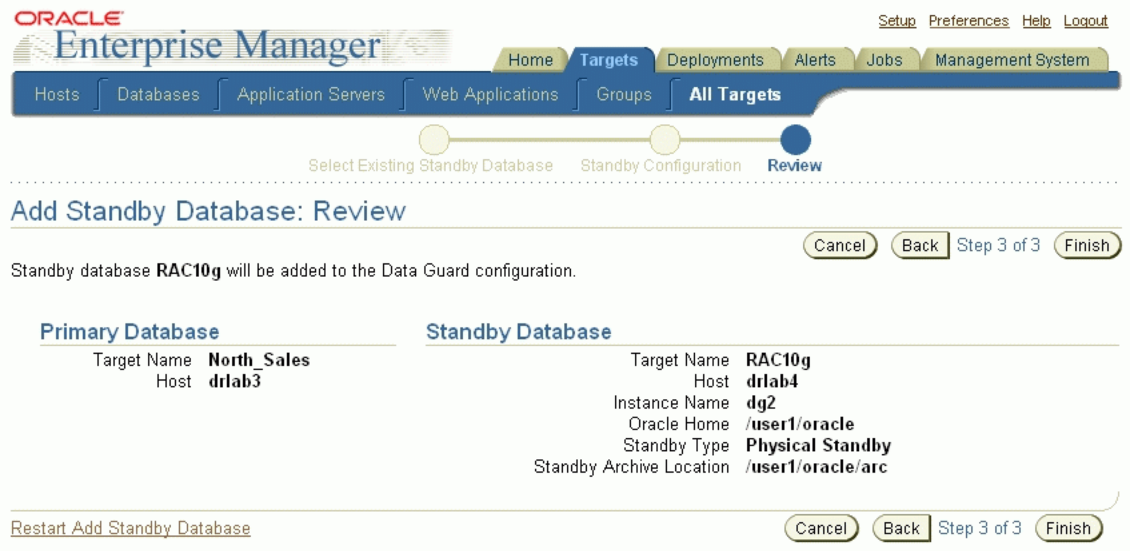
Step 3 Review the configuration settings.

Review the data for the configuration and standby database, as shown in Figure 5–17 (Step 3 of 3).

You can click **Restart Add Standby Database** to terminate the current process and begin again at the introductory page of the Add Standby Database wizard.

If you wish to complete the operation, click **Finish**.

Figure 5–17 Add Standby Database: Review


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Select Existing Standby Database
Standby Configuration
Review
Cancel
Back
Step 3 of 3
Finish
Restart Add Standby Database

Add Standby Database: Review

Standby database **RAC10g** will be added to the Data Guard configuration.

Primary Database

Target Name	**North_Sales**
Host	**drlab3**

Standby Database

Target Name	**RAC10g**
Host	**drlab4**
Instance Name	**dg2**
Oracle Home	**/user1/oracle**
Standby Type	**Physical Standby**
Standby Archive Location	**/user1/oracle/arc**

5.4 Scenario 4: Performing Routine Maintenance

The Data Guard GUI can help simplify some of the routine maintenance tasks you must perform in the configuration. The following examples, provided in the following sections, show:

- Changing the State of a Database
- Changing the Properties of a Database
- Changing the Database Protection Mode

5.4.1 Changing the State of a Database

This section describes how to set the standby database to Log Apply Off.

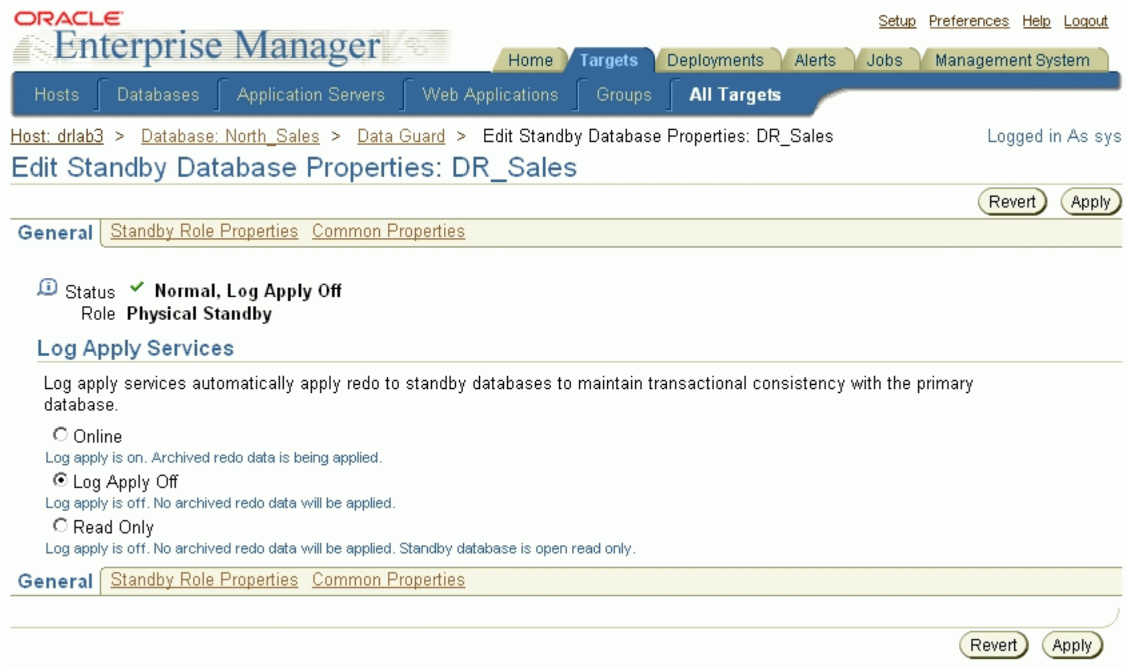
To change the state of the standby database to Log Apply Off, follow these steps:

1. From the Data Guard Overview page, select the standby database you want to change to the Log Apply Off state.
2. Click **Edit** to go to the Edit Properties page.

3. Select **Log Apply Off**.
4. Click **Apply**.
5. When the process completes, a message indicating success is returned.

Figure 5–18 shows the Edit Properties page from where you can change the state of a database.

Figure 5–18 Changing the State or Properties of a Database


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Host: drlab3 > Database: North_Sales > Data Guard > Edit Standby Database Properties: DR_Sales
Logged in As sys
Revert Apply
General Standby Role Properties Common Properties

Edit Standby Database Properties: DR_Sales

Status ✓ **Normal, Log Apply Off**
Role **Physical Standby**

Log Apply Services

Log apply services automatically apply redo to standby databases to maintain transactional consistency with the primary database.

○ Online
Log apply is on. Archived redo data is being applied.

● Log Apply Off
Log apply is off. No archived redo data will be applied.

○ Read Only
Log apply is off. No archived redo data will be applied. Standby database is open read only.

****

When you change the state of the standby database to Log Apply Off, it stops log apply services from applying the archived redo log files to the standby database. However, log transport services are unaffected.

After completing your maintenance tasks, you can follow the same sequence of steps to bring the database online again.

5.4.2 Changing the Properties of a Database

The Data Guard GUI divides the database properties into primary, standby, and common properties.

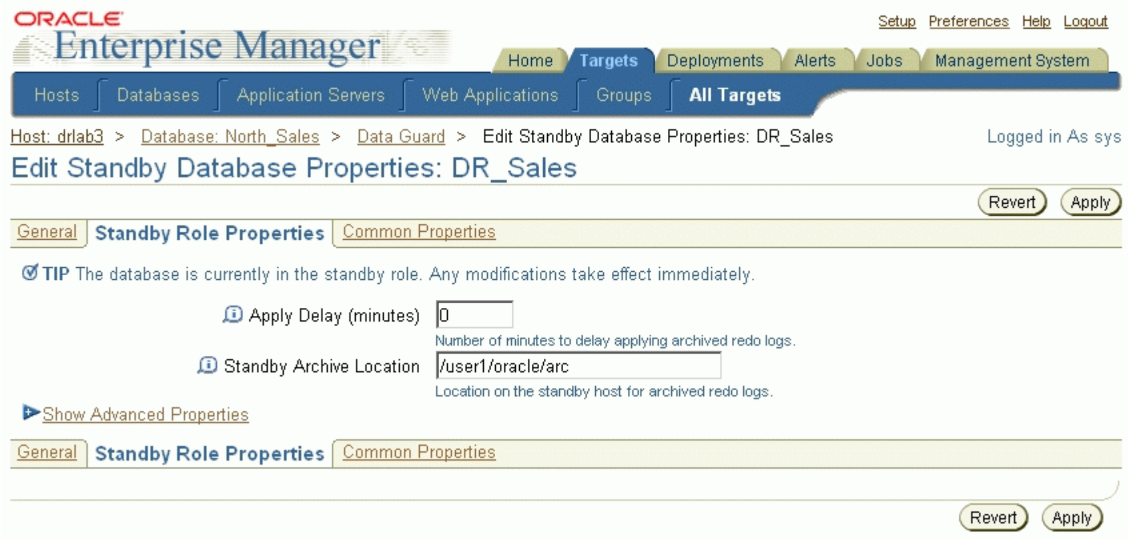
To view or change properties:

1. Click either **Edit** from the Primary Database section or click **Edit** from the Standby Databases section of the Data Guard Overview page.
2. Click one of the following:
   - Standby Role Properties
   - Common Properties
3. Make the appropriate change and click **Apply**.
4. When the process completes, a message indicating success is returned.

Figure 5–18 shows the Edit Properties page from where you can alter a property of a database.

Figure 5–19 shows properties specific to the standby database.

Figure 5–19 Standby Role Properties


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Logged in As sys
Revert Apply
General Standby Role Properties Common Properties
Show Advanced Properties
General Standby Role Properties Common Properties
Revert Apply

Host: drlab3 > Database: North_Sales > Data Guard > Edit Standby Database Properties: DR_Sales

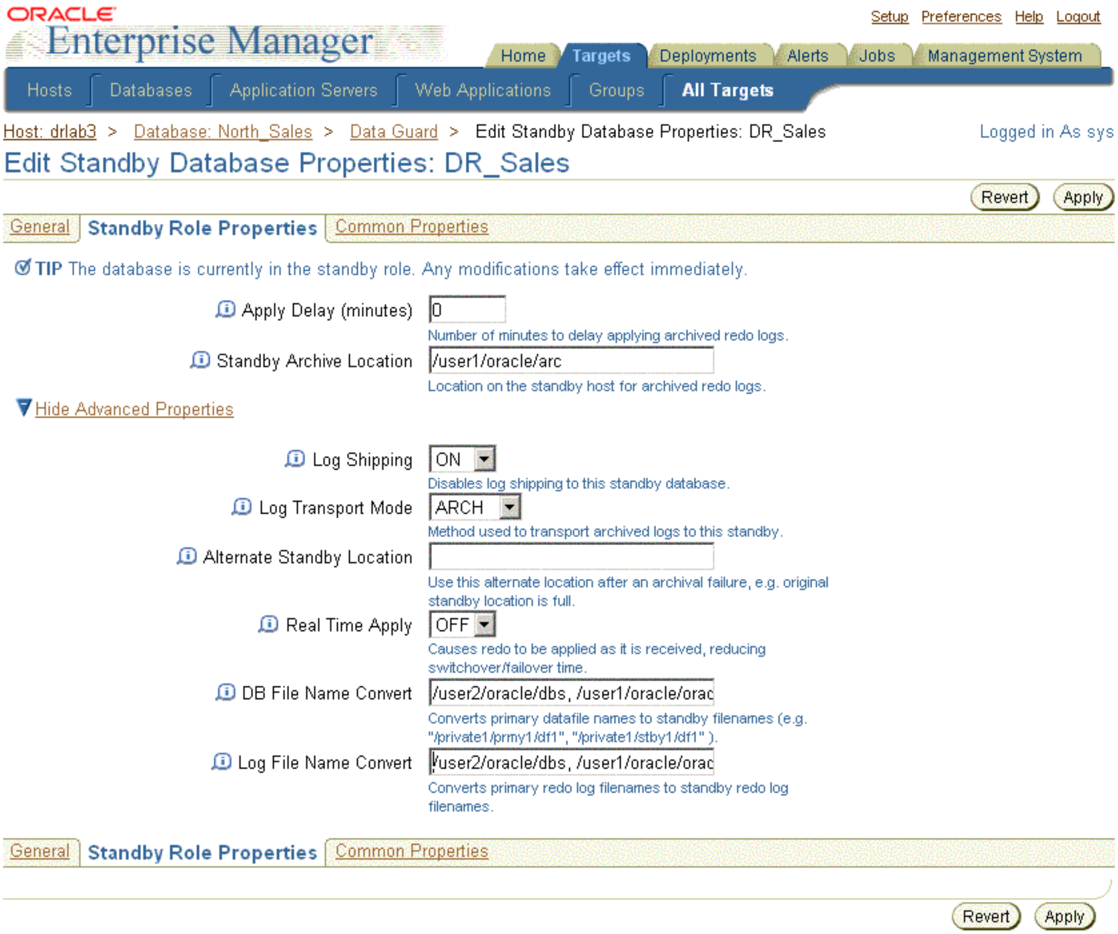
Edit Standby Database Properties: DR_Sales

TIP The database is currently in the standby role. Any modifications take effect immediately.

Apply Delay (minutes)	0 Number of minutes to delay applying archived redo logs.
Standby Archive Location	/user1/oracle/arc Location on the standby host for archived redo logs.

By clicking **Show Advanced Properties**, you can view additional properties specific to the standby database, as shown in Figure 5–20.

Figure 5–20 Standby Advanced Properties



Host: drlab3 > Database: North_Sales > Data Guard > Edit Standby Database Properties: DR_Sales

Logged in As sys

Edit Standby Database Properties: DR_Sales

TIP The database is currently in the standby role. Any modifications take effect immediately.

Apply Delay (minutes): 0
Number of minutes to delay applying archived redo logs.

Standby Archive Location: /user1/oracle/arc
Location on the standby host for archived redo logs.

Log Shipping: ON
Disables log shipping to this standby database.

Log Transport Mode: ARCH
Method used to transport archived logs to this standby.

Alternate Standby Location:
Use this alternate location after an archival failure, e.g. original standby location is full.

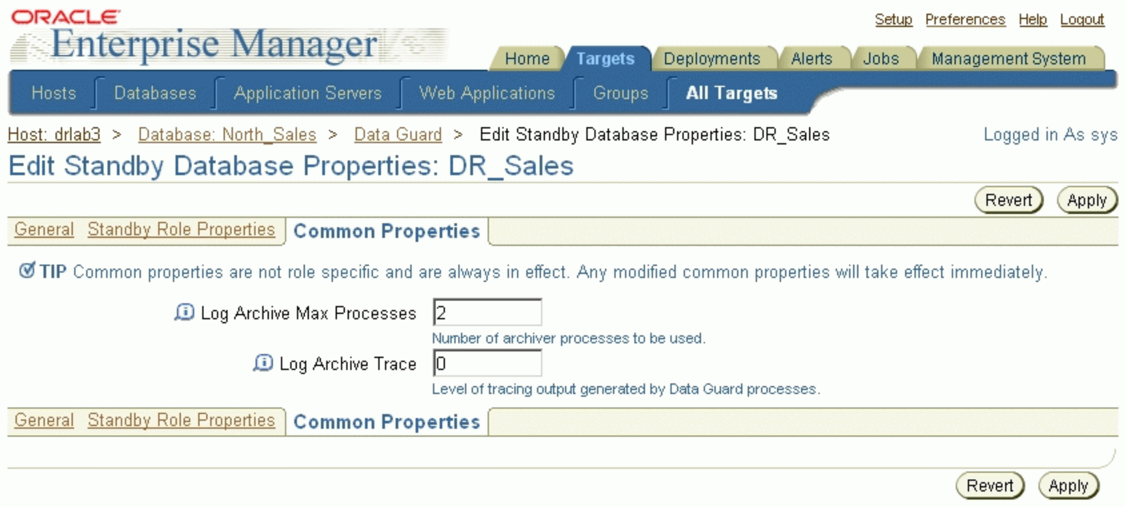
Real Time Apply: OFF
Causes redo to be applied as it is received, reducing switchover/failover time.

DB File Name Convert: /user2/oracle/dbs, /user1/oracle/orac
Converts primary datafile names to standby filenames (e.g. "/private1/prmy1/df1", "/private1/stby1/df1").

Log File Name Convert: /user2/oracle/dbs, /user1/oracle/orac
Converts primary redo log filenames to standby redo log filenames.

Figure 5–21 shows properties common to both the primary database and the standby database.

Figure 5–21 Common Properties



ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Host: drlab3 > Database: North_Sales > Data Guard > Edit Standby Database Properties: DR_Sales
Logged in As sys
Revert Apply
General Standby Role Properties Common Properties
Revert Apply

Edit Standby Database Properties: DR_Sales

TIP Common properties are not role specific and are always in effect. Any modified common properties will take effect immediately.

Log Archive Max Processes: 2
Number of archiver processes to be used.

Log Archive Trace: 0
Level of tracing output generated by Data Guard processes.

5.4.3 Changing the Database Protection Mode

You can change the broker configuration's overall protection mode with the Data Guard GUI.

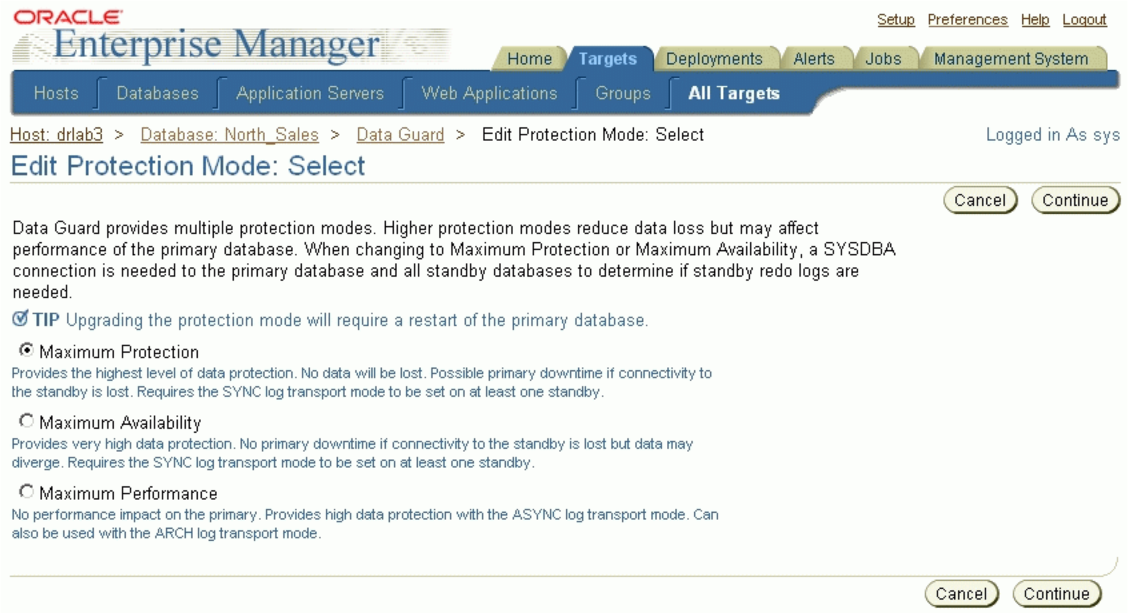
When the configuration was first created it was placed in the maximum performance mode by default. This section describes the process for upgrading to the maximum protection mode. The maximum protection mode guarantees that no data loss will occur if the primary database fails.

To set the maximum protection mode:

1. Click **Maximum Performance** under the Overview section on the Data Guard Overview page.
2. Select **Maximum Protection** and click **Continue**.

   Figure 5–22 shows the Edit Protection Mode page.

Figure 5–22 Edit Protection Mode Page



Edit Protection Mode: Select

Data Guard provides multiple protection modes. Higher protection modes reduce data loss but may affect performance of the primary database. When changing to Maximum Protection or Maximum Availability, a SYSDBA connection is needed to the primary database and all standby databases to determine if standby redo logs are needed.

TIP Upgrading the protection mode will require a restart of the primary database.

● Maximum Protection
Provides the highest level of data protection. No data will be lost. Possible primary downtime if connectivity to the standby is lost. Requires the SYNC log transport mode to be set on at least one standby.

○ Maximum Availability
Provides very high data protection. No primary downtime if connectivity to the standby is lost but data may diverge. Requires the SYNC log transport mode to be set on at least one standby.

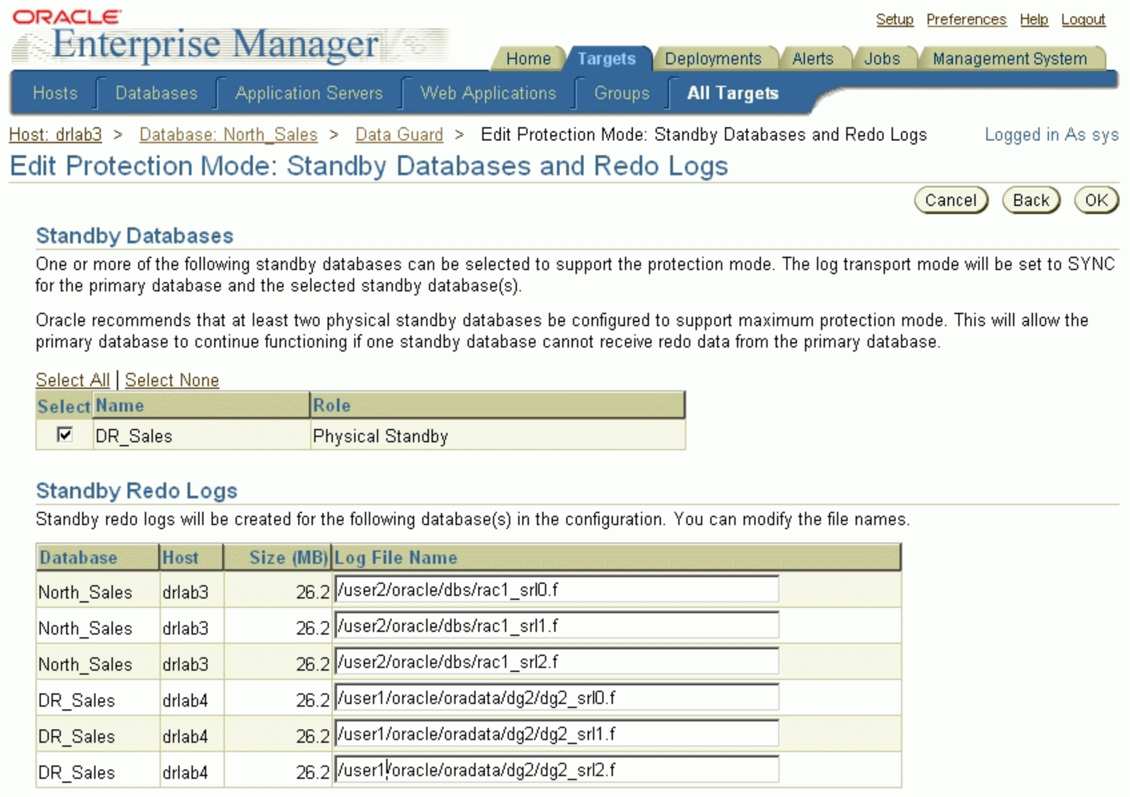
○ Maximum Performance
No performance impact on the primary. Provides high data protection with the ASYNC log transport mode. Can also be used with the ARCH log transport mode.

****

3. Select one or more of the standby databases listed in Figure 5–23 that you want to support the maximum protection mode. The log transport mode (and `LogXptMode` property) will automatically be set to `SYNC` for the primary database and the selected standby databases.

   Data Guard broker automatically determines the correct number and size of standby redo log files needed for all databases in the configuration and adds those log files using the directory locations you specify. Click **OK** to continue.

Figure 5–23 Edit Protection Mode - Standby Databases and Online Redo Log Files



ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Cancel Back OK
Select All | Select None

Host: drlab3 > Database: North_Sales > Data Guard > Edit Protection Mode: Standby Databases and Redo Logs   Logged in As sys

Edit Protection Mode: Standby Databases and Redo Logs

Standby Databases

One or more of the following standby databases can be selected to support the protection mode. The log transport mode will be set to SYNC for the primary database and the selected standby database(s).

Oracle recommends that at least two physical standby databases be configured to support maximum protection mode. This will allow the primary database to continue functioning if one standby database cannot receive redo data from the primary database.

Select	Name	Role
☑	DR_Sales	Physical Standby

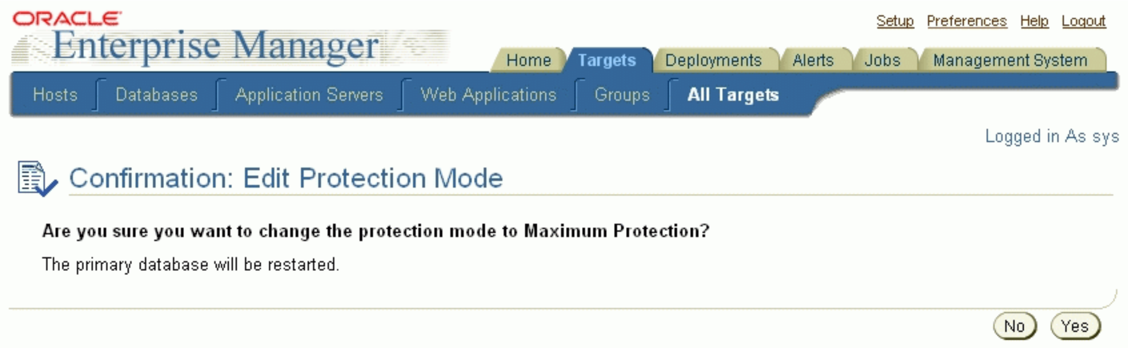
Standby Redo Logs

Standby redo logs will be created for the following database(s) in the configuration. You can modify the file names.

Database	Host	Size (MB)	Log File Name
North_Sales	drlab3	26.2	/user2/oracle/dbs/rac1_srl0.f
North_Sales	drlab3	26.2	/user2/oracle/dbs/rac1_srl1.f
North_Sales	drlab3	26.2	/user2/oracle/dbs/rac1_srl2.f
DR_Sales	drlab4	26.2	/user1/oracle/oradata/dg2/dg2_srl0.f
DR_Sales	drlab4	26.2	/user1/oracle/oradata/dg2/dg2_srl1.f
DR_Sales	drlab4	26.2	/user1/oracle/oradata/dg2/dg2_srl2.f

4. Confirm that you want to change the protection mode. Click **Yes** to continue or **No** to cancel. Figure 5–24 shows this confirmation page.

Figure 5–24 Edit Protection Mode - Confirmation



ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Logged in As sys

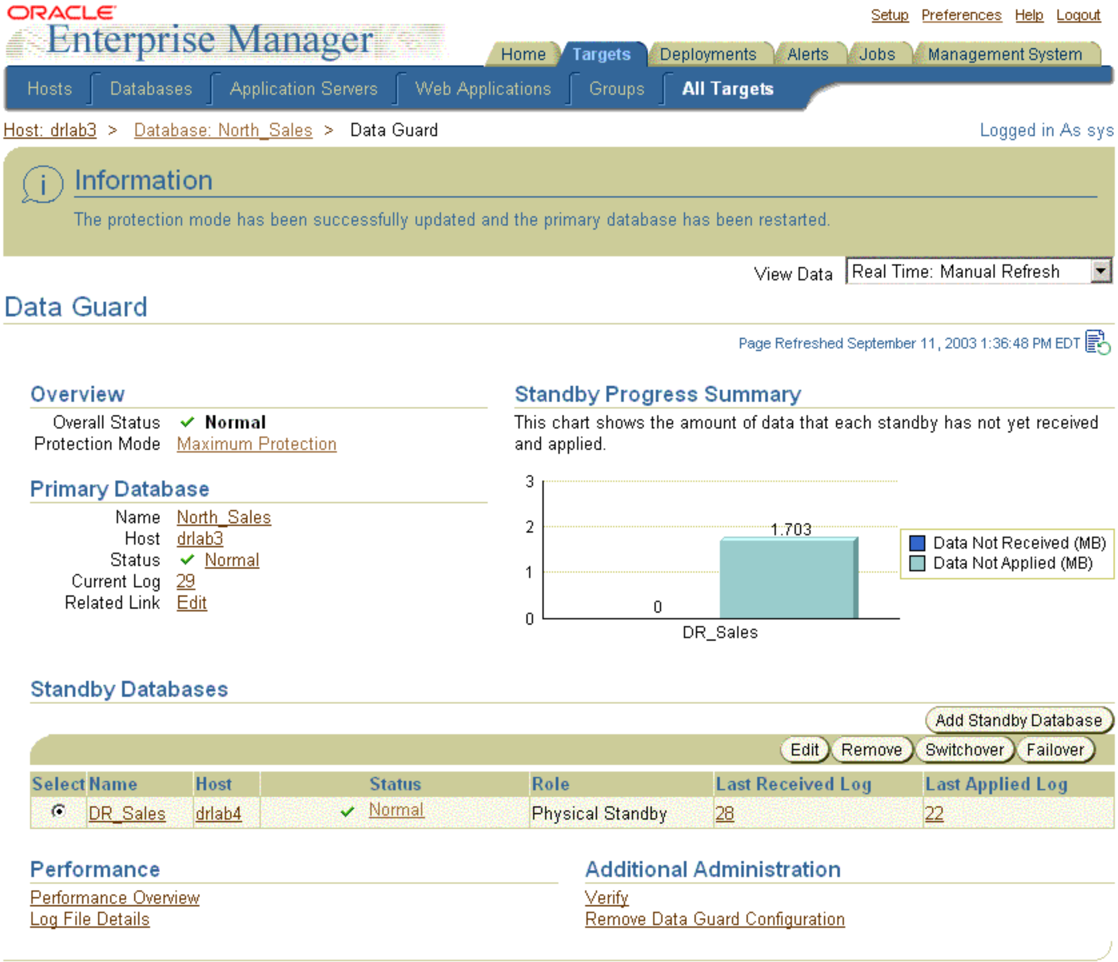
Confirmation: Edit Protection Mode

Are you sure you want to change the protection mode to Maximum Protection?

The primary database will be restarted.

5. Once the protection mode is successfully updated, the Data Guard Overview page is displayed as shown in Figure 5–25.

Figure 5–25 Protection Mode Successfully Changed



Host: drlab3 > Database: North_Sales > Data Guard

Logged in As sys

Information

The protection mode has been successfully updated and the primary database has been restarted.

View Data Real Time: Manual Refresh

Data Guard

Page Refreshed September 11, 2003 1:36:48 PM EDT

Overview

Overall Status ✓ **Normal**
Protection Mode Maximum Protection

Primary Database

Name	North_Sales
Host	drlab3
Status	✓ Normal
Current Log	29
Related Link	Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	DR_Sales	drlab4	✓ Normal	Physical Standby	28	22

Performance

Performance Overview
Log File Details

Additional Administration

Verify
Remove Data Guard Configuration

5.5 Scenario 5: Performing a Switchover Operation

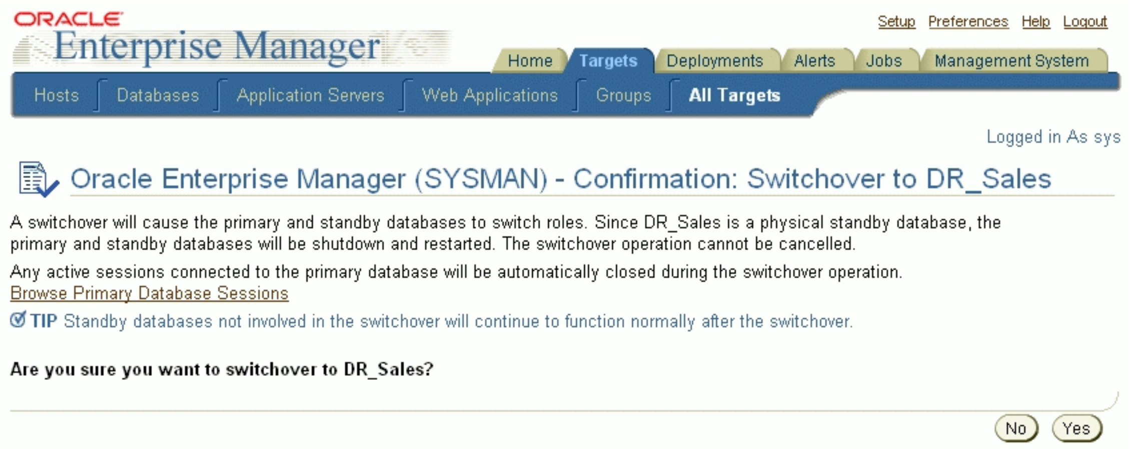
There may be occasions when you might want to perform a switchover between the primary database and standby databases. This scenario steps you through the process of using the switchover operation to switch roles between the primary database and a standby database.

To execute a switchover, perform the following steps:

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 5–26 shows the Switchover Confirmation page.

Figure 5–26 Switchover Operation


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Logged in As sys
No
Yes

Oracle Enterprise Manager (SYSMAN) - Confirmation: Switchover to DR_Sales

A switchover will cause the primary and standby databases to switch roles. Since DR_Sales is a physical standby database, the primary and standby databases will be shutdown and restarted. The switchover operation cannot be cancelled.

Any active sessions connected to the primary database will be automatically closed during the switchover operation.
Browse Primary Database Sessions

TIP Standby databases not involved in the switchover will continue to function normally after the switchover.

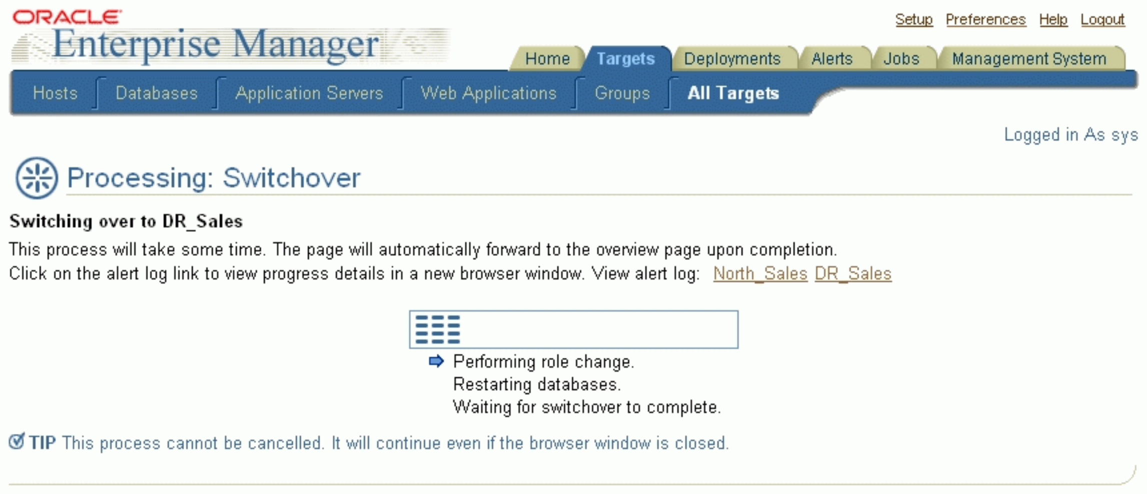
Are you sure you want to switchover to DR_Sales?

The Switchover operation performs the following steps:

1. Ensures that the primary database and standby database are not currently in an error status condition and verifies that broker management of the primary database is enabled and online.
2. If the switchover target is a physical standby database, you can view any active user sessions connected to the primary using the Browse Primary Database Sessions link. These sessions will be closed automatically during the switchover.
3. Performs the switchover by first changing the primary database to the standby role, and then the standby database to the primary role, displaying a progress indicator as the switchover progresses.
4. If the switchover target is a physical standby database, both it and the primary database will be restarted.

Figure 5–27 shows the Processing page during the switchover.

Figure 5–27 Processing Page During Switchover



ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Logged in As sys

Processing: Switchover

Switching over to DR_Sales

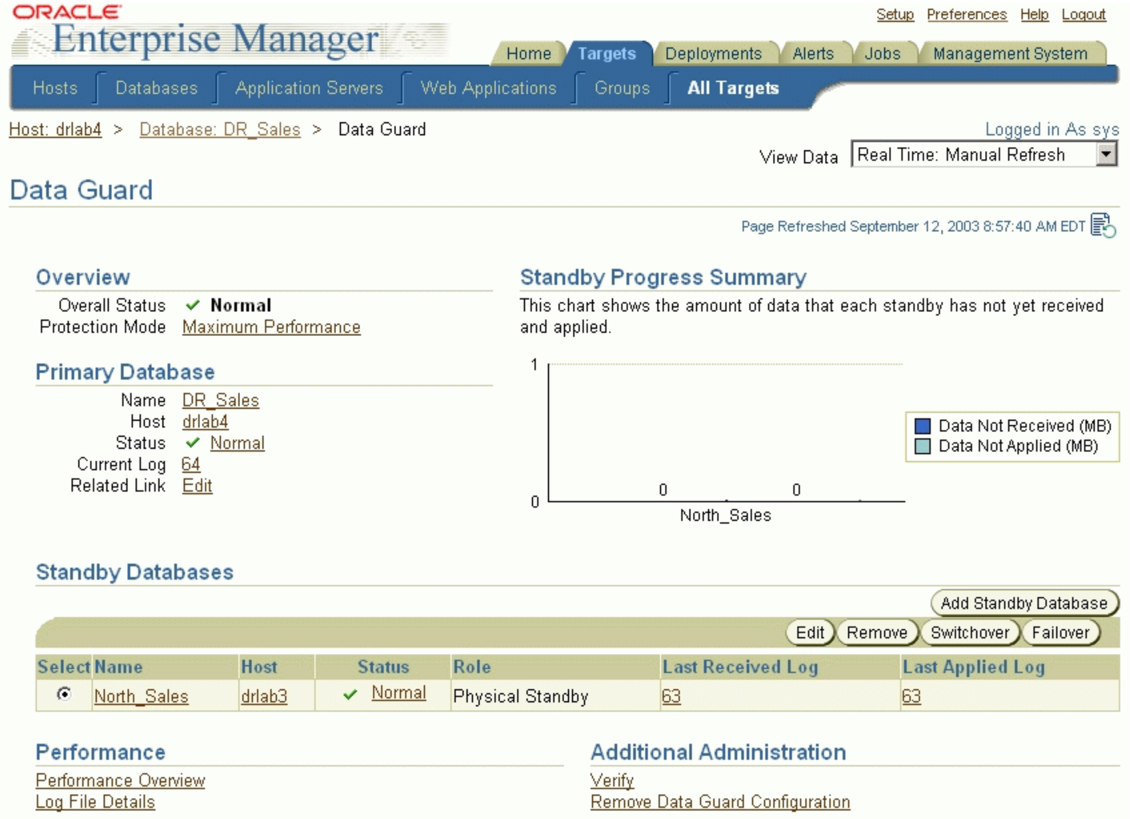
This process will take some time. The page will automatically forward to the overview page upon completion.
Click on the alert log link to view progress details in a new browser window. View alert log: North_Sales DR_Sales

➡ Performing role change.
Restarting databases.
Waiting for switchover to complete.

TIP This process cannot be cancelled. It will continue even if the browser window is closed.

Figure 5–28 shows the Data Guard Overview page after a successful switchover.

Figure 5–28 New Primary Database After Switchover



Host: drlab4 > Database: DR_Sales > Data Guard

Logged in As sys

View Data: Real Time: Manual Refresh

Data Guard

Page Refreshed September 12, 2003 8:57:40 AM EDT

Overview

Overall Status	✓ **Normal**
Protection Mode	Maximum Performance

Primary Database

Name	DR_Sales
Host	drlab4
Status	✓ Normal
Current Log	64
Related Link	Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	North_Sales	drlab3	✓ Normal	Physical Standby	63	63

Performance

Performance Overview
Log File Details

Additional Administration

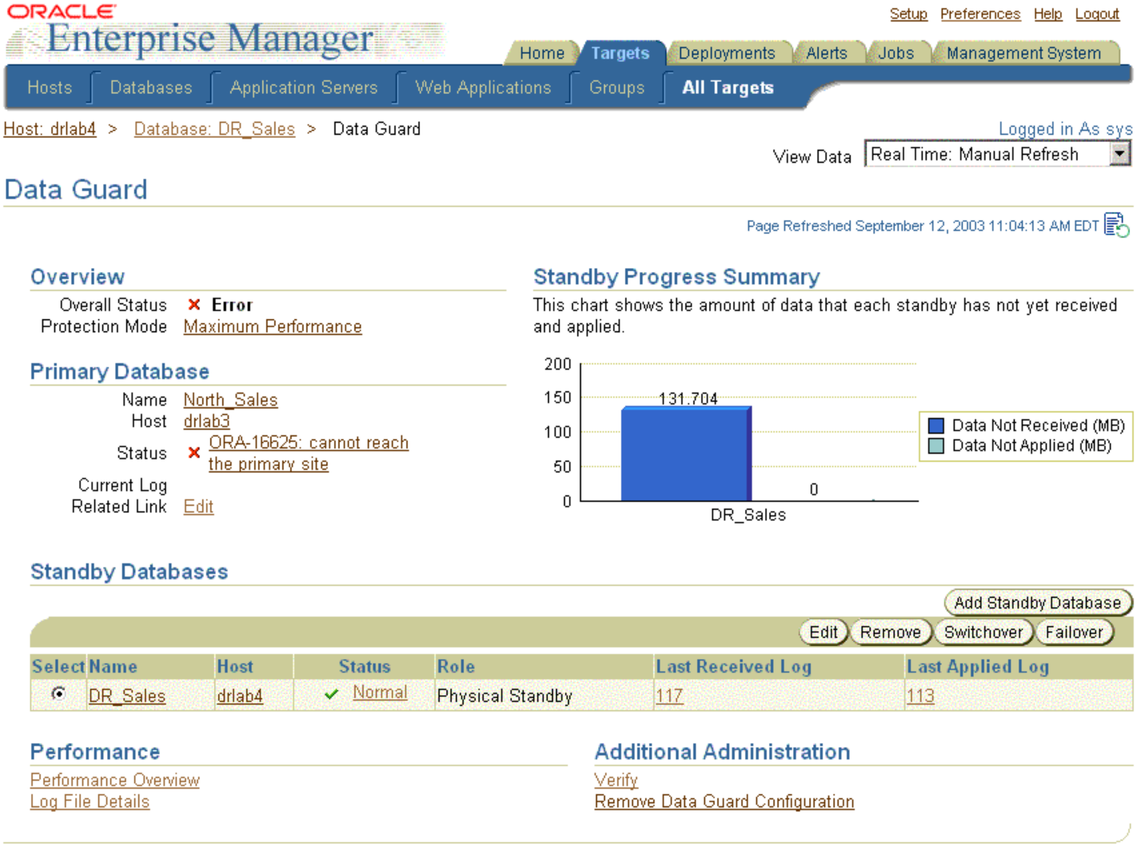
Verify
Remove Data Guard Configuration

5.6 Scenario 6: Performing a Failover Operation

This scenario steps you through the process of using the Failover operation to transition one of the physical standby databases, in this case `DR_Sales`, into the primary role. You should perform a failover only in the event of a software or system failure that results in the loss of the primary database. The failed primary database is disabled by the broker and must be re-created, and the standby database assumes the primary database role. See Section 4.2.5 for steps to re-create the failed primary database as a standby database to the new primary database.

In Figure 5–29 the Data Guard Overview page shows the ORA-16625 error status that indicates problems accessing the primary database.

Figure 5–29 Data Guard Overview Page Showing ORA-16625 Error



Host: drlab4 > Database: DR_Sales > Data Guard

Logged in As sys

View Data: Real Time: Manual Refresh

Data Guard

Page Refreshed September 12, 2003 11:04:13 AM EDT

Overview

Overall Status: ✗ Error

Protection Mode: Maximum Performance

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

Primary Database

Name: North_Sales

Host: drlab3

Status: ✗ ORA-16625: cannot reach the primary site

Current Log

Related Link: Edit

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	DR_Sales	drlab4	✓ Normal	Physical Standby	117	113

Performance

Performance Overview

Log File Details

Additional Administration

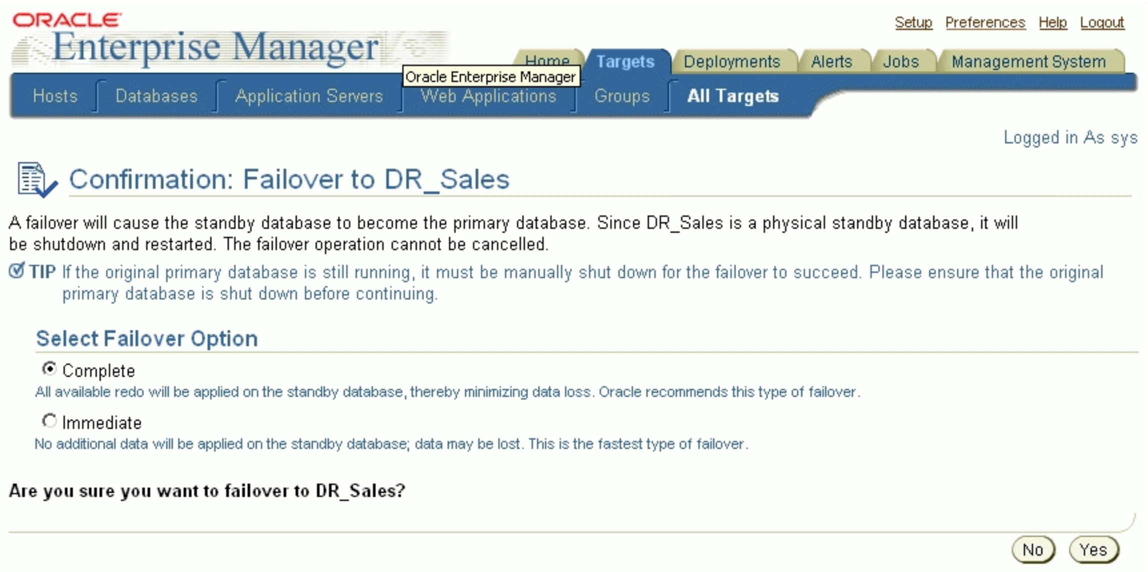
Verify

Remove Data Guard Configuration

****

To transition `DR_Sales` into the primary role, select `DR_Sales` in the Standby Databases table and click **Failover**. Figure 5–30 shows the Failover Confirmation page.

Figure 5–30 Failover Confirmation Page



Confirmation: Failover to DR_Sales

A failover will cause the standby database to become the primary database. Since DR_Sales is a physical standby database, it will be shutdown and restarted. The failover operation cannot be cancelled.

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

● Complete
All available redo will be applied on the standby database, thereby minimizing data loss. Oracle recommends this type of failover.

○ 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?

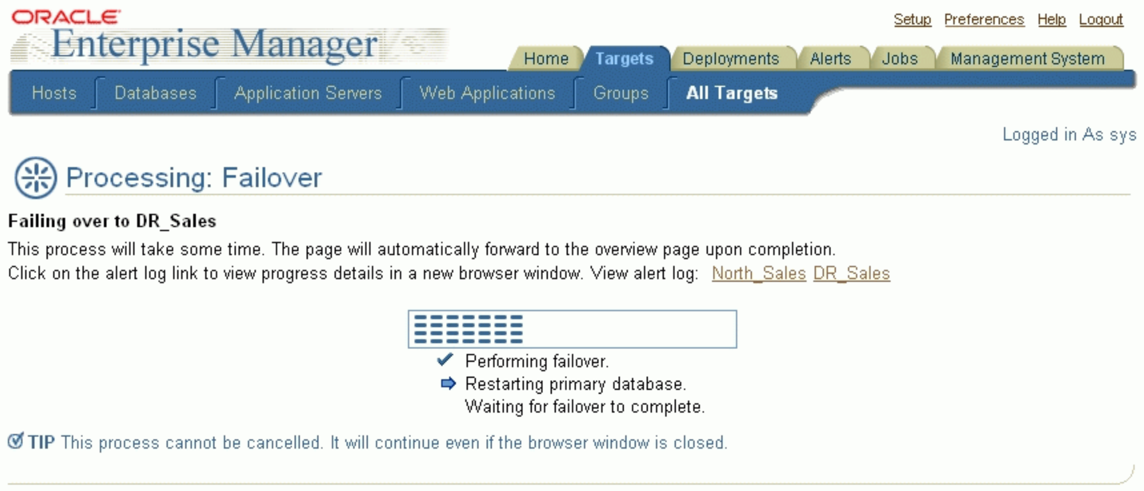
If you determine that a failure occurred on the primary database and there is no possibility of recovering the primary database in a timely manner, you can start the Failover operation. Oracle recommends that you choose a physical standby database (instead of a logical standby database) as the target of a failover.

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

- Complete—attempts to minimize data loss by applying all available redo on the standby database.
- Immediate—no additional data will be applied on the standby database; data may be lost. This is the fastest type of failover.

Figure 5–31 shows the progress of the failover operation.

Figure 5–31 Failover Progress Page



Processing: Failover

Failing over to DR_Sales

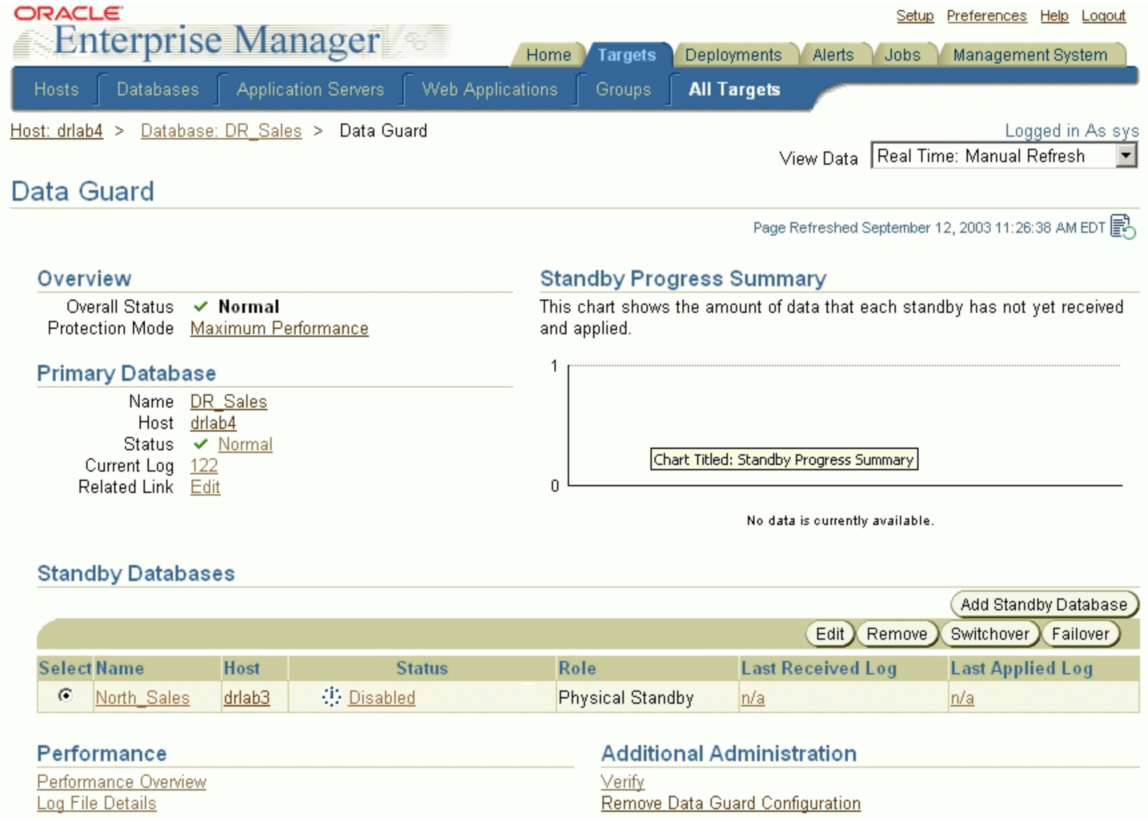
This process will take some time. The page will automatically forward to the overview page upon completion.
Click on the alert log link to view progress details in a new browser window. View alert log: North_Sales DR_Sales

✓ Performing failover.
⇨ Restarting primary database.
Waiting for failover to complete.

TIP This process cannot be cancelled. It will continue even if the browser window is closed.

During the failover, the selected 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 5–32.

Figure 5–32 Data Guard Overview Page After a Failover Operation Completes



Host: drlab4 > Database: DR_Sales > Data Guard

View Data: Real Time: Manual Refresh

Data Guard

Page Refreshed September 12, 2003 11:26:38 AM EDT

Overview

Overall Status	✓ **Normal**
Protection Mode	Maximum Performance

Primary Database

Name	DR_Sales
Host	drlab4
Status	✓ Normal
Current Log	122
Related Link	Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

No data is currently available.

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	North_Sales	drlab3	Disabled	Physical Standby	n/a	n/a

Performance

Performance Overview
Log File Details

Additional Administration

Verify
Remove Data Guard Configuration

In the figure, the Status column indicates that the original primary database (`North_Sales`) is disabled and can no longer be managed through the Data Guard GUI until it has been re-created.

> **See Also:** Section 4.2.5 for more information about post-failover re-creation steps

5.7 Scenario 7: Monitoring a Data Guard Configuration

The Data Guard GUI provides ways to monitor the status of a configuration as well as the online redo log file activity of the primary and standby databases. At the most basic level, the Data Guard Overview page for the configuration not only displays information about the configuration, but it also includes summary information about its databases.

> **Note:** You can access the monitoring functions of Data Guard even if you are logged in as a non-SYSDBA user. However, all management features, such as standby creation, switchover, and failover, require a SYSDBA connection.

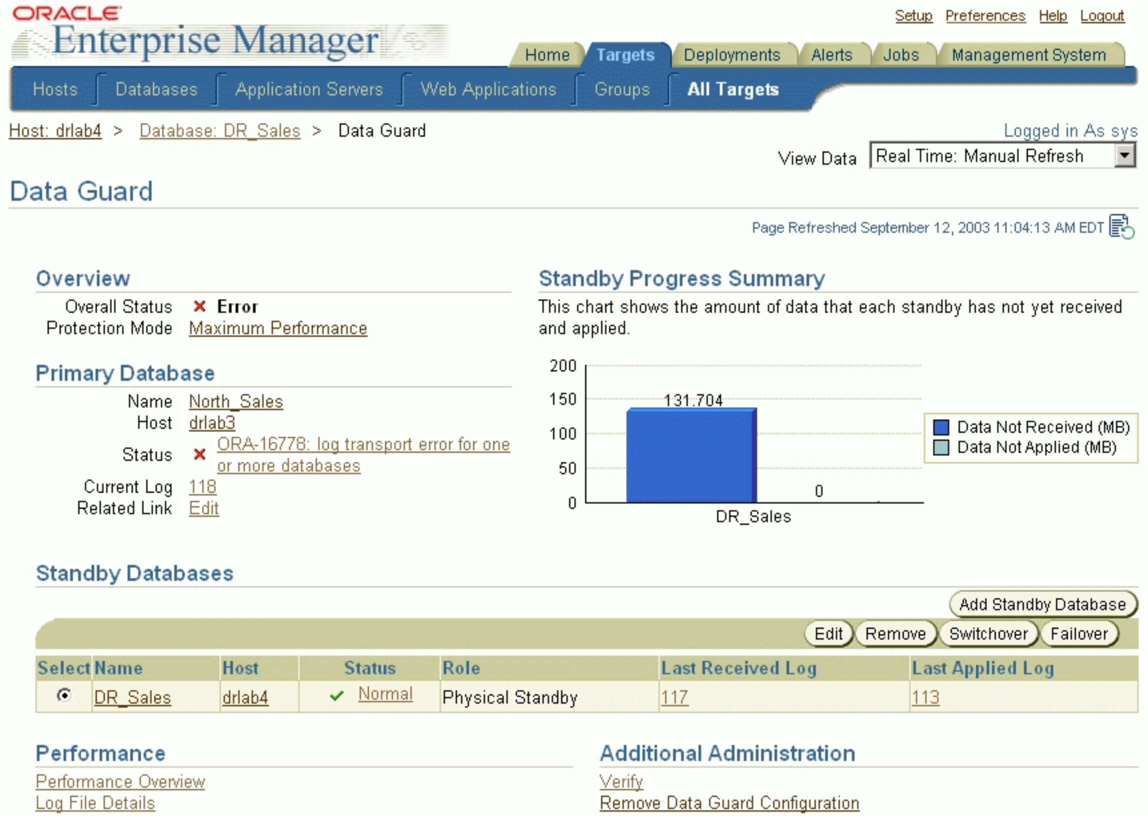
For example, the summary information on the Data Guard Overview page shows the status for all of the databases in the configuration. If you want more information about why log apply services are off for a specific standby database, select the Status link of the database in the Standby Databases table and view the property pages. Any Data Guard specific database properties that are incorrect, inconsistent, or known to be in conflict with other parameters will be flagged with a warning in the Edit Properties page for the database. A variety of icons indicating the status condition can appear next to the Status link. A green check mark appears if the primary database is functioning normally; a yellow triangle containing an exclamation mark appears if there is a warning; and a red X appears if there is an error condition.

- To check the primary database status, select **Status** under the Primary Database section of the Data Guard Overview page.
- To check the status of a standby database listed in the Standby Databases table, select the link under the Status column.

  For example, a yellow warning icon may display the message "A yellow warning indicates an inconsistent property has been detected. The value for this property is inconsistent between Data Guard and the database, Data Guard and the server parameter file, or Data Guard and both the database and server parameter file."

Figure 5–33 shows the Data Guard Overview page in the case where the configuration has a log transport error.

Figure 5–33 Data Guard Overview Page Showing Log Transport Error


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Add Standby Database
Edit Remove Switchover Failover
200
150
100
50
0
131.704
0
DR_Sales
Data Not Received (MB)
Data Not Applied (MB)

Host: drlab4 > Database: DR_Sales > Data Guard

Logged in As sys

View Data Real Time: Manual Refresh

Data Guard

Page Refreshed September 12, 2003 11:04:13 AM EDT

Overview

Overall Status ✗ Error
Protection Mode Maximum Performance

Primary Database

Name North_Sales
Host drlab3
Status ✗ ORA-16778: log transport error for one or more databases
Current Log 118
Related Link Edit

Standby Progress Summary

This chart shows the amount of data that each standby has not yet received and applied.

Standby Databases

Select	Name	Host	Status	Role	Last Received Log	Last Applied Log
●	DR_Sales	drlab4	✓ Normal	Physical Standby	117	113

Performance

Performance Overview
Log File Details

Additional Administration

Verify
Remove Data Guard Configuration

The following sections provide information on ways to monitor the status and health of a configuration:

- Section 5.7.1, "Verifying a Broker Configuration"
- Section 5.7.2, "Viewing Log File Details"
- Section 5.7.3, "Monitoring Configuration Performance"

5.7.1 Verifying a Broker Configuration

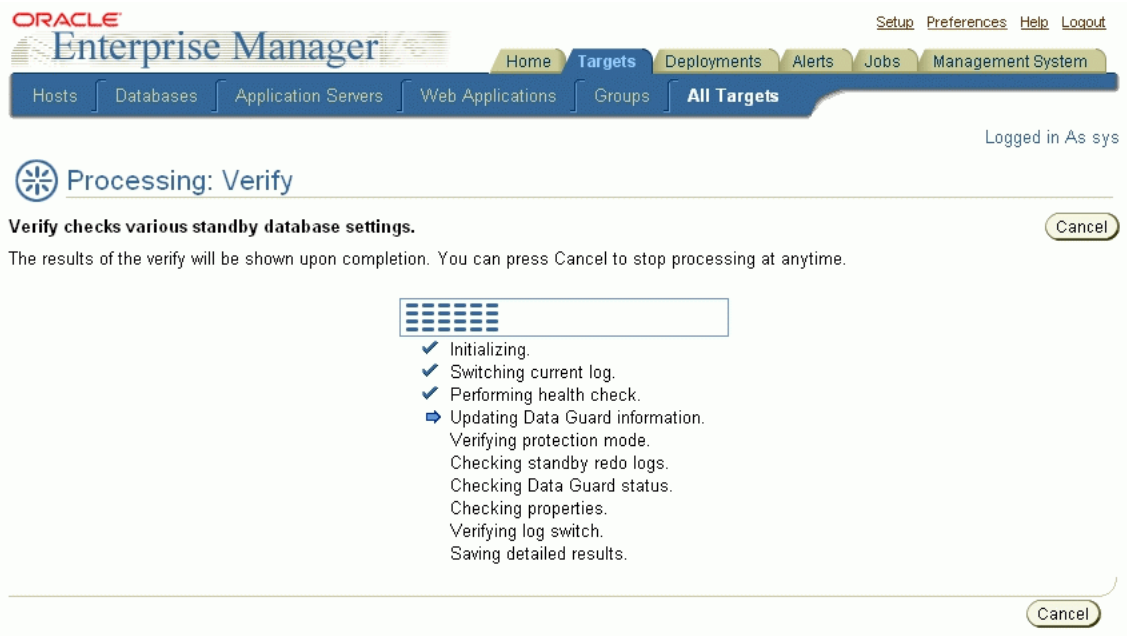
Another way to quickly check the overall health of a broker configuration is to run the Verify operation. The Verify operation performs a series of validation checks on the broker configuration, including a health check of each database in the configuration. The checks include:

1. Shows the current data protection mode setting, including the current log transport mode settings for each database and whether or not the standby redo log files are configured properly. If standby redo log files are needed for any database, the Verify results will allow you to automatically configure them.
2. Validates each database for the current status.
3. Performs a log switch on the primary database and then verifies that the log file was applied on each standby database.
4. Shows the results of the Verify operation, including errors, if any. The Verify operation completes successfully if there are no errors and an online redo log file was successfully applied to at least one standby database.
5. Shows any databases or RAC instances that are not discovered. Undiscovered databases and instances could prevent a failover or switchover from completing successfully.
6. Detects inconsistencies between database properties and their corresponding values in the database itself. It also provides a mechanism for resolving these inconsistencies.

Note: You can click **Cancel** at any time to stop the Verify operation.

To verify the configuration, click **Verify** under the Additional Administration section of the Data Guard Overview page. See Figure 5–34. The Verify command displays a progress indicator. When the Verify operation completes successfully, the broker configuration is healthy, guarding the data, and ready for a switchover or failover.

Figure 5–34 Verifying the Configuration



Processing: Verify

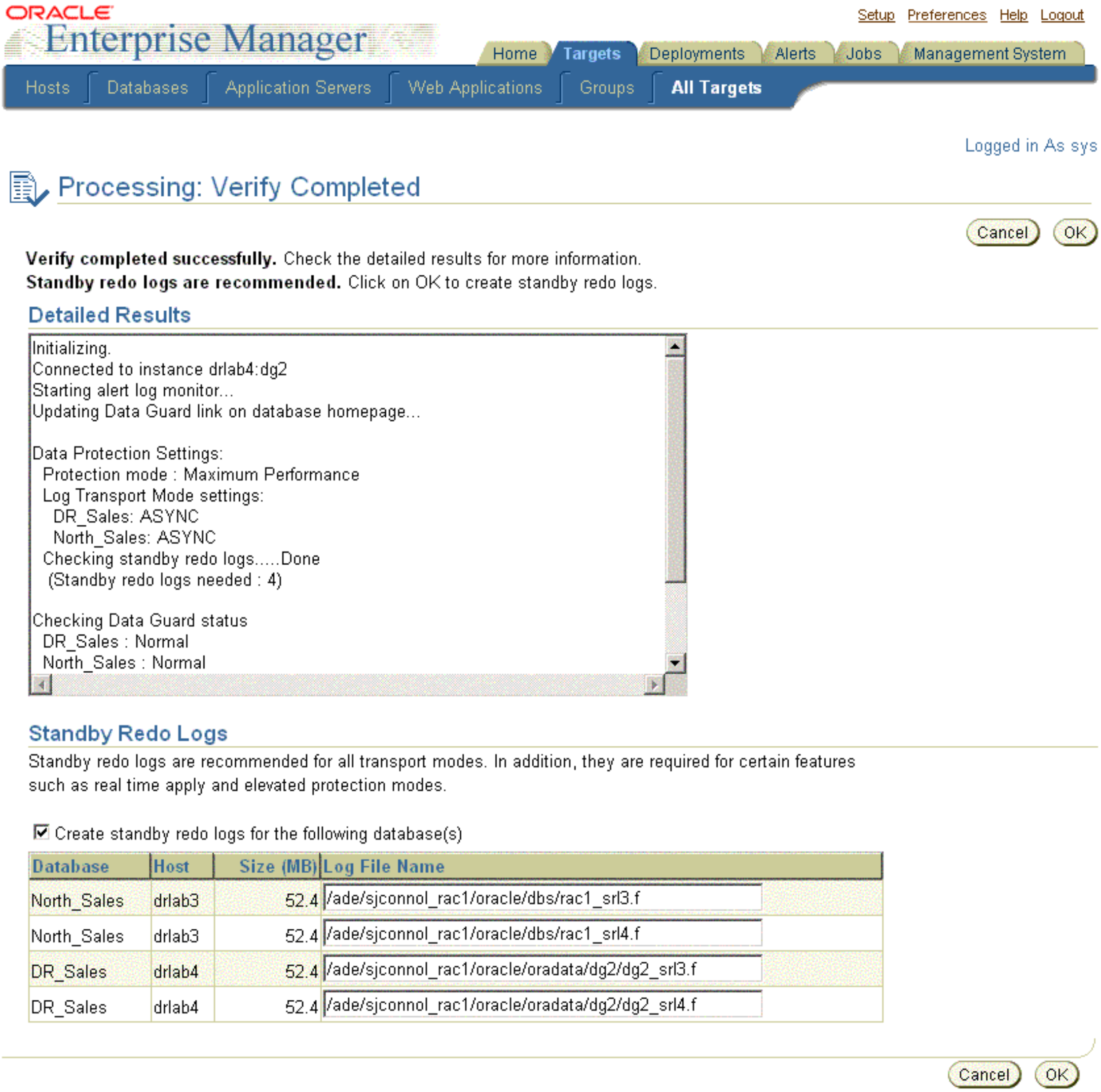
Verify checks various standby database settings.

The results of the verify will be shown upon completion. You can press Cancel to stop processing at anytime.

✓ Initializing.
✓ Switching current log.
✓ Performing health check.
⇨ Updating Data Guard information.
Verifying protection mode.
Checking standby redo logs.
Checking Data Guard status.
Checking properties.
Verifying log switch.
Saving detailed results.

Figure 5–35 shows the results of running the Verify operation.

Figure 5–35 Results of the Verify Command


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Logged in As sys

Processing: Verify Completed

Verify completed successfully. Check the detailed results for more information.
Standby redo logs are recommended. Click on OK to create standby redo logs.

Detailed Results

```
Initializing.
Connected to instance drlab4:dg2
Starting alert log monitor...
Updating Data Guard link on database homepage...

Data Protection Settings:
  Protection mode : Maximum Performance
  Log Transport Mode settings:
    DR_Sales: ASYNC
    North_Sales: ASYNC
  Checking standby redo logs.....Done
   (Standby redo logs needed : 4)

Checking Data Guard status
  DR_Sales : Normal
  North_Sales : Normal
```

Standby Redo Logs

Standby redo logs are recommended for all transport modes. In addition, they are required for certain features such as real time apply and elevated protection modes.

☑ Create standby redo logs for the following database(s)

Database	Host	Size (MB)	Log File Name
North_Sales	drlab3	52.4	/ade/sjconnol_rac1/oracle/dbs/rac1_srl3.f
North_Sales	drlab3	52.4	/ade/sjconnol_rac1/oracle/dbs/rac1_srl4.f
DR_Sales	drlab4	52.4	/ade/sjconnol_rac1/oracle/oradata/dg2/dg2_srl3.f
DR_Sales	drlab4	52.4	/ade/sjconnol_rac1/oracle/oradata/dg2/dg2_srl4.f

5.7.2 Viewing Log File Details

For each standby database in the configuration, there is a table that shows the status of archived redo log files that were generated on the primary database but not received by the standby database, and for archived redo log files that were received but not applied to the standby database. This Log File Details page is useful to determine log file information when log transport or log apply services are stopped. Under normal operations, each standby table is empty.

For example, if log transport services go offline unexpectedly, the primary database continues to generate archived redo log files, but log transport services will not send the archived redo log files to the standby databases. You can view the Log File Details page to find out which log files have not yet been received for each standby database.

On the Log File Details page, there is a *Primary Current Log* entry that indicates the log sequence number of the current log file on the primary.

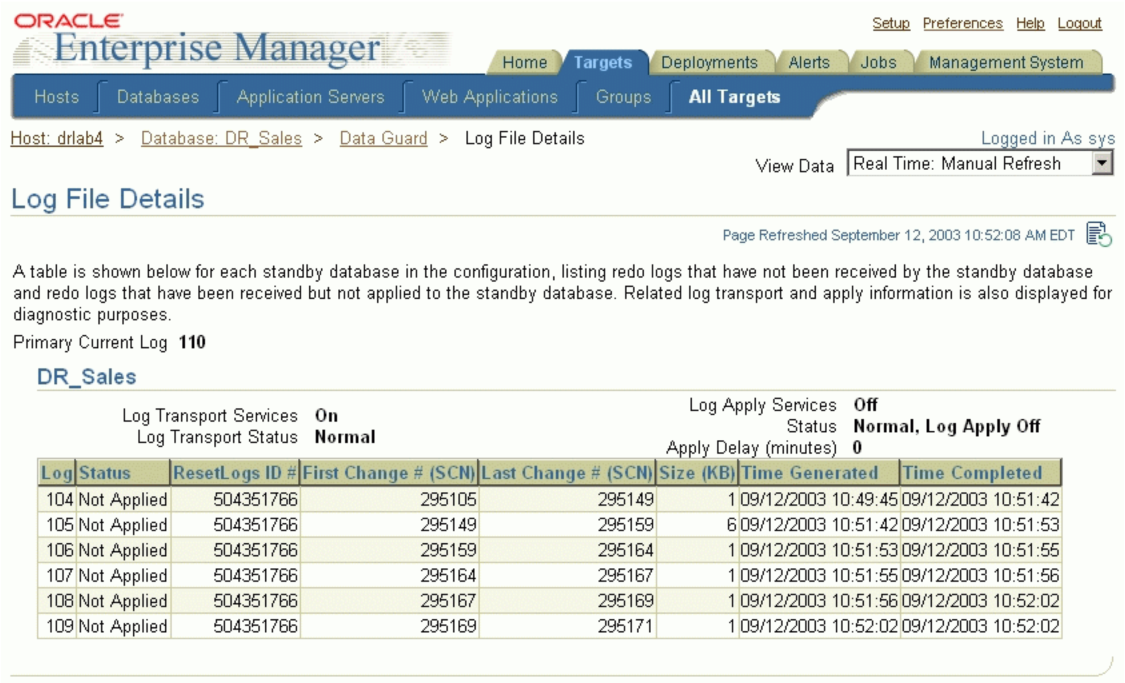
For each standby database, log transport and log apply information is displayed to help diagnose any log file buildup. Table 5–1 describes the columns for the standby table:

Table 5–1 Log File Details Page

Column Title	Description
Log	The log sequence number.
Status	The status of the log file.
Not Received	The log file has not been received.
Not Applied	The log file has not been applied.
First Change # (SCN)	The first system change number.
Last Change # (SCN)	The last system change number.
Size (KB)	The size, in kilobytes, of the log file.
Time Generated	The time when the log file was generated.
Time Completed	The time at which the log file was completed.

Click **Log File Details** from the Data Guard Overview page to see the page shown in Figure 5–36.

Figure 5–36 Viewing Log File Details



Host: drlab4 > Database: DR_Sales > Data Guard > Log File Details

Logged in As sys

View Data Real Time: Manual Refresh

Log File Details

Page Refreshed September 12, 2003 10:52:08 AM EDT

A table is shown below for each standby database in the configuration, listing redo logs that have not been received by the standby database and redo logs that have been received but not applied to the standby database. Related log transport and apply information is also displayed for diagnostic purposes.

Primary Current Log **110**

DR_Sales

Log Transport Services **On**
Log Transport Status **Normal**

Log Apply Services **Off**
Status **Normal, Log Apply Off**
Apply Delay (minutes) **0**

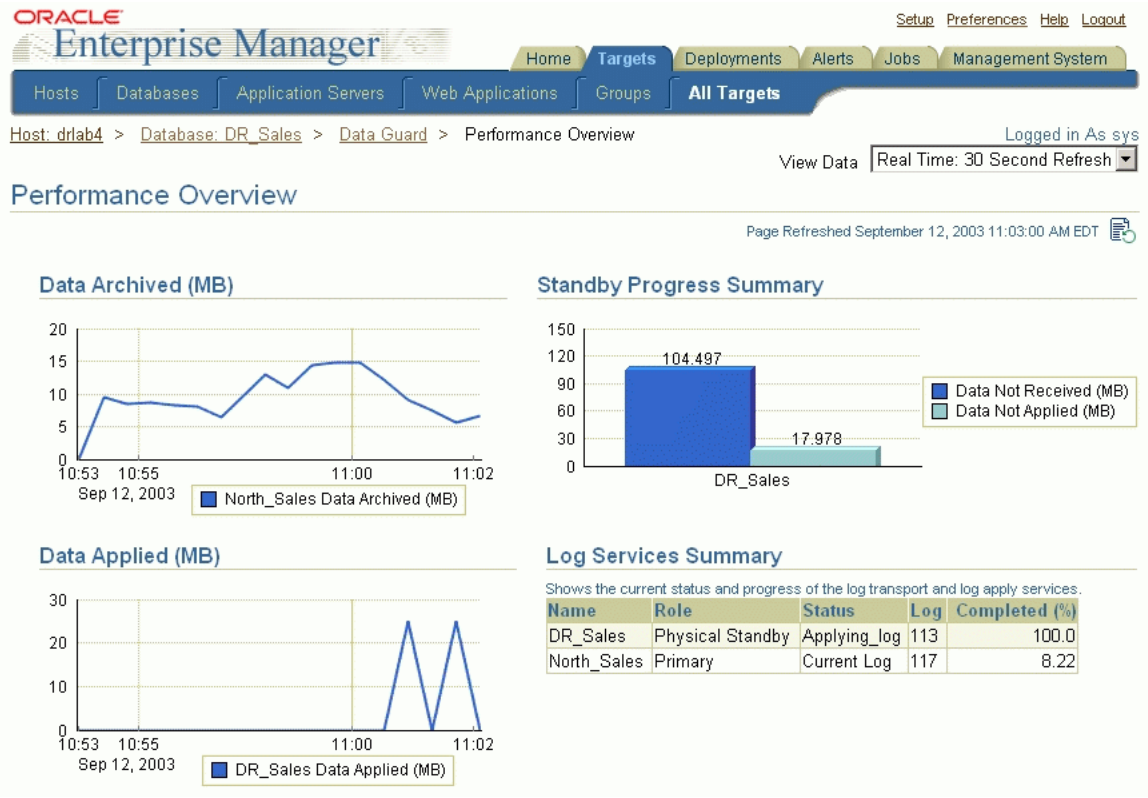
Log	Status	ResetLogs ID #	First Change # (SCN)	Last Change # (SCN)	Size (KB)	Time Generated	Time Completed
104	Not Applied	504351766	295105	295149	1	09/12/2003 10:49:45	09/12/2003 10:51:42
105	Not Applied	504351766	295149	295159	6	09/12/2003 10:51:42	09/12/2003 10:51:53
106	Not Applied	504351766	295159	295164	1	09/12/2003 10:51:53	09/12/2003 10:51:55
107	Not Applied	504351766	295164	295167	1	09/12/2003 10:51:55	09/12/2003 10:51:56
108	Not Applied	504351766	295167	295169	1	09/12/2003 10:51:56	09/12/2003 10:52:02
109	Not Applied	504351766	295169	295171	1	09/12/2003 10:52:02	09/12/2003 10:52:02

5.7.3 Monitoring Configuration Performance

For more in-depth performance and monitoring, you can display detailed performance statistics for a broker configuration using performance charts that provide a graphical summary of all online redo log file activity in the configuration. The charts are refreshed based on a collection interval (the rate at which data is sampled from the primary database) that you can specify. The default collection interval is 30 seconds, which can be changed. See the online Help for detailed information about performance sampling rates.

For example, Figure 5–37 shows performance information for all of the objects in the configuration.

Figure 5–37 Performance Page


ORACLE Enterprise Manager
Setup Preferences Help Logout
Home Targets Deployments Alerts Jobs Management System
Hosts Databases Application Servers Web Applications Groups All Targets
Data Archived (MB)
20 15 10 5 0
10:53 10:55 11:00 11:02
Sep 12, 2003
North_Sales Data Archived (MB)
Standby Progress Summary
150 120 90 60 30 0
104.497
17.978
DR_Sales
Data Not Received (MB)
Data Not Applied (MB)
Data Applied (MB)
30 20 10 0
10:53 10:55 11:00 11:02
Sep 12, 2003
DR_Sales Data Applied (MB)

Host: drlab4 > Database: DR_Sales > Data Guard > Performance Overview

Logged in As sys

View Data: Real Time: 30 Second Refresh

Performance Overview

Page Refreshed September 12, 2003 11:03:00 AM EDT

Log Services Summary

Shows the current status and progress of the log transport and log apply services.

Name	Role	Status	Log	Completed (%)
DR_Sales	Physical Standby	Applying_log	113	100.0
North_Sales	Primary	Current Log	117	8.22

The Performance Overview page begins charting information as soon as the page is displayed. Data will not be collected for any offline or disabled databases.

5.8 Scenario 8: Using Metrics

In addition to monitoring the status and log file activity using the Data Guard GUI, Oracle Enterprise Manager automatically monitors the status and archived redo log file activity on the primary and standby databases.

Table 5–2 describes the five Data Guard metrics.

Table 5–2 Data Guard Metrics

Metric	Description
Data Guard Status	Checks the status of each database in the broker configuration.
Data Not Applied (MB)[1]	Measures the difference (in megabytes) between the last log file received and the last log file applied on each standby database.
Data Not Applied (log files)	Measures the difference (in number of archived redo log files) between the last log file received and the last log file applied on each standby database.
Data Not Received (MB)[1]	Measures the difference (in megabytes) between the current log file on the primary database and the last log file received on each standby database.
Data Not Received (log files)	Measures the difference (in number of archived redo log files) between the current log file on the primary database and the last log file received on each standby database.

[1] For Oracle Database 10*g* database only.

You can set up Email Services to notify you through your e-mail if any of the metrics are triggered.

> **See Also:** Oracle Enterprise Manager help and documentation for more information about registering metrics and using Email Services

5.8.1 Understanding the Data Guard Metrics

The following sections provide additional information about the Data Guard metrics.

5.8.1.1 Data Guard Status

The Data Guard Status metric checks the status of each database in the broker configuration and triggers a warning or critical alert if necessary.

5.8.1.2 Data Not Applied (MB)

The broker computes the highest applied SCN and uses its value to find the last continuous log that was successfully archived to the standby database. The size of redo data in all subsequent log files are counted as data not applied. If the primary database goes down at this point, these logs can be applied on the standby database. If there is a gap in the logs received on the standby database, any logs received after the gap cannot be applied. See the *Oracle Data Guard Concepts and Administration* for information about archive gaps.

For example, if logs 1, 2, 3, 6, 7, and 9 are received on the standby database and log apply services is currently applying log 1, log apply services can continue to apply up to log 3. Log apply services cannot apply any more logs because log 4 is missing. Even though logs 6, 7, and 9 are received, they cannot be applied and they will not be counted as data not applied. In this case, the total size of logs 1, 2, and 3 is the size of Data Not Applied.

If all the archived log files on the standby database are continuous, and standby redo logs are used, the standby redo logs are also counted as data not applied, unless real-time apply is turned on and log apply services is already working on the standby redo logs. The size of an archived log file is its file size. However, the size of a standby redo log is the size of the actual redo in the log and not the file size.

If the standby redo logs are multithreaded, the broker computes the highest applied SCN for every thread and totals the numbers. If there are multiple incarnations and the standby database is in a different incarnation from the primary database, each incarnation is computed separately and the results are then totaled.

5.8.1.3 Data Not Applied (Log Files)

The broker computes the highest applied SCN and uses its value to find the last continuous log that was successfully archived to the standby database. Redo data in all subsequent log files are counted as logs not applied. If the primary database goes down at this point, these logs can be applied on the standby database. If there is a gap in the logs received on the standby database, any logs received after the gap cannot be applied. See the *Oracle Data Guard Concepts and Administration* for information about archive gaps.

For example, if logs 1, 2, 3, 6, 7, and 9 are received on the standby database and log apply services is currently applying log 1, log apply services can continue to apply up to log 3. Log apply services cannot apply any more logs because log 4 is missing. Even though logs 6, 7, and 9 are received, they cannot be applied and they will not be counted as data not applied.

If all the archived log files on the standby database are continuous, and standby redo logs are used, the standby redo logs are also counted as data not applied, unless real-time apply is turned on and log apply services is already working on the standby redo logs.

If the standby redo logs are multithreaded, the broker computes the highest applied SCN for every thread and totals the numbers. If there are multiple incarnations and the standby database is in a different incarnation from the primary database, each incarnation is computed separately and the results are then totaled.

5.8.1.4 Data Not Received (MB)

The broker computes the highest applied SCN and uses its value to find the last continuous log that was successfully archived to the standby database. The size of redo data in all subsequent log files, including the current online redo log file, are counted as data for potential data loss and will be unrecoverable if the primary database goes down at this point. The size of an archived log file is its file size, and the size of the online redo log file is the size of the actual redo in the online log, not the file size of the online redo log file.

For example, if logs 1, 2, 3, 6, 7, and 9 are received on the standby database, and if log 10 is the current online log file, and if log apply services is currently applying log 1, the last continuous log after the highest applied SCN is log 3. All logs after log 3, that is logs 4 through 10, are counted as data not received and the total size of redo data in these logs is the size of Data Not Received.

If the primary database is multithreaded (in a RAC database), the broker computes the highest applied SCN for every thread and totals the numbers. If the primary database has multiple incarnations (for example, due to a flashback operation) and the standby database is in a different incarnation from the primary database, the computation is done on each incarnation and the results are then totaled.

5.8.1.5 Data Not Received (Log Files)

The broker computes the highest applied SCN and uses its value to find the last continuous log that was successfully archived to the standby database. Redo data in all subsequent log files, including the current online redo log file, are counted as logs for potential data loss and will be unrecoverable if the primary database goes down at this point.

For example, if logs 1, 2, 3, 6, 7, and 9 are received on the standby database, and if log 10 is the current online log file, and if log apply services is currently applying log 1, the last continuous log after the highest applied SCN is log 3. All logs after log 3, that is logs 4 through 10, are counted as data not received. If the primary database goes down at this point, all redo data in logs 4 through 10 are lost on the standby database.

If the primary database is multithreaded (in a RAC database), the broker computes the highest applied SCN for every thread and totals the numbers. If the primary database has multiple incarnations (for example, due to a flashback operation) and the standby database is in a different incarnation from the primary database, the computation is done on each incarnation and the results are then totaled.

5.8.2 Managing Data Guard Metrics

The example in this section describes Data Guard metrics and how to set up for notification through e-mail when a metric is triggered. Take the following steps to manage Data Guard metrics:

1. Set up to receive metric notification by e-mail.
2. View the All Metrics page.
3. Set or change metric thresholds for Data Guard thresholds.
4. View triggered metrics.

Step 1 Set up to receive metric notification by e-mail.

You can set up the Notification Methods to notify you through e-mail if any of the metrics are triggered. To set up Notification Methods, take the following steps:

1. Click **Setup** at the bottom of the Database Home page.
2. Click **Notification Methods** on the Setup page.
3. Set up the required mail server information.

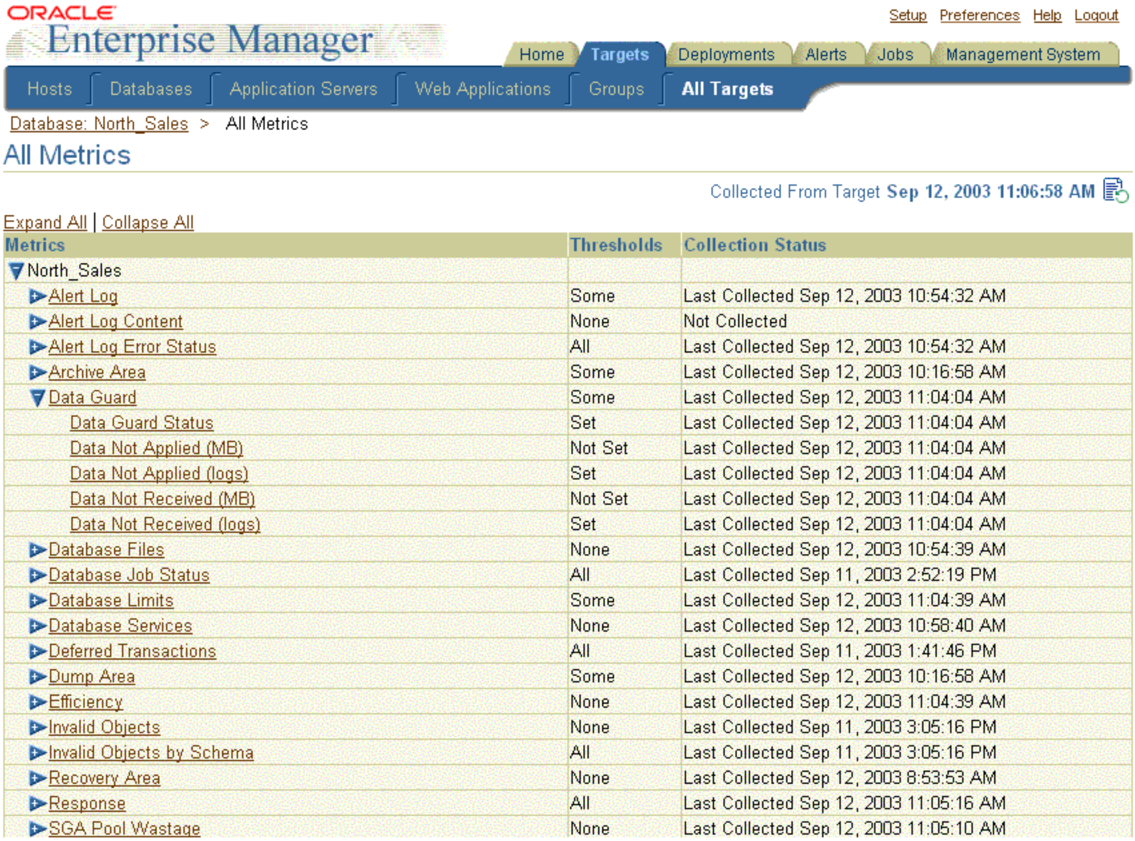
> **See Also:** Oracle Enterprise Manager documentation and help for a complete description of Email Services

Step 2 View the All Metrics page.

You can view all of the metrics for Oracle Enterprise Manager, including Data Guard, by clicking **All Metrics** at the bottom of the Database Home page.

Figure 5–38 shows the All Metrics page for the primary database with the Data Guard metrics expanded.

Figure 5–38 ***Data Guard Metrics***



Database: North_Sales > All Metrics

All Metrics

Collected From Target **Sep 12, 2003 11:06:58 AM**

Metrics	Thresholds	Collection Status
▼North_Sales		
▶Alert Log	Some	Last Collected Sep 12, 2003 10:54:32 AM
▶Alert Log Content	None	Not Collected
▶Alert Log Error Status	All	Last Collected Sep 12, 2003 10:54:32 AM
▶Archive Area	Some	Last Collected Sep 12, 2003 10:16:58 AM
▼Data Guard	Some	Last Collected Sep 12, 2003 11:04:04 AM
Data Guard Status	Set	Last Collected Sep 12, 2003 11:04:04 AM
Data Not Applied (MB)	Not Set	Last Collected Sep 12, 2003 11:04:04 AM
Data Not Applied (logs)	Set	Last Collected Sep 12, 2003 11:04:04 AM
Data Not Received (MB)	Not Set	Last Collected Sep 12, 2003 11:04:04 AM
Data Not Received (logs)	Set	Last Collected Sep 12, 2003 11:04:04 AM
▶Database Files	None	Last Collected Sep 12, 2003 10:54:39 AM
▶Database Job Status	All	Last Collected Sep 11, 2003 2:52:19 PM
▶Database Limits	Some	Last Collected Sep 12, 2003 11:04:39 AM
▶Database Services	None	Last Collected Sep 12, 2003 10:58:40 AM
▶Deferred Transactions	All	Last Collected Sep 11, 2003 1:41:46 PM
▶Dump Area	Some	Last Collected Sep 12, 2003 10:16:58 AM
▶Efficiency	None	Last Collected Sep 12, 2003 11:04:39 AM
▶Invalid Objects	None	Last Collected Sep 11, 2003 3:05:16 PM
▶Invalid Objects by Schema	All	Last Collected Sep 11, 2003 3:05:16 PM
▶Recovery Area	None	Last Collected Sep 12, 2003 8:53:53 AM
▶Response	All	Last Collected Sep 12, 2003 11:05:16 AM
▶SGA Pool Wastage	None	Last Collected Sep 12, 2003 11:05:10 AM

Step 3 Set or change metric thresholds for Data Guard thresholds.

There are five metrics for Data Guard. These metrics can be changed from the Manage Metrics page.

- Data Guard Status

  If the Data Guard Status metric contains Warning, a warning alert is triggered. If the metric contains Error, a critical alert is triggered.

- Data Not Applied (MB)

There is no default value. If you want to receive alerts for these metrics, you need to set these thresholds from the Manage Metrics link from the bottom of the Database Home page.

- Data Not Applied (log files)

  The default values for log files are 1 for the warning threshold and 3 for the critical threshold. If the number of log files is greater than 1, a warning alert is triggered. If the number of log files is greater than 3, a critical alert is triggered. You can change these values from the Manage Metrics link from the bottom of the Database Home page.

- Data Not Received (MB)

  There is no default value. If you want to receive alerts for these metrics, you need to set these thresholds from the Manage Metrics link from the bottom of the Database Home page.

- Data Not Received (log files)

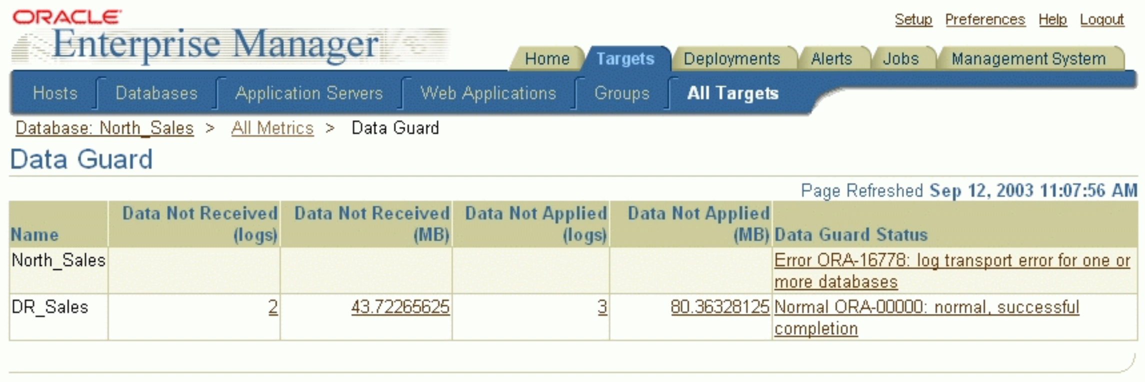
  The default values for log files are 1 for the warning threshold and 3 for the critical threshold. If the number of log files is greater than 1, a warning alert is triggered. If the number of log files is greater than 3, a critical alert is triggered. You can change these values from the Manage Metrics link from the bottom of the Database Home page.

Step 4 View triggered metrics.

If a metric condition is triggered or a threshold value is exceeded, an alert is issued. Click **Data Guard** from the All Metrics page to view triggered metrics.

Figure 5–39 is an example of what is shown when the Data Guard Status metric triggers after a log transport error is detected in the configuration. The table shows the status of each database in the configuration. The status of the primary database shows the ORA-16778 error.

Figure 5–39 Data Guard Triggered Metrics



Database: North_Sales > All Metrics > Data Guard

Data Guard

Page Refreshed **Sep 12, 2003 11:07:56 AM**

Name	Data Not Received (logs)	Data Not Received (MB)	Data Not Applied (logs)	Data Not Applied (MB)	Data Guard Status
North_Sales					Error ORA-16778: log transport error for one or more databases
DR_Sales	2	43.72265625	3	80.36328125	Normal ORA-00000: normal, successful completion

Because Email Services was set up, DBAs are also notified by an e-mail message.

After a metric condition is fixed, the metric is cleared automatically.

5.9 Scenario 9: Removing a Standby Database and Configuration

You can remove a standby database or broker configuration so that it is no longer controlled by the Data Guard broker.

5.9.1 Remove a Standby Database

Removing a standby database from Data Guard broker control does not permanently delete the database itself. This operation permanently removes the profile of the standby database from the broker configuration file.

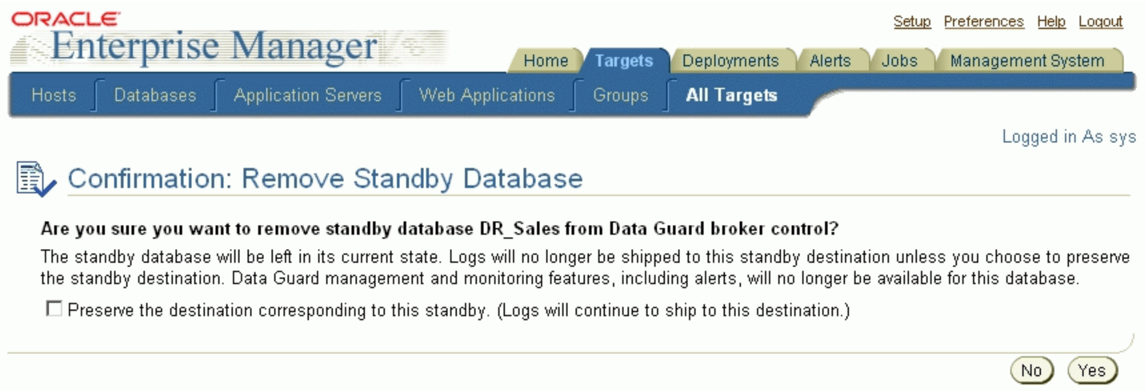
By default, the standby destination is also removed from the primary database so that log files are no longer archived to that standby database.

Caution: The Oracle9*i* Data Guard Manager default was to preserve the standby destination on the primary database.

To remove a standby database from the broker configuration, click **Remove** in the Standby Databases section of the Data Guard Overview page. Data Guard GUI will ask you to confirm that you really want to remove the standby database from the configuration. Click **Yes** to continue or **No** to cancel.

Figure 5–40 shows removing a standby database.

Figure 5–40 Removing a Standby Database



Confirmation: Remove Standby Database

Are you sure you want to remove standby database DR_Sales from Data Guard broker control?

The standby database will be left in its current state. Logs will no longer be shipped to this standby destination unless you choose to preserve the standby destination. Data Guard management and monitoring features, including alerts, will no longer be available for this database.

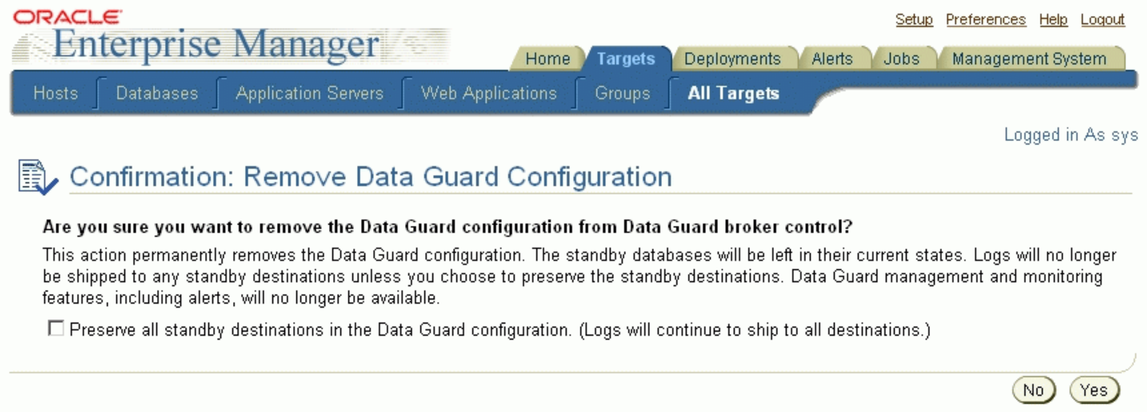
□ Preserve the destination corresponding to this standby. (Logs will continue to ship to this destination.)

5.9.2 Remove the Data Guard Configuration

To remove the broker configuration, you must be connected to the configuration through the primary database. Click **Remove Data Guard Configuration** under the Additional Administration section. The Data Guard GUI will ask you to confirm that you want to remove the configuration. Click **Yes** to continue or **No** to cancel.

Figure 5–41 shows removing a Data Guard broker configuration.

Figure 5–41 Removing a Data Guard Broker Configuration



Confirmation: Remove Data Guard Configuration

Are you sure you want to remove the Data Guard configuration from Data Guard broker control?

This action permanently removes the Data Guard configuration. The standby databases will be left in their current states. Logs will no longer be shipped to any standby destinations unless you choose to preserve the standby destinations. Data Guard management and monitoring features, including alerts, will no longer be available.

□ Preserve all standby destinations in the Data Guard configuration. (Logs will continue to ship to all destinations.)

When you choose this option, the Data Guard GUI removes (deletes) the selected broker configuration permanently. When you permanently remove a configuration, the remove operation:

- Does not affect the underlying operations of the standby databases or log apply services if you check the box to preserve all standby destinations. Operations such as log transport services and log apply services continue to run; however, these services are no longer manageable through the Data Guard GUI.
- By default, all standby database profiles are removed from the broker configuration, and log transport services to all standby databases in the configuration are stopped.
- Destroys all broker configuration profile information maintained for each database; the configuration is then unknown to the broker and can no longer be managed from the Data Guard GUI.

Although the configuration information is not recoverable once you delete a broker configuration permanently, you can easily re-create the configuration with the Add Standby Database wizard.

6

Data Guard Scenarios - Using DGMGRL CLI

This chapter provides several scenarios that show how to use the Data Guard command-line interface (CLI) to create, manage, and monitor a broker configuration.

In addition to the prerequisites for getting started, this chapter describes the following scenarios:

- Scenario 1: Creating a Configuration
- Scenario 2: Setting Database Properties
- Scenario 3: Enabling the Configuration and Databases
- Scenario 4: Setting the Configuration Protection Mode
- Scenario 5: Performing Routine Management Tasks
- Scenario 6: Performing a Switchover Operation
- Scenario 7: Performing a Failover Operation
- Scenario 8: Monitoring a Data Guard Configuration

6.1 Prerequisites for Getting Started

One of the prerequisites for using the CLI is that a primary database and any standby databases must already exist. The DG_BROKER_START initialization parameter must be set to TRUE for all instances in the configuration. You must use a server parameter file with the broker (see Section 1.7.5 and Section 7.1.3).

After starting the Oracle instance, set the DG_BROKER_START=true initialization parameter using the SQL ALTER SYSTEM statement. The parameter value will be saved in the server parameter file. The next time you start the Oracle instance, the broker is started automatically, and you do not need to issue the SQL ALTER SYSTEM statement again.

Convert the initialization parameter files (PFILE) on both primary and standby databases into server parameter files (SPFILE), if necessary. Use the following SQL*Plus command:

```
CREATE SPFILE FROM PFILE='pfilename';
```

If an instance was not started with a server parameter file, then you must shut down the instance and restart it using the server parameter file.

> **See Also:** *Oracle Database Administrator's Guide* for detailed information about creating server parameter files

The following assumptions are made for the scenarios in this chapter:

- TCP/IP is used to connect to primary and standby databases.
- The standby database has been created from backups of the primary database control files and datafiles as described in the *Oracle Data Guard Concepts and Administration*
- The scenarios in this chapter assume the following broker configuration:
  - The configuration name is `DRSolution`.
  - The database unique name (`DB_UNIQUE_NAME`) of the primary database is `North_Sales`.
  - The database unique name (`DB_UNIQUE_NAME`) of the remote standby database is `DR_Sales`.
  - The protection mode is maximum performance mode.
  - There are no standby redo log files.
  - The standby database is a physical standby database.

6.2 Scenario 1: Creating a Configuration

This section provides examples that create a broker configuration named `DRSolution` that includes a primary and standby database named `North_Sales` and `DR_Sales`.

The following steps show how to create a configuration and add one physical standby database.

Step 1 Invoke the Data Guard CLI.

To start the CLI, enter DGMGRL at the command-line prompt on a system where Oracle Data Guard is installed:

```
% DGMGRL
DGMGRL for Solaris:  Version 10.1

Copyright (c) 2000, 2003, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL>
```

Step 2 Connect to the primary database.

Before you specify any command (other than the HELP, EXIT, or QUIT), you must first connect to the primary database using the DGMGRL CONNECT command.

The account from which you connect to the database (SYS in this example) must have SYSDBA privileges on the primary and standby databases.

> **Note:** You do not have to include AS SYSDBA on the CONNECT command because SYSDBA is the default setting for this command.

The following examples show two variations of the CONNECT command. Example 6–1 shows how to connect to the default database on the local system, and Example 6–2 includes the Oracle Net Services connect identifier (North_Sales.foo.com) to make a connection to a database located on a remote system.

Example 6–1 Connecting to the Primary Database on the Local System

```
DGMGRL> CONNECT sys/change_on_install;
Connected.
```

Example 6–2 Connecting to the Primary Database on a Remote System

```
DGMGRL> CONNECT sys/change_on_install@North_Sales.foo.com;
Connected.
```

Step 3 Create the broker configuration.

To create the broker configuration, you first define the configuration including a profile for the primary database, which in this case is called North_Sales. In a later command, you will add a profile for the standby database, DR_Sales.

Note: The names for the primary and standby databases must match their database unique names. Fetch these from their DB_UNIQUE_NAME initialization parameter as follows:

```
SQL> SHOW PARAMETER DB_UNIQUE NAME;
```

Use the CREATE CONFIGURATION command to create the DRSolution configuration and define the primary database, North_Sales:

```
DGMGRL> CREATE CONFIGURATION 'DRSolution' AS
>  PRIMARY DATABASE IS 'North_Sales'
>  CONNECT IDENTIFIER IS North_Sales.foo.com;
```

The CLI returns the following information:

```
Configuration "DRSolution" created with primary database "North_Sales"
```

The name North_Sales is the value of this database's DB_UNIQUE_NAME initialization parameter.

Step 4 Show the configuration information.

Use the SHOW CONFIGURATION command to display a brief summary of the configuration:

```
DGMGRL> SHOW CONFIGURATION;
```

The CLI returns the following information:

```
Configuration
  Name:            DRSolution
  Enabled:         NO
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database

Current status for "DRSolution":
DISABLED
```

Step 5 Add a standby database to the configuration.

To add a standby database to the DRSolution configuration, use the ADD DATABASE command to create a broker configuration profile for the standby database.

The following command defines DR_Sales as a standby database, which is the standby database associated with the primary database called North_Sales:

```
DGMGRL> ADD DATABASE 'DR_Sales' AS
>  CONNECT IDENTIFIER IS DR_Sales.foo.com
>  MAINTAINED AS PHYSICAL;
```

The CLI returns the following information:

```
Database "DR_Sales" added.
```

The name DR_Sales is the value of the database's DB_UNIQUE_NAME initialization parameter.

Use the SHOW CONFIGURATION command to verify that the DR_Sales database was added to the DRSolution configuration:

```
DGMGRL> SHOW CONFIGURATION;
```

The CLI returns the following information:

```
Configuration
  Name:            DRSolution
  Enabled:         NO
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
DISABLED
```

6.3 Scenario 2: Setting Database Properties

After you create the configuration with the CLI, you can set database properties at any time. For example, the following statements set the LogArchiveFormat and StandbyArchiveLocation properties for the DR_Sales standby database:

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'LogArchiveFormat'='log_%t_%s_%r_%d.arc';
Property "LogArchiveFormat" updated.

DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'StandbyArchiveLocation'='/archfs/arch/';
Property "StandbyArchiveLocation" updated.

DGMGRL> SHOW DATABASE VERBOSE 'DR_Sales';

Database
```

```
  Name:            DR_Sales
  Role:            PHYSICAL STANDBY
  Enabled:         NO
  Intended State:  OFFLINE
  Instance(s):
    dr_sales1

  Properties:
    InitialConnectIdentifier        = 'DR_Sales.foo.com'
    LogXptMode                      = 'ARCH'
    Dependency                      = ''
    DelayMins                       = '0'
    Binding                         = 'OPTIONAL'
    MaxFailure                      = '0'
    ReopenSecs                      = '300'
    AsyncBlocks                     = '2048'
    NetTimeout                      = '30'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '120'
    RealTimeApply                   = 'OFF'
    ApplyNoDelay                    = 'NO'
    ApplyNext                       = '0'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '5'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = 'dbs/t, dbs/s2t'
    LogFileNameConvert              = 'dbs/t, dbs/s2t'
    StatusReport                    = '(monitor)'
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    HostName                        = 'dr.foo.com'
    SidName                         = 'dr_sales1'
    LocalListenerAddress            = '(ADDRESS=(PROTOCOL=TCP)(HOST=dr.foo.com)(PORT=1514))'
    StandbyArchiveLocation          = '/archfs/arch/'
    AlternateLocation               = ''
    LogArchiveTrace                 = '4095'
    LogArchiveFormat                = 'log_%t_%s_%r_%d.arc'
    LatestLog                       = '(monitor)'
    TopWaitEvents                   = '(monitor)'

Current status for "DR_Sales":
DISABLED
```

These properties map directly to the LOG_ARCHIVE_FORMAT and STANDBY_ARCHIVE_DEST initialization parameters. If broker management of the database is enabled, setting a database property value causes the underlying parameter value to be changed in the corresponding database, and the value for the changed parameter is reflected in the server parameter file. Thus, if the database is shut down and restarted outside of the Data Guard GUI and the CLI (such as from the SQL*Plus interface), the database uses the new parameter values from the updated server parameter file when it starts. However, you should not make changes to the log transport services initialization parameters through SQL statements. Doing so will cause an inconsistency between the database and the broker.

Note: The database properties are typically displayed in mixed-case (for example, LogArchiveFormat) typeface to help you visually differentiate database properties (from the corresponding initialization parameter, SQL statement, or PL/SQL procedure), which are typically documented in UPPERCASE typeface.

You can change a property if the database is enabled or disabled. However, if the database is disabled when you change a property, the change does not take effect until the database is enabled.

6.4 Scenario 3: Enabling the Configuration and Databases

So far, the DRSolution configuration was disabled, which means it is not under the control of the Data Guard broker. When you finish configuring the databases into a broker configuration and setting any necessary database properties, you must enable the configuration to allow the Data Guard broker to manage it. This brings the primary and standby databases online.

You can enable:

- The entire configuration, including all of its databases
- A standby database

Enable the entire configuration.

You can enable the entire configuration, including all of the databases, with the following command:

```
DGMGRL> ENABLE CONFIGURATION;
```

```
Enabled.
```

Show the configuration.

Use the SHOW command to verify that the configuration and its databases were successfully enabled and brought online:

```
DGMGRL> SHOW CONFIGURATION;
```

The CLI returns the following information:

```
Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
SUCCESS
```

Enable the database.

This step is unnecessary except if the standby database was previously disabled with the DISABLE DATABASE command. Normally, enabling the configuration also enables the standby database.

```
DGMGRL> ENABLE DATABASE 'DR_Sales';
Enabled.
```

Show the database.

```
DGMGRL> SHOW DATABASE 'DR_Sales';

Database
  Name:            DR_Sales
  Role:            PHYSICAL STANDBY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    dr_sales1

Current status for "DR_Sales":
SUCCESS
```

6.5 Scenario 4: Setting the Configuration Protection Mode

You can change the protection mode of the configuration at any time. However, it is best if you do this when there is no activity occurring in the configuration.

> **Note:** If the protection mode to be set is higher than what is currently set in the configuration, the broker will automatically restart the primary database.

This scenario sets the protection mode of the configuration to the MAXPROTECTION mode. Note that this protection mode requires that there be at least one standby database configured to use standby redo log files.

Step 1 Configure standby redo log files, if necessary.

Because you will be setting the protection mode to the MAXPROTECTION mode, it is important to ensure that sufficient standby redo log files are configured on the standby database.

The Data Guard GUI provides the Standby Redo Log Assistant to configure standby redo log files automatically for you. If you are using the CLI, see *Oracle Data Guard Concepts and Administration* for information about creating standby redo log files.

Step 2 Set the LogXptMode property appropriately.

Use the EDIT DATABASE (property) command on the standby database to set the log transport mode that corresponds to the protection mode you plan to set. If the protection mode to be set is MAXPROTECTION, it is required that the log transport mode of at least one standby database is set to SYNC. For example:

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'LogXptMode'='SYNC';
Property "LogXptMode" updated.
```

The broker will not allow this command to succeed unless the standby database is configured with standby redo log files in the configuration.

Step 3 Change the overall protection mode for the configuration.

Use the EDIT CONFIGURATION command to upgrade the broker configuration to the MAXPROTECTION protection mode:

```
DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MAXPROTECTION;
Operation requires shutdown of instance "sales1" on database "North_Sales".
Shutting down instance "sales"...
```

```
Database closed.
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "sales1" on database "North_Sales".
Starting instance "sales1"...
ORACLE instance started.
Database mounted.
```

After you change the protection mode, the primary database instances automatically restart.

If the configuration is disabled when you enter this command, the actual protection mode change is not applied until you enable the configuration with the `ENABLE CONFIGURATION` command. The broker will not allow you to enable the configuration if it does not find a standby database in the configuration that can support the requirements of the protection mode.

Step 4 Verify the protection mode was changed.

Use the `SHOW CONFIGURATION` command to display the current protection mode for the configuration:

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxProtection
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
SUCCESS
```

See Also: Section 3.6, "Managing Data Protection Modes"

6.6 Scenario 5: Performing Routine Management Tasks

There may be situations in which you want to change the state or properties of the databases in a broker configuration to perform routine maintenance on one or more databases. You might also need to temporarily disable broker management of databases in a configuration.

6.6.1 Changing States and Properties

As you monitor the configuration, you might need to dynamically modify the states of the databases or their properties. The following sections show how to change the state or properties of the databases in the configuration.

6.6.1.1 Alter a Database Property

You can modify the values of database properties at any time—if the database is enabled, disabled, online, or offline.

Example 6–3 shows how to use the EDIT DATABASE command to change the LogArchiveTrace property to the value 127 for the North_Sales database.

Example 6–3 Altering a Database Property

```
DGMGRL> EDIT DATABASE 'North_Sales' SET PROPERTY 'LogArchiveTrace'='127';
```

The CLI returns the following message to indicate that the LogArchiveTrace property was updated successfully in the Data Guard configuration file:

```
Property "LogArchiveTrace" updated
```

If the configuration is currently disabled, the database does not use the new property value until you enable the broker configuration with the ENABLE CONFIGURATION command.

6.6.1.2 Alter the State of a Standby Database

You might want to use your physical standby database temporarily for reporting applications. To change the state of the standby database to read-only, enter the EDIT DATABASE command as shown in Example 6–4.

Example 6–4 Altering a Standby Database State

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET STATE='READ-ONLY';
Succeeded.
```

Log files are still being received when you put the physical standby database in the read-only state. The broker stops log apply services from applying the archived redo log files to the standby database.

6.6.1.3 Alter the State of a Primary Database

You might want to stop the transmittal of log files to the standby database. To change the state of the primary database to accommodate this, use the `EDIT DATABASE`
`North_Sales SET STATE=Log-Transport-Off` command. You can also set the primary database `OFFLINE`, which effectively shuts down the primary database and disables the broker from managing the configuration (shown in Example 6–5).

Example 6–5 Altering a Primary Database State

```
DGMGRL> EDIT DATABASE 'North_Sales' SET STATE='Offline';
```

The CLI returns the following message to indicate the command was successfully executed:

```
Operation requires shutdown of instance "sales1" on database
"North_Sales".
Shutting down instance "sales1"...
Database closed.
Database dismounted.
ORACLE instance shut down.
```

To change the primary state back to `ONLINE`, you must start the primary database.

6.6.2 Disabling the Configuration and Databases

When you disable the broker configuration or any of its databases, you are disabling the broker’s management of those objects and are effectively removing your ability to use the CLI to manage and monitor the disabled object. However, disabling the broker’s management of a broker configuration does not affect the actual operation of the underlying Data Guard configuration or the databases. For example, the log transport services and log apply services in the Data Guard configuration continue to function unchanged, but you can no longer manage them with the CLI.

In addition, disabling the broker’s management of an object *does not* remove or delete its profile from the broker configuration file. You can reenable your ability to use the CLI (or the Data Guard GUI) to manage the object by entering the appropriate `ENABLE CONFIGURATION` or `ENABLE DATABASE` command.

After you enter a `DISABLE CONFIGURATION` or `DISABLE DATABASE` command, the CLI returns the following message to indicate that the command successfully updated the Data Guard configuration file:

```
Disabled.
```

6.6.2.1 Disable a Configuration

You must use the `DISABLE CONFIGURATION` command to disable management of the entire broker configuration including the primary database as shown in Example 6–6.

Example 6–6 Disabling the Configuration and Primary Database

```
DGMGRL> DISABLE CONFIGURATION;
```

The only way to disable broker management of the primary database is to use the `DISABLE CONFIGURATION` command; the `DISABLE DATABASE` command only disables management of a standby database.

> **Note:** If you disable management of a configuration while connected to the standby database, you must connect to the primary database to reenable the configuration.

6.6.2.2 Disable a Standby Database

You use the `DISABLE DATABASE` command when you temporarily do not want the broker to manage and monitor a standby database.

You can explicitly disable broker management of a standby database to prevent it from being brought online when the rest of the configuration is brought online. Example 6–7 shows how to disable the `DR_Sales` standby database.

Example 6–7 Disabling a Standby Database

```
DGMGRL> DISABLE DATABASE 'DR_Sales';
```

> **Note:** To disable management of a primary database, you must use the `DISABLE CONFIGURATION` command.

> **Caution: If you disable broker management of a standby database in the broker configuration, that standby database cannot be used by the broker as a failover target in the event of loss of the primary database.**

When running in either the maximum protection or maximum availability mode, the broker prevents you from disabling the last standby database that supports the protection mode.

6.6.3 Removing the Configuration or a Standby Database

When you use either the `REMOVE CONFIGURATION` or `REMOVE DATABASE` command, you effectively delete the configuration or standby database profile from the broker configuration file, removing the ability of the Data Guard broker to manage the configuration or the standby database, respectively.

A remove operation *does not* remove or delete the actual Data Guard configuration underneath, nor does it affect the operation of the actual Data Guard configuration and its databases.

Caution: After you use the `REMOVE CONFIGURATION` or `REMOVE DATABASE` command, you cannot recover the configuration or database profile that was deleted from the broker configuration file. You must go through the steps in Section 6.2 as necessary, to create a broker configuration that can be managed with the CLI (or the Data Guard GUI).

Step 1 Remove a standby database from the configuration.

When you use the `REMOVE DATABASE` command, broker management and monitoring of the database ceases and the database's profile is deleted from the broker configuration file:

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
SUCCESS

DGMGRL> REMOVE DATABASE 'DR_Sales';
```

The CLI returns the following message to indicate the command successfully removed the `DR_Sales` database information from the Data Guard configuration file:

```
Removed database "DR_Sales" from the configuration.

DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database

Current status for "DRSolution":
SUCCESS
```

When running in either the maximum protection or maximum availability mode, the broker prevents you from deleting the last standby database that supports the protection mode.

Step 2 Remove the broker configuration.

Use the following command to remove the entire configuration from management and monitoring by the broker:

```
DGMGRL> REMOVE CONFIGURATION;
```

The CLI returns the following message to indicate the command successfully removed all of the configuration information from the Data Guard configuration file:

```
Removed configuration.

DGMGRL> SHOW CONFIGURATION;
Error: ORA-16532: Data Guard configuration does not exist

unable to describe configuration
```

6.7 Scenario 6: Performing a Switchover Operation

You can switch the role of the primary database and a standby database using the `SWITCHOVER` command. Before you issue the `SWITCHOVER` command, you must ensure:

- The state of the primary and standby databases are online.
- All participating databases are in good health, without any errors or warnings present.
- The standby database properties were set on the primary database, so that the primary database can function correctly when transitioning to a standby database (shown in the following examples in boldface type).
- Standby redo log files on the primary database are set up, if necessary.

See Also: *Oracle Data Guard Concepts and Administration*

Step 1 Check the primary database.

Use the `SHOW DATABASE VERBOSE` command to check the state, health, and properties of the primary database, as follows:

```
DGMGRL> SHOW DATABASE VERBOSE 'North_Sales';

Database
  Name:            North_Sales
  Role:            PRIMARY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    sales1

  Properties:
    InitialConnectIdentifier          = 'North_Sales.foo.com'
    LogXptMode                        = 'ARCH'
    Dependency                        = ''
    DelayMins                         = '0'
    Binding                           = 'OPTIONAL'
    MaxFailure                        = '0'
    ReopenSecs                        = '300'
    AsyncBlocks                       = '2048'
    NetTimeout                        = '30'
    LogShipping                       = 'ON'
    PreferredApplyInstance            = ''
    ApplyInstanceTimeout              = '120'
    RealTimeApply                     = 'OFF'
    ApplyNoDelay                      = 'NO'
    ApplyNext                         = '0'
    ApplyParallel                     = 'AUTO'
    StandbyFileManagement             = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '5'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = 'dbs/s2t, dbs/t'
    LogFileNameConvert              = 'dbs/s2t, dbs/t'
    StatusReport                    = '(monitor)'
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    HostName                        = 'north.foo.com'
    SidName                         = 'sales1'
    LocalListenerAddress            = '(ADDRESS=(PROTOCOL=TCP)(HOST=north.foo.com)(PORT=1514))'
    StandbyArchiveLocation          = '/archfs/arch/'
    AlternateLocation               = ''
    LogArchiveTrace                 = '4095'
    LogArchiveFormat                = 'r_%t_%s_%r.arc'
    LatestLog                       = '(monitor)'
    TopWaitEvents                   = '(monitor)'

Current status for "North_Sales":
SUCCESS
```

In particular, you should examine the boldface properties and the current status item, and some of the standby properties such as `StandbyArchiveLocation`, `DbFileNameConvert`, and `LogFileNameConvert`. See Chapter 3 for information about managing databases.

Step 2 Check the standby database that is the target of the switchover.

Use the `SHOW DATABASE VERBOSE` command to check the state, health, and properties of the standby database that is the target of the switchover. For example:

```
DGMGRL> SHOW DATABASE VERBOSE 'DR_Sales';

Database
  Name:            DR_Sales
  Role:            PHYSICAL STANDBY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    dr_sales1

  Properties:
    InitialConnectIdentifier        = 'DR_Sales.foo.com'
    LogXptMode                      = 'SYNC'
    Dependency                      = ''
    DelayMins                       = '0'
    Binding                         = 'OPTIONAL'
    MaxFailure                      = '0'
    ReopenSecs                      = '300'
    AsyncBlocks                     = '2048'
    NetTimeout                      = '30'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '120'
    RealTimeApply                   = 'OFF'
    ApplyNoDelay                    = 'NO'
    ApplyNext                       = '0'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '5'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = 'dbs/t, dbs/s2t'
    LogFileNameConvert              = 'dbs/t, dbs/s2t'
    StatusReport                    = '(monitor)'
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    HostName                        = 'dr.foo.com'
    SidName                         = 'dr_sales1'
    LocalListenerAddress            = '(ADDRESS=(PROTOCOL=TCP)(HOST=dr.foo.com)(PORT=1514))'
    StandbyArchiveLocation          = '/archfs/arch/'
    AlternateLocation               = ''
    LogArchiveTrace                 = '4095'
    LogArchiveFormat                = 'log_%t_%s_%r_%d.arc'
    LatestLog                       = '(monitor)'
    TopWaitEvents                   = '(monitor)'

Current status for "DR_Sales":
SUCCESS
```

In particular, you should examine the current status of the database.

Step 3 Issue the switchover command.

Issue the SWITCHOVER command to swap the roles of the primary and standby databases. The following example shows how the broker automatically shuts down and restarts the two participating databases as a part of the switchover. (See the usage notes in Section 7.1.3 for information about how to set up the broker environment so that the CLI can automatically restart the primary and standby databases for you.)

```
DGMGRL> SWITCHOVER TO "DR_Sales";
Performing switchover NOW. Please wait...
Operation requires shutdown of instance "sales1" on database
"North_Sales".
Shutting down instance "sales1"...
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
Operation requires shutdown of instance "dr_sales1" on database
"DR_Sales".
Shutting down instance "dr_sales1"...
database not mounted
ORACLE instance shut down.
Operation requires startup of instance "sales1" on database "North_Sales".
Starting instance "sales1"...
ORACLE instance started.
Database mounted.
Operation requires startup of instance "dr_sales1" on database "DR_Sales".
Starting instance "dr_sales1"...
ORACLE instance started.
Database mounted.
Switchover succeeded. New primary is "DR_Sales"
```

After the switchover completes, use the SHOW CONFIGURATION and SHOW DATABASE commands to verify that the switchover operation was successful.

Step 4 Show the configuration.

Issue the SHOW CONFIGURATION command to verify that the switchover was successful.

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxProtection
  Databases:
    North_Sales - Physical standby database
    DR_Sales    - Primary database

Current status for "DRSolution":
SUCCESS
```

6.8 Scenario 7: Performing a Failover Operation

You invoke a failover operation in response to an emergency situation, usually when the primary database cannot be accessed or is unavailable. See Section 4.2 before you fail over to decide which standby database should be the target of the failover. The following scenario describes a failover to the remote database called DR_Sales.

Step 1 Connect to the target standby database.

To perform the failover operation, you must connect to the standby database to which you want to fail over using the SYSDBA username and password of that database. For example:

```
DGMGRL> CONNECT sys/knl_test7@DR_Sales.foo.com
Connected.
```

Step 2 Issue the failover command.

Now you can issue the failover command to make the target standby database the new primary database for the configuration. Note that after the failover completes, the original primary database cannot be used as a viable standby database of the new primary database unless it is re-created as described in Section 4.2. The following example shows how the broker automatically shuts down and restarts the new primary database as a part of the failover. (See the usage notes in Section 7.1.3 for information about how to set up the broker environment so that the CLI can automatically restart the new primary database for you.)

```
DGMGRL> FAILOVER TO "DR_Sales";
Performing failover NOW. Please wait...
Operation requires shutdown of instance "dr_sales1" on database
"DR_Sales".
Shutting down instance "dr_sales1"...
database not mounted
ORACLE instance shut down.
Operation requires startup of instance "dr_sales1" on database "DR_Sales".
Starting instance "dr_sales1"...
```

```
ORACLE instance started.
Database mounted.
Failover succeeded. New primary is "DR_Sales"
```

Step 3 Show the configuration.

Issue the `SHOW CONFIGURATION` command to verify the failover.

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Physical standby database
    DR_Sales    - Primary database

Current status for "DRSolution":
SUCCESS
```

Step 4 Show the database.

Issue the SHOW DATABASE command to see that the old primary database was disabled by the broker as a consequence of the failover. It must be re-created as described in Section 4.2.5.

```
DGMGRL> SHOW DATABASE 'North_Sales';

Database
  Name:             North_Sales
  Role:             PHYSICAL STANDBY
  Enabled:          NO
  Intended State:   ONLINE
  Instance(s):
    sales1

Current status for "North_Sales":
Error: ORA-16795: Database resource guard detects that database reinstantiation is required
```

6.9 Scenario 8: Monitoring a Data Guard Configuration

The scenario in this section demonstrates how to use the `SHOW` command and monitorable database properties to identify and resolve a failure situation.

Step 1 Check the configuration status.

The status of the broker configuration is an aggregated status of all databases and instances in the broker configuration. You can check the configuration status first to determine whether or not any further action needs to be taken. If the configuration status is `SUCCESS`, everything in the broker configuration works fine. However, if you see the following warning, it means something is wrong in the configuration:

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
Warning: ORA-16607: one or more databases have failed
```

In this case, you need to continue on to Step 2 to determine the actual failure.

Step 2 Check the database status.

To identify which database has the failure, you need to go through all of the databases in the configuration one by one. In this example, the error happens to be on the primary database `North_Sales`:

```
DGMGRL> SHOW DATABASE 'North_Sales';
```

The command returns the following output:

```
Database
  Name:            North_Sales
  Role:            PRIMARY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    sales1

Current status for "North_Sales":
Error: ORA-16810: multiple errors or warnings detected for the database
```

Step 3 Check the monitorable property StatusReport.

When you see message ORA-16810, you can use the monitorable property `StatusReport` to identify each of the errors or warnings:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'StatusReport';
STATUS REPORT
       INSTANCE_NAME   SEVERITY ERROR_TEXT
              sales1      ERROR ORA-16737: The log transport service for
standby "DR_Sales" has an error.
             sales1    WARNING ORA-16714: The value of property
LogArchiveTrace is inconsistent with the database setting.
             sales1    WARNING ORA-16715: Log transport related property
ReopenSecs of standby "DR_Sales" is inconsistent.
```

Step 4 Check the monitorable property LogXptStatus.

You see error ORA-16737 in the previous status report in Step 3. To identify the exact log transport error, you can use monitorable property `LogXptStatus`:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'LogXptStatus';
LOG TRANSPORT STATUS
PRIMARY_INSTANCE_NAME STANDBY_DATABASE_NAME               STATUS
              sales1             DR_Sales ORA-12541: TNS:no listener
```

Now you know the exact reason why log transport services failed. To fix this error, start the listener for the physical standby database `DR_Sales`.

Step 5 Check the monitorable property InconsistentProperties.

You also see warning ORA-16714 reported in Step 3. To identify the inconsistent values for property `LogArchiveTrace`, you can use monitorable property `InconsistentProperties`:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'InconsistentProperties';

INCONSISTENT PROPERTIES
   INSTANCE_NAME   PROPERTY_NAME    MEMORY_VALUE    SPFILE_VALUE    BROKER_VALUE
          sales1   LogArchiveTrace           255            4095            4095
```

It seems that the current database memory value (255) is different from both the server parameter file (SPFILE) value (4095) and Data Guard broker's property value (4095). If you decide the database memory value is correct, you can update Data Guard broker's property value using the following command:

```
DGMGRL> EDIT DATABASE 'North_Sales' SET PROPERTY 'LogArchiveTrace'=255;
Property "LogArchiveTrace" updated.
```

In the previous command, Data Guard broker also updates the spfile value for you so that value for `LogArchiveTrace` is kept consistent.

Step 6 Check the monitorable property InconsistentLogXptProps.

Another warning you see in the status report returned in Step 3 is ORA-16715. To identify the inconsistent values for the log transport property, `ReopenSecs`, you can use monitorable property `InconsistentLogXptProps`:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'InconsistentLogXptProps';

INCONSISTENT LOG TRANSPORT PROPERTIES
   INSTANCE_NAME    STANDBY_NAME   PROPERTY_NAME    MEMORY_VALUE    BROKER_VALUE
          sales1        DR_Sales      ReopenSecs             600             300
```

The current database memory value (600) is different from the Data Guard broker's property value (300). If you think the broker's property value is correct, you can fix the inconsistency by re-editing the property of the standby database with the same value, as shown in the following example:

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'ReopenSecs'=300;
Property "ReopenSecs" updated.
```

You can also reenable the standby database or reset the primary database state to `ONLINE` to fix the inconsistency, but re-editing the property is the simplest.

7 Data Guard Command-Line Interface Reference

The Data Guard command-line interface enables you to manage a Data Guard broker configuration and its databases directly from the command line, or from batch programs or scripts. You can use the Data Guard command-line interface as an alternative to the Data Guard graphical user interface (GUI), integrated with Oracle Enterprise Manager, for managing a Data Guard configuration.

This chapter provides reference information for the Data Guard command-line interface.

7.1 Starting the Data Guard Command-Line Interface

To run the Data Guard command-line interface, you must have SYSDBA privileges.

Start the command-line interface by entering DGMGRL at the command-line prompt on a system where Oracle is installed:

```
% DGMGRL
DGMGRL for Solaris:  Version 10.1

Copyright (c) 2000, 2003 Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL>
```

7.1.1 DGMGRL Optional Parameters

You can supply optional parameters on the command line to indicate how you want the Data Guard command-line interface to display output such as command prompts, banners, and messages.

Additionally, a single command mode is available. In this mode, DGMGRL executes one command and exits upon the completion of the command. The exit code is the result of the command. If the exit code is 0, the command completed successfully. Otherwise, there was an error.

The command line of DGMGRL appears as follows:

```
% DGMGRL [<options>] [<logon> [<command>] ]
```

Specify none, one, or all of the following keywords when you invoke the DGMGRL command-line interface:

- `<options>` can be one of the following choices:
  - `-echo`

    Displays command input and output to the default display device. If you do not use this parameter, only the output from the command is displayed.
  - `-silent`

    Suppresses the display of the DGMGRL (`DGMGRL>`) command prompt on your default display device.This option is useful if you are directing the command output to a file or to another display tool.
- `<logon>` is:
  - `username/password [@connect-identifier]`

    The `username` and `password` with which you want to connect to the database. The `connect-identifier` is a fully specified connect descriptor or a name to be resolved by an Oracle naming method (for example, TNS). The `connect-identifier` is optional.
- `<command>` is a single command.

  For example:

  ```
  % DGMGRL sys/knl_test7@primary "show database db"
  ```

The following subsections specify the command format that you enter at the `DGMGRL>` command prompt.

7.1.2 DGMGRL Command Format and Parameters

The DGMGRL commands allow you to create and maintain one broker configuration at a time. A broker configuration can consist of a primary database and up to 9 standby databases.

After you invoke the command-line interface, you can enter any of the DGMGRL commands listed in Table 7–1. Each command and its associated parameters are described in detail in later sections of this chapter.

Table 7–1 Summary of DGMGRL Commands

Command	Effect
ADD DATABASE	Adds a new standby database profile to the existing broker configuration.
CONNECT	Connects to the specified database using the specified username.
CREATE CONFIGURATION	Creates a broker configuration and creates and adds a primary database profile to the configuration.
DISABLE CONFIGURATION	Disables broker management of a configuration so that the configuration and all of its databases are no longer managed by the broker.
DISABLE DATABASE	Disables broker management of the named standby database.
EDIT CONFIGURATION (Protection Mode)	Changes the current protection mode setting for the broker configuration.
EDIT DATABASE (Property)	Changes the value of a property for the named database.
EDIT DATABASE (Rename)	Changes the name used by the broker to refer to the specified database.
EDIT DATABASE (State)	Changes the state of the specified database.
EDIT INSTANCE (AUTO PFILE)	Sets the name of the initialization parameter file for the specified instance.
EDIT INSTANCE (Property)	Changes the value of a property for the specified instance.
ENABLE CONFIGURATION	Enables broker management of the broker configuration and all of its databases.
ENABLE DATABASE	Enables broker management of the specified database.
EXIT	Exits the Data Guard command-line interface.
FAILOVER	Performs a database failover operation in which the standby database, to which the CLI is currently connected, fails over to the role of primary database.
HELP	Displays online help for the Data Guard command-line interface.
QUIT	Quits the Data Guard command-line interface.

Table 7–1 (Cont.) Summary of DGMGRL Commands

Command	Effect
REMOVE CONFIGURATION	Removes the broker configuration including all of its database profiles from the broker configuration file.
REMOVE DATABASE	Removes the specified standby database profile from the broker configuration.
REMOVE INSTANCE	Removes knowledge of an instance from an existing database profile in the broker configuration.
SHOW CONFIGURATION	Displays information about the broker configuration.
SHOW DATABASE	Displays information about the specified database.
SHOW INSTANCE	Displays information about the specified instance.
SHUTDOWN	Shuts down a currently running Oracle database.
STARTUP	Starts an Oracle instance with the same options as SQL*Plus, including mounting and opening a database.
SWITCHOVER	Performs a switchover operation in which the current primary database becomes a standby database, and the specified standby database becomes the primary database.

Note: Existing Oracle9*i* command-line scripts are supported in Oracle Database 10*g* for non-RAC databases. See Appendix A for information about deprecated commands.

7.1.3 DGMGRL Command Usage Notes

To use the Data Guard command-line interface, the following must be true:

- The `DG_BROKER_START` dynamic initialization parameter is set to `true`.
- To enable broker operations that require restarting instances without manual intervention, Oracle Net Services must be configured on each of the hosts that contain the primary and standby database instances. Specifically, the listener.ora file must contain static configuration information about the instance. The `GLOBAL_DBNAME` attribute must be set to *`db_unique_name`*`_DGMGRL.`*`db_domain`*. See Chapter 6 for additional information.
- The net service name, if used, must be resolvable from any of the hosts in the configuration.

> **See Also:** Section 1.7.5 for information about the listener prerequisites. Chapter 6 for more information about preparing and starting Oracle Data Guard broker. See the *Oracle Database Administrator's Guide* for more information about setting up the network files and listener on the standby database.

- You must have SYSDBA privileges to use the Data Guard command-line interface. Do not include `AS SYSDBA` on the `CONNECT` command because SYSDBA is the default setting for this command.
- The password for SYS needs to be identical on all databases, and all databases should use the remote password file (either SHARED or EXCLUSIVE).

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

- A semicolon is required at the end of each DGMGRL command.
- Characters specified in a DGMGRL command string value are interpreted as lowercase characters, unless enclosed in double (") or single (') quotation marks. For example, database and DatAbaSe are equivalent, but "database" and "DatAbaSe" are distinctive.
- You can use the backslash (\) to escape a single quotation mark ('), a double quotation mark ("), and the backslash character (\) itself if these characters appear in a character string.
- Some operations on a broker configuration may require that one or more databases be shut down and restarted. In most cases, the CLI will automatically shut down and restart a given database for you if the following are true:
  - The *`instance-name`* is the SID (this applies for the GUI as well as the CLI).
  - The broker must be able to connect to the database using the username and password given in the last `CONNECT` command, even if the last `CONNECT` command was used to connect to another database. Thus, the remote password file for the database must contain the username and password given in the last `CONNECT` command.

> **See Also:** *Oracle Database Administrator's Guide* for more information about setting up remote password files and the default location of the PFILE and SPFILE initialization parameter files.

Command Examples

Example 1

This example demonstrates how to connect to the DGMGRL command-line interface on a local system.

```
% DGMGRL

Welcome to DGMGRL, type "help" for information.

DGMGRL> CONNECT sys/change_on_install;
Connected.
```

Example 2

This example demonstrates how to connect to the Data Guard (DGMGRL) command-line interface on a remote system.

```
DGMGRL> CONNECT sys/change_on_install@remote-stby;
Connected.
```

7.2 Stopping the Data Guard Command-Line Interface

When you are done working with the command-line interface and want to return to the operating system, enter the `EXIT` or `QUIT` command at the DGMGRL command prompt. For example:

```
DGMGRL> EXIT;
```

ADD DATABASE

Creates a new standby database profile and adds it to the existing broker configuration. The standby database type (physical or logical) is specified by the `MAINTAINED AS` clause.

Format

ADD DATABASE *database-name* AS

CONNECT IDENTIFIER IS *connect-identifier*

MAINTAINED AS {PHYSICAL | LOGICAL};

Command Parameters

database-name
The name that will be used by the broker to refer to this standby database. It must match (case-insensitive) the value of the corresponding database `DB_UNIQUE_NAME` initialization parameter.

connect-identifier
A fully specified connect descriptor or a name to be resolved by an Oracle Net Services naming method (for example, TNS).

Usage Notes

- You must connect to the primary database to issue this command.
- The broker uses the specified `connect-identifier` to communicate with the specified database from other databases. Therefore, you must ensure that the `connect-identifier` can be used to address the specified database from all databases in your configuration. For example, if you use tnsnames.ora files to resolve the `connect-identifier`, you must ensure it will be resolved to the same connect descriptor at all tnsnames.ora files and the resulting connect descriptor can be used to reach the database specified in this `ADD DATABASE` command.
- If the connection cannot be made, the broker does not add the new database to the configuration.

Command Example

The following example shows how to add a database named DR_Sales.

```
DGMGRL> ADD DATABASE  'DR_Sales' AS
> CONNECT IDENTIFIER IS DR_Sales.foo.com
> MAINTAINED AS PHYSICAL;
Database "DR_Sales" added.
```

CONNECT

Connects a given username to the specified database.

Format

CONNECT *username/password* [@*connect-identifier*];

Command Parameters

username/password
Represents the username and password with which you want to connect to the database.

connect-identifier
Consists of the Oracle Net Services connect identifier of the database to which you want to connect. The exact syntax depends upon the Oracle Net Services communications protocol your Oracle installation uses.

Usage Notes

- The username and password must be valid for the database to which you are trying to connect. The username you specify must have the SYSDBA privilege. You do not have to include `AS SYSDBA` on the `CONNECT` command because SYSDBA is the default setting for this command.
- If the `CONNECT` command returns an error, check to see that you specified a valid `connect-identifier`.

Command Examples

Example 1

This example connects to the default database on the local system.

```
DGMGRL> CONNECT sys/change_on_install;
Connected.
```

Example 2

This example connects to a remote database whose `connect-identifier` is `prmy`.

```
DGMGRL> CONNECT sys/change_on_install@prmy;
Connected.
```

CREATE CONFIGURATION

Creates a new broker configuration, and creates and adds a primary database profile to the configuration.

Format

CREATE CONFIGURATION *configuration-name* AS

PRIMARY DATABASE IS *database-name*

CONNECT IDENTIFIER IS *connect-identifier;*

Command Parameters

configuration-name
A user-friendly name for the configuration you are creating. Valid names contain any alphanumeric characters. If spaces are included in the name, the name must be enclosed in double or single quotation marks. The name must consist of 30 or fewer bytes.

database-name
The name that will be used by the broker to refer to the primary database. It must match (case-insensitive) the value of the corresponding database `DB_UNIQUE_NAME` initialization parameter.

connect-identifier
A fully specified connect descriptor or a name to be resolved by an Oracle Net Services naming method (for example, TNS).

Usage Notes

- A broker configuration is a named collection of one or more databases that you want to manage as a group. You must specify a value for each of the command parameters. There are no default values.
- You must connect to the primary database to issue this command.
- The broker uses the specified `connect-identifier` to communicate with the specified database from other databases. Therefore, you must ensure that the `connect-identifier` can be used to address the specified database from all databases in your configuration. For example, if you use tnsnames.ora files to

resolve the `connect-identifier`, you must ensure it will be resolved to the same connect descriptor at all tnsnames.ora files and the resulting connect descriptor can be used to reach the primary database specified in this `CREATE CONFIGURATION` command.

- To add standby databases after you create the broker configuration, use the ADD DATABASE command.

Command Example

The following example creates a new broker configuration named `DRSolution` with a primary database named `North_Sales`.

```
DGMGRL> CREATE CONFIGURATION 'DRSolution' AS
> PRIMARY DATABASE IS 'North_Sales'
> CONNECT IDENTIFIER IS North_Sales.foo.com;
Configuration "DRSolution" created with primary database "North_Sales".
```

DISABLE CONFIGURATION

Disables broker management of a configuration so that the configuration and all of its databases are no longer managed by the broker.

Format

```
DISABLE CONFIGURATION;
```

Command Parameters

None.

Usage Notes

- A disabled configuration and all of its constituent databases are no longer managed by the broker.
- The only way to disable broker management of the primary database is to use the `DISABLE CONFIGURATION` command.
- This command does not remove the broker configuration. See the REMOVE CONFIGURATION command for more information about removing the configuration.
- You can edit database properties and modify the configuration's protection mode while the configuration is disabled. However, any changes made to properties or to the protection mode will not take effect until the configuration is enabled.

Command Example

The following example disables management of the broker configuration and all of its databases.

```
DGMGRL> DISABLE CONFIGURATION;
Disabled.
```

DISABLE DATABASE

Disables broker management of the named standby database. This means that broker directed state changes will be disallowed for this database, and the broker will not monitor the database for health status or for monitorable property values.

Format

DISABLE DATABASE *database-name*;

Command Parameter

database-name
Name of the standby database to be disabled.

Usage Notes

- You cannot specify the name of a primary database.
- Use the `DISABLE CONFIGURATION` command to disable the primary and all standby databases.
- If the sole standby database is disabled, you have no failover option. This standby database is not viable for failover until it is reenabled.

Command Example

The following example shows how to disable a database named `DR_Sales`.

```
DGMGRL> DISABLE DATABASE 'DR_Sales';
Disabled.
```

EDIT CONFIGURATION (Protection Mode)

Edits the current protection mode setting for the broker configuration.

Format

EDIT CONFIGURATION SET PROTECTION MODE AS *protection-mode*;

Command Parameter

protection-mode
The data protection mode in which you want the configuration to run when the configuration is enabled. The possible protection modes are:

```
MAXPROTECTION
MAXAVAILABILITY
MAXPERFORMANCE
```

Usage Notes

- Before you use the `EDIT CONFIGURATION` command to set the protection mode to either the `MAXPROTECTION` or `MAXAVAILABILITY` mode, ensure that standby redo log files are configured on a standby database.
- The following table shows the configuration protection modes and the minimum corresponding settings for log transport services:

Protection Mode	Log Transport Mode	Standby Redo Log Files Needed?
`MAXPROTECTION`	`SYNC`	Yes
`MAXAVAILABILITY`	`SYNC`	Yes
`MAXPERFORMANCE`	`ARCH` or `ASYNC`	Yes for `ASYNC`

  The default protection mode for the configuration is `MAXPERFORMANCE`.

  See Also: Chapter 3 for more information about the protection modes and log transport modes

- After you change the protection mode, the primary database will automatically restart, if necessary.

- Use the SHOW CONFIGURATION command to display the current protection mode for the configuration.

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:             DRSolution
  Enabled:          YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
SUCCESS
```

If broker management of the configuration is disabled when you enter the EDIT CONFIGURATION command, the protection mode of the configuration does not take effect until the next time you enable the configuration with the ENABLE CONFIGURATION command.

Command Example

The following example shows how to upgrade the broker configuration to the MAXPROTECTION protection mode. The broker configuration will have the maximum amount of data protection after these commands complete.

Verify that standby redo log files are configured on the standby database and that the log transport mode is set to SYNC, for example:

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY 'LogXptMode'='SYNC';
Property "LogXptMode" updated.

DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MAXPROTECTION;
Operation requires shutdown of instance "sales1" on database "North_Sales".
Shutting down instance "sales"...
Database closed.
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "sales1" on database "North_Sales".
Starting instance "sales1"...
ORACLE instance started.
Database mounted.
```

The broker automatically stops and restarts the primary database.

EDIT DATABASE (Property)

Changes the value of a property for the named database.

Format

EDIT DATABASE *database-name*

SET PROPERTY *property-name* = *value*;

Command Parameters

database-name
The name of the database for which you want to change a property value.

property-name
The name of an existing database-specific property. If this is a RAC database, this property change affects all instances of the database.

> **See Also:** Chapter 3 and Chapter 8 for information about properties.

value
The new value for the property.

> **Caution:** This command can be used to change the value of a per-instance property if and only if just one instance is known by the broker for the named database. An attempt to use this command to change a per-instance property when the broker knows of multiple instances of the database will be rejected. It is recommended to only use `EDIT INSTANCE (property)` to change the value of a per-instance property.

Command Examples

Example 1

Edit a database level property.

```
DGMGRL> EDIT DATABASE 'North_Sales' SET PROPERTY 'ArchiveLagTarget'=1200;
```

```
Property "ArchiveLagTarget" updated.
```

Example 2

Edit an instance level property of a non-RAC database.

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET PROPERTY
> 'StandbyArchiveLocation'='/archfs/arch/';
Property "StandbyArchiveLocation" updated.
```

Example 3

Edit an instance level property of a RAC database. This will not succeed because it is not clear to which instance the property change should be applied.

```
DGMGRL> EDIT DATABASE 'North_Sales' SET PROPERTY
> 'StandbyArchiveLocation'='/archfs/arch/';
Error: ORA-16587: Ambiguous object specified to Data Guard broker

Failed.
```

EDIT DATABASE (Rename)

Changes the name used by the broker to refer to the specified database, as recorded in that database's profile in the broker configuration.

Format

EDIT DATABASE *database-name*

  RENAME TO *new-database-name*;

Command Parameters

database-name
The name of the database that you want to change.

new-database-name
The name of the new database.

Usage Notes

- Use this command to track changes to the `DB_UNIQUE_NAME` initialization parameter for this database.

  > **Caution: The `database-name` must always match the value for that database's `DB_UNIQUE_NAME` initialization parameter.**

- This command can only be done when broker management of the database that you are renaming is disabled.

Command Example

The following example shows how to edit and rename a database.

```
DGMGRL> EDIT DATABASE 'DR_Sales_typo' RENAME TO 'DR_Sales';
Succeeded.

DGMGRL> ENABLE DATABASE 'DR_Sales';
Enabled.
```

EDIT DATABASE (State)

Changes the state of the specified database.

Format

```
EDIT DATABASE database-name
  SET STATE = state
  [WITH APPLY INSTANCE = instance-name];
```

Command Parameters

database-name
The name of the database for which you want to change the state.

state
The state in which you want the database to be running. The possible states are:

`ONLINE`
`LOG-TRANSPORT-OFF` (primary database only)
`LOG-APPLY-OFF` (standby database only)
`READ-ONLY` (physical standby database only)
`OFFLINE`

instance-name
The name of the instance you want to become the apply instance if this is a RAC standby database.

Usage Notes

- If the target state is `ONLINE` and this database is currently in the standby role, the optional `WITH APPLY INSTANCE` clause specifies which instance will become the apply instance.
- If the target state is not `ONLINE` or if the database is currently in the primary role, the `WITH APPLY INSTANCE` clause is ignored even if it is specified.
- All instances of a RAC database are affected by this database state change.

Command Example

The following example shows how to change the state of a database.

```
DGMGRL> EDIT DATABASE 'DR_Sales' SET STATE='READ-ONLY';
Succeeded.

DGMGRL> EDIT DATABASE 'North_Sales' SET STATE='OFFLINE';
Operation requires shutdown of instance "sales1" on database "North_Sales".
Shutting down instance "sales1"...
Database closed.
Database dismounted.
ORACLE instance shut down.
```

EDIT INSTANCE (AUTO PFILE)

Sets the name of the initialization parameter file for the specified instance.

Format

EDIT INSTANCE *instance-name*

[ON DATABASE *database-name*]

SET AUTO PFILE [= { *initialization-file* | OFF }];

Command Parameters

instance-name
The name of the instance (SID) for which you want to specify its initialization parameter file.

database-name
The name of the database to which the `instance-name` is associated.

initialization-file
Executes the startup operation for the instance when a subsequent broker operation requires the instance to be started automatically. If `SET AUTO PFILE` is set to `OFF`, automatic restart of that instance is disabled. When a subsequent operation needs to start that instance, you must start it manually. If you do not specify `SET AUTO PFILE` for the instance, the automatic startup operation looks for the initialization parameter file at the default location.

Usage Notes

- The *`instance-name`* can be unique across the configuration. If *`instance-name`* is not unique, you must specify both the *`database-name`* and the *`instance-name`* to fully identify the instance.
- `SET AUTO PFILE` is valid only for the duration of the current DGMGRL session. You must specify `SET AUTO PFILE` again if you quit and reenter DGMGRL.

Command Example

The following example shows how to edit an instance of a database.

```
DGMGRL> EDIT INSTANCE 'dr_sales1' ON DATABASE 'DR_Sales'
> SET AUTO PFILE='initsales1.ora';
Instance 'dr_sales1' updated.
```

EDIT INSTANCE (Property)

Changes the value of a property for the specified instance.

Format

EDIT INSTANCE *instance-name*

[ON DATABASE *database-name*]

SET PROPERTY *property-name* = *value*;

Command Parameters

instance-name
The name of the instance (SID) for which you want to change a per-instance property value.

database-name
The name of the database to which the `instance-name` is associated.

property-name
The name of the per-instance property for which you want to set a new value.

> **See Also:** Chapter 3 and Chapter 8 for information about properties.

value
The new value for the property.

Usage Notes

- The *`instance-name`* can be unique across the configuration. If *`instance-name`* is not unique, you must specify both the *`database-name`* and the *`instance-name`* to fully identify the instance.
- This command cannot be used to change a database-specific property.

Command Examples

Example 1

Edit an instance level property.

```
DGMGRL> EDIT INSTANCE 'sales1' ON DATABASE 'North_Sales'
> SET PROPERTY 'StandbyArchiveLocation'='/archfs/arch/';
Property "StandbyArchiveLocation" updated.
```

Example 2

Edit a database level property. This will not be allowed.

```
DGMGRL> EDIT INSTANCE 'sales1' ON DATABASE 'North_Sales'
> SET PROPERTY 'LogXptMode'='SYNC';
Error: ORA-16586: Could not edit database property through instance.

Failed.
```

ENABLE CONFIGURATION

Enables the broker to actively manage the broker configuration including all of its databases.

Format

ENABLE CONFIGURATION;

Command Parameters

None.

Usage Notes

- Use this command to enable broker management of the primary database.
- By default, broker management of the configuration's databases is enabled in the `ONLINE` state with log transport services turned on at the primary database and log apply services turned on at the standby databases. You can change the state of any database using the EDIT DATABASE (State) command, but not when the database or the entire configuration is disabled.
- Use the SHOW CONFIGURATION command to display information about the configuration.

Command Example

The following example enables management of a broker configuration.

```
DGMGRL> ENABLE CONFIGURATION;
Enabled.
```

ENABLE DATABASE

Enables broker management of the specified standby database.

> **Caution:** Do not issue the `ENABLE DATABASE` command on a standby database that needs to be re-created until it has been re-created as described in Section 4.2.5.

Format

ENABLE DATABASE *database-name*;

Command Parameter

database-name
The name of the standby database for which you want to enable broker management.

Usage Notes

- A standby database may have been disabled by the broker as a consequence of a prior failover or switchover operation. Recovery of the database is required as described in Section 4.2.5 before this command should be issued.
- By default, broker management of the standby database is enabled in the `ONLINE` state with log apply services enabled. You can change the state of the standby database using the EDIT DATABASE (State) command, but only when the database is enabled.
- Use the SHOW DATABASE command to display information about the database.
- For a RAC database, only one instance is required to be started and mounted for this command to succeed.

Command Example

The following example shows how to enable a database named `DR_Sales`.

```
DGMGRL> ENABLE DATABASE 'DR_Sales';
Enabled.
```

EXIT

Exits (quits) the command-line interface.

Format

EXIT;

Command Parameters

None.

Usage Notes

- This command has the same effect as the QUIT command.
- A database connection is not required to execute this command. However, if you are connected, this command breaks the connection.

Command Example

The following example demonstrates how to exit (quit) the command-line interface.

```
DGMGRL> EXIT;
```

FAILOVER

A failover operation changes the named standby database into the role of a primary database.

> **Note:** Because a failover results in a role transition that may result in the loss of application data, you should perform a failover *only* if the primary database failed. If you want the current primary database and a standby database to switch roles, then use the SWITCHOVER command.

Format

FAILOVER TO *database-name*

  [IMMEDIATE];

Command Parameters

database-name

The name of the standby database you want to fail over to the primary database role.

Usage Notes

- To be considered a viable candidate for the failover operation, the specified standby database must be enabled before the primary fails. However, an enabled standby database that was taken offline can be a candidate for the failover operation. In this case, restart the standby database using the CLI `STARTUP` command, then issue the `FAILOVER` command.
- The failover operates on the specified database. Thus, the failover changes one of the standby databases into the role of a primary database. Any other standby databases not involved in the failover remain in the standby role.
- Before you issue the `FAILOVER` command, verify that you are connected to the standby database that will become the new primary database. If necessary, issue a CONNECT command to connect to the standby database.

- If the FAILOVER command is issued without any options, the standby database chosen as the failover target recovers all log files received before changing to the primary role.
- If the standby database that you want to fail over to the primary role is a RAC database, the broker will shut down all of the instances except the apply instance before it continues the failover operation. The broker will restart instances that it shut down prior to the failover. See Section 4.2 for details.
- If the standby database that is transitioning into the role of primary database is a physical standby database, then the database instance (or instances) will be restarted after the failover completes. If the database is a logical standby database, the database instance (or instances) does not need to be restarted.
- If the broker configuration is in either MAXPROTECTION or MAXAVAILABILITY protection mode, the failover operation will force the protection mode to be MAXPERFORMANCE. The log transport mode settings are unaffected. You need to restore the desired protection mode for the resulting configuration after the failover operation.
- If the FAILOVER command is issued with the IMMEDIATE option, no attempt is made to recover available log files. This option more likely results in lost application data even when standby redo log files are configured on the standby database. Additionally, any remaining standby databases in the configuration cannot function as such until they are re-created.
- The original primary database can only participate in the configuration as a standby database after it is re-created.

Caution: You should shut down the original primary database if it still has any active instances running.

See Also: Section 4.2.5 about re-creating the original primary database so that it could serve as a standby database to the primary database

Command Examples

The following example performs a failover in which the standby database, DR_Sales, transitions to the primary role:

```
DGMGRL> FAILOVER TO "DR_Sales";
Performing failover NOW. Please wait...
```

```
Operation requires shutdown of instance "dr_sales1" on database "DR_Sales".
Shutting down instance "dr_sales1"...
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "dr_sales1" on database "DR_Sales".
Starting instance "dr_sales1"...
ORACLE instance started.
Database mounted.
Failover succeeded. New primary is "DR_Sales"
```

HELP

Displays online help for the Data Guard command-line interface.

Format

HELP [*topic*];

Command Parameter

topic
The topic for which you want to display help information. If you do not specify a topic, the command lists all of the topics and the format. Valid topics are:

```
ADD
CONNECT
CREATE
DISABLE
EDIT
ENABLE
EXIT
FAILOVER
HELP
QUIT
REMOVE
SHOW
SHUTDOWN
STARTUP
SWITCHOVER
```

Usage Note

- A database connection is not required to execute this command.

Command Examples

Example 1

The following examples get help on the `HELP` and `CONNECT` commands.

```
DGMGRL> HELP HELP;
```

```
Display description and syntax for a given command

Syntax:

  HELP [<command>];

DGMGRL> HELP CONNECT;

Connect to an Oracle instance

Syntax:

  CONNECT <username>/<password>[@<connect identifier>]
```

Example 2

The following example gets help on the `EDIT` commands.

```
DGMGRL> HELP EDIT

Edit a configuration, database or instance

Syntax:

  EDIT CONFIGURATION SET PROTECTION MODE AS
    {MaxProtection|MaxAvailability|MaxPerformance};

  EDIT DATABASE <database name> SET PROPERTY <property name> = <value>;

  EDIT DATABASE <database name> RENAME TO <new database name>;

  EDIT DATABASE <database name> SET STATE = <state>
    [WITH APPLY INSTANCE = <instance name>];

  EDIT INSTANCE <instance name> [ON DATABASE <database name>]
    SET AUTO PFILE [ = {<initialization file path>|OFF} ];

  EDIT INSTANCE <instance name> [ON DATABASE <database name>]
    SET PROPERTY <property name> = <value>;
```

QUIT

Quits (exits) the Data Guard command-line interface.

Format

QUIT;

Command Parameters

None.

Usage Notes

- This command has the same effect as the EXIT command.
- A database connection is not required to execute this command. However, if you are connected, this command breaks the connection.

Command Example

The following example shows how to quit (exit) the command-line interface.

```
DGMGRL> QUIT;
```

REMOVE CONFIGURATION

Removes all of the broker configuration information, including all database profiles, from the Data Guard configuration file, and terminates broker management of all of the databases associated with the broker configuration.

> **Caution:** When you use the `REMOVE CONFIGURATION` command, all profile information is deleted from the Data Guard configuration file and cannot be recovered.

Format

REMOVE CONFIGURATION [PRESERVE DESTINATIONS];

Command Parameters

None.

Usage Notes

- When you remove a broker configuration, management of all of the databases associated with that configuration is disabled.
- By default, the command removes the corresponding broker settings of the `LOG_ARCHIVE_DEST_`*n* initialization parameter on the primary database and the `LOG_ARCHIVE_CONFIG` initialization parameters on all databases in the configuration. To preserve these settings, use the `PRESERVE DESTINATIONS` option.
- This command does not remove or affect the actual primary or standby database instances, databases, datafiles, control files, initialization parameter files, server parameter files, or log files of the underlying Data Guard configuration.

Command Example

The following example shows how to remove configuration information from the configuration file.

```
DGMGRL> REMOVE CONFIGURATION;
Removed configuration.
```

```
DGMGRL> SHOW CONFIGURATION;
Error: ORA-16532: Data Guard configuration does not exist
```

REMOVE DATABASE

Removes the specified standby database's profile from the broker configuration and terminates broker management of the standby database.

> **Caution:** When you use the `REMOVE DATABASE` command, the database's profile information is deleted from the broker configuration file and cannot be recovered.

Format

REMOVE DATABASE *database-name* [PRESERVE DESTINATIONS];

Command Parameter

database-name
The name of the standby database whose profile you want to remove from the broker configuration.

Usage Note

- An error is returned if you specify the name of the primary database in the broker configuration.
- By default, the command removed the corresponding broker settings of the `LOG_ARCHIVE_DEST_`*n* initialization parameter on the primary database and the `LOG_ARCHIVE_CONFIG` initialization parameter on all databases in the configuration. To preserve these settings, use the `PRESERVE DESTINATIONS` option.

Command Example

The following example shows how to remove a database from the Data Guard configuration.

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
SUCCESS

DGMGRL> REMOVE DATABASE 'DR_Sales';
Removed database "DR_Sales" from the configuration.

DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database

Current status for "DRSolution":
SUCCESS
```

REMOVE INSTANCE

Removes an instance from an existing database profile in the broker configuration.

Format

REMOVE INSTANCE *instance-name*

  [ON DATABASE *database-name*];

Command Parameters

instance-name
The name of the instance (SID) that you want to remove from the broker configuration.

database-name
The name of the database to which the `instance-name` is associated.

Usage Notes

- In a RAC database, the broker automatically adds started instances into the corresponding database profile. However, the broker may not automatically remove instances from the database profile. The `REMOVE INSTANCE` command can be used to manually remove any instance that no longer exists from the database profile.
- The *`instance-name`* can be unique across the configuration. If *`instance-name`* is not unique, you must specify both the *`database-name`* and the *`instance-name`* to fully identify the instance.
- This command is rejected for an instance that is currently active in the broker configuration.
- This command is rejected if this is the only instance currently associated with a database profile.

Command Example

The following example shows how to remove an instance of the database.

```
DGMGRL> REMOVE INSTANCE 'dr_sales3' ON DATABASE 'DR_SALES' ;
Removed instance "dr_sales3" from the database "DR_SALES".
```

SHOW CONFIGURATION

Displays a summary and status of the broker configuration. The summary lists all databases included in the broker configuration and other information pertaining to the broker configuration itself.

Format

SHOW CONFIGURATION;

Command Parameters

None.

Usage Notes

None.

Command Examples

The following example provides a summary of the DRSolution configuration.

```
DGMGRL> SHOW CONFIGURATION;

Configuration
  Name:            DRSolution
  Enabled:         YES
  Protection Mode: MaxPerformance
  Databases:
    North_Sales - Primary database
    DR_Sales    - Physical standby database

Current status for "DRSolution":
SUCCESS
```

SHOW DATABASE

Displays information or property values of the specified database and its instances.

Format

SHOW DATABASE [VERBOSE] *database-name* [*property-name*];

Command Parameters

database-name
The name of the database for which you want to display information.

property-name
The name of the property for which you want to display a value.

> **See Also:** Chapter 3 and Chapter 8 for information about properties.

Usage Notes

- The `SHOW DATABASE` command shows a brief summary of the database. `SHOW DATABASE VERBOSE` shows properties of the database in addition to the brief summary. They both show the status of the database.
- The `SHOW DATABASE VERBOSE` command shows database-specific properties and instance-specific properties. For a non-RAC database, the values of the instance-specific properties are those of the only instance of the database. For a RAC database, the values of the instance-specific properties will not be shown, although the property names are still listed. To see the instance-specific values of these properties, use the `SHOW INSTANCE` command.
- The properties that the `SHOW DATABASE VERBOSE` command shows depend on the database role and the configuration composition:
  - For the primary database, properties specific to physical standby databases are shown only if there is at least one physical standby database in the configuration. The properties specific to logical standby databases are shown only if there is at least one logical standby database in the configuration.

  - For physical standby databases, properties specific to logical standby databases are not shown.
  - For logical standby databases, properties specific to physical standby databases are not shown.
- This command is rejected if you use `SHOW DATABASE` *`property-name`* command to show an instance-specific property in a RAC database.

Command Examples

Example 1

Shows database information in an abbreviated format.

```
DGMGRL> SHOW DATABASE 'DR_Sales';

Database
  Name:            DR_Sales
  Role:            PHYSICAL STANDBY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    dr_sales1

Current status for "DR_Sales":
SUCCESS
```

Example 2

Shows database information in an extended format.

```
DGMGRL> SHOW DATABASE VERBOSE 'North_Sales';

Database
  Name:            North_Sales
  Role:            PRIMARY
  Enabled:         YES
  Intended State:  ONLINE
  Instance(s):
    sales1

  Properties:
    InitialConnectIdentifier        = 'North_Sales.foo.com'
    LogXptMode                      = 'ARCH'
    Dependency                      = ''
    DelayMins                       = '0'
    Binding                         = 'OPTIONAL'
    MaxFailure                      = '0'
    ReopenSecs                      = '300'
    AsyncBlocks                     = '2048'
    NetTimeout                      = '30'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '120'
    RealTimeApply                   = 'OFF'
    ApplyNoDelay                    = 'NO'
    ApplyNext                       = '0'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '5'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = 'dbs/s2t, dbs/t'
    LogFileNameConvert              = 'dbs/s2t, dbs/t'
    StatusReport                    = '(monitor)'
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    HostName                        = 'north.foo.com'
    SidName                         = 'sales1'
    LocalListenerAddress            = '(ADDRESS=(PROTOCOL=TCP)(HOST=north.foo.com)(PORT=1514))'
    AlternateLocation               = ''
    StandbyArchiveLocation          = '/archfs/arch/'
    LogArchiveTrace                 = '4095'
    LogArchiveFormat                = 'r_%t_%s_%r_%d.arc'
    LatestLog                       = '(monitor)'
    TopWaitEvents                   = '(monitor)'

Current status for "North_Sales":
SUCCESS
```

SHOW INSTANCE

Displays information or property value of the specified instance.

Format

SHOW INSTANCE [VERBOSE] *instance-name* [*property-name*]

[ON DATABASE *database-name*];

Command Parameters

instance-name
The name of the instance for which you want to display information.

property-name
The name of the property for which you want to display a value.

> **See Also:** Chapter 3 and Chapter 8 for information about properties.

database-name
The name of the database to which is associated the instance for which you want to show information.

Usage Notes

- The `SHOW INSTANCE` command shows a brief summary of the instance. `SHOW INSTANCE VERBOSE` shows properties of the instance in addition to the brief summary. They both show the status of the instance.
- The `SHOW INSTANCE VERBOSE` command only shows instance-specific properties.
- The properties that the `SHOW INSTANCE VERBOSE` command shows depend on the database role and the configuration composition:
  - For instances of the primary database, properties specific to physical standby instances are shown only if there is at least one physical standby database in the configuration. The properties specific to logical standby instances are shown only if there is at least one logical standby database in the configuration.

  - For instances of physical standby databases, properties specific to logical standby instances are not shown.
  - For instances of logical standby databases, properties specific to physical standby instances are not shown.
- The *instance-name* can be unique across the configuration. If *instance-name* is not unique, you must specify both the *database-name* and the `instance-name` to fully identify the instance.

Command Example

Example 1

The following example shows information about a specific instance of a database.

```
DGMGRL> SHOW INSTANCE sales1;

Instance 'sales1' of database 'North_Sales'
  Host Name:       north.foo.com

Current status for "sales1":
SUCCESS
```

Example 2

Shows instance information in an extended format.

```
DGMGRL> SHOW INSTANCE VERBOSE 'sales1';

Instance 'sales1' of database 'North_Sales'
  Host Name:       north.foo.com
  PFILE:

  Properties:
    HostName                        = 'north.foo.com'
    SidName                         = 'sales1'
    LocalListenerAddress            ='(ADDRESS=(PROTOCOL=TCP)(HOST=north.foo.com)(PORT=1514))'
    StandbyArchiveLocation          = '/archfs/arch/'
    AlternateLocation               = ''
    LogArchiveTrace                 = '4095'
    LogArchiveFormat                = 'r_%d_%t_%s_%r.arc'
    LatestLog                       = '(monitor)'
    TopWaitEvents                   = '(monitor)'
```

```
Current status for "sales1":
SUCCESS
```

SHUTDOWN

Shuts down a currently running Oracle instance.

Format

SHUTDOWN [ABORT | IMMEDIATE | NORMAL];

Command Parameters

None.

Usage Notes

- Using the `SHUTDOWN` command with no arguments is equivalent to using the `SHUTDOWN NORMAL` command.
- The following list describes the options to the `SHUTDOWN` command:
  - `ABORT`

    Proceeds with the fastest possible shutdown of the database without waiting for calls to complete or for users to disconnect from the database. Uncommitted transactions are not rolled back. Client SQL statements being processed are terminated. All users connected to the database are implicitly disconnected, and the next database startup will require instance recovery. You must use this option if a background process terminates abnormally.
  - `IMMEDIATE`

    Does not wait for current calls to complete or users to disconnect from the database. Further connections are prohibited. The database is closed and dismounted. The instance is shut down, and no instance recovery is required on the next database startup.
  - `NORMAL`

    This is the default option. The process waits for users to disconnect from the database. Further connections are prohibited. The database is closed and dismounted. The instance is shut down, and no instance recovery is required on the next database startup.

Command Example

The following command shuts down the database in normal mode.

```
DGMGRL > SHUTDOWN;

Database closed.
Database dismounted.
Oracle instance shut down.
```

STARTUP

Starts an Oracle database instance with any of the following options:

- FORCE: shuts down the current Oracle instance in the SHUTDOWN ABORT mode before restarting it.
- RESTRICT: allows only Oracle users with the RESTRICTED SESSION system privilege to connect to the instance.
- PFILE: specifies the PFILE initialization parameter file to be used when the database instance is started.
- MOUNT: mounts the specified database without opening it. If you do not specify a database name, the database name is taken from the initialization parameter DB_NAME.
- OPEN: mounts and opens the specified database.
- NOMOUNT: starts the specified database instance without mounting the database.

Format

```
STARTUP
  [FORCE]
  [RESTRICT]
  [PFILE=filename]
  [MOUNT [db_name] | OPEN [open-options] [db_name] | NOMOUNT];
```

Command Parameters

filename
The name of the initialization parameter file to be used when starting the database instance. If you do not specify the PFILE parameter option, then the default server parameter file (specific to your operating system) is used.

db_name
The name of the database to mount or open. If you do not specify the *db_name* parameter, the database name is taken from the initialization parameter DB_NAME.

open-options

The mode of access in which you want the specified database to start. The possible modes are:

```
READ ONLY
READ WRITE
```

Usage Notes

- Using the `STARTUP` command with no arguments is equivalent to using the `STARTUP OPEN` command.
- If you do not use the `FORCE` clause when you use the `STARTUP` command and the current database instance is running, an error results. The `FORCE` clause is useful when you are debugging or when error conditions are occurring. Otherwise, it should not be used.
- Use the `RESTRICT` clause to allow only Oracle users with the `RESTRICTED SESSION` system privilege to connect to the instance. Later, you can use the `ALTER SYSTEM` command through SQL*Plus to disable the restricted session feature.
- If you do not use the `PFILE` clause to specify the initialization parameter file, the `STARTUP` command uses the default server parameter file, if it exists. Otherwise, the `STARTUP` command uses the default initialization parameter file. The default files are platform specific.

  See your operating system-specific documentation for more information about the default parameter files.
- Use the `MOUNT` clause to mount a primary database or a logical standby database without opening it. If you do not specify a database name, the database name is taken from the initialization parameter `DB_NAME`.
- Use the `OPEN` clause to mount and open the specified database.
- The `NOMOUNT` clause starts the database instance without mounting the database. You cannot use the `NOMOUNT` clause with the `MOUNT` or `OPEN` options.

Command Examples

Example 1

The following examples show two different methods for starting a database instance. Each command starts a database instance using the standard parameter file, mounts the default database in exclusive mode, and opens the database.

```
DGMGRL> STARTUP;
DGMGRL> STARTUP OPEN database;
```

Example 2

The following command shuts down the current instance, immediately restarts it without mounting or opening the database, and allows only users with restricted session privileges to connect to it.

```
DGMGRL > STARTUP FORCE RESTRICT NOMOUNT;
```

Example 3

The following command starts an instance using the parameter file `testparm` without mounting the database.

```
DGMGRL > STARTUP PFILE=testparm NOMOUNT;
```

Example 4

The following example starts and mounts a database instance, but does not open it.

```
DGMGRL> STARTUP MOUNT;
```

SWITCHOVER

A switchover operation is a planned transition in which the primary database exchanges roles with one of the standby databases. When you issue the `SWITCHOVER` command, the current primary database becomes a standby database, and the specified standby database becomes the primary database.

Format

SWITCHOVER TO *database-name*;

Command Parameter

database-name
The name of the standby database you want to change to the primary database role.

Usage Notes

- The broker verifies that the primary and standby databases are in the following states before starting the switchover:
  - The primary database must be enabled and `ONLINE`, with log transport services started.
  - The standby database must be enabled and `ONLINE`, with log apply services started.
- The broker allows the switchover to proceed as long as there are no log transport services errors for the standby database that you selected to participate in the switchover. However, errors occurring for any other standby database not involved in the switchover will not prevent the switchover from proceeding.
- If the broker configuration is in either `MAXPROTECTION` or `MAXAVAILABILITY` mode, the switchover maintains the protection mode even after the operation (see Section 4.1.1). The switchover will not be allowed if the mode cannot be maintained because the target standby database of the switchover was the only standby that satisfied the protection mode requirement.
- If the standby database that is assuming the primary role is a physical standby database, then both the primary and standby databases will be restarted after

the switchover completes. If the standby database is a logical standby database, then neither the primary database nor the logical standby database is restarted.

- If the standby database that is assuming the primary role is a physical standby database, then the original primary becomes a physical standby database. Otherwise, it becomes a logical standby database.
- If the primary database is a RAC database, the broker will keep only one instance running and shut down all other instances before it continues the switchover. If the standby database you want to switch to the primary role is a RAC database, the broker will shut down all instances except the apply instance before it continues the switchover. The broker will restart instances that it shut down prior to the switchover. See Section 4.1 for details.
- If the standby database that is assuming the primary role is a logical standby database and there are physical standby databases in the configuration, after the switchover, the physical standby databases will not be able to participate as such until they are re-created.

Caution: For this reason, Oracle generally recommends that you specify your physical standby database for switchover instead of your logical standby database. If you must switch over to your logical standby database, see Section 4.2.5 to re-create your physical standby database.

Command Examples

Example 1

The following example shows a successful switchover in which the standby database, `DR_Sales`, transitions into the primary role.

```
DGMGRL> SWITCHOVER TO "DR_Sales";
Performing switchover NOW. Please wait...
Operation requires shutdown of instance "sales1" on database "North_Sales".
Shutting down instance "sales1"...
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
Operation requires shutdown of instance "dr_sales1" on database "DR_Sales".
Shutting down instance "dr_sales1"...
ORA-01109: database not open
```

```
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "sales1" on database "North_Sales".
Starting instance "sales1"...
ORACLE instance started.
Database mounted.
Operation requires startup of instance "dr_sales1" on database "DR_Sales".
Starting instance "dr_sales1"...
ORACLE instance started.
Database mounted.
Switchover succeeded. New primary is "DR_Sales"
```

Example 2

If you connect to the database using operating system authentication, you can use any username and password to connect. However, the CLI may not be able to shut down and start up the primary and standby database automatically because it cannot remotely authenticate itself.

The following example shows a switchover that succeeded but returns an error because the CLI cannot shut down and start up the primary and standby databases.

```
DGMGRL> CONNECT /
Connected

DGMGRL> SWITCHOVER TO "DR_Sales";
Performing switchover NOW. Please wait...
Operation requires shutdown of instance "sales1" on database "North_Sales".
Shutting down instance "sales1"...
ORA-01017: invalid username/password; logon denied

You are no longer connected to ORACLE
Please connect again.
Unable to shut down instance "sales1".
You must shut down instance "sales1" manually.
Operation requires shutdown of instance "dr_sales1" on database "DR_Sales".
You must shut down instance "dr_sales1" manually.
Operation requires startup of instance "sales1" on database "North_Sales".
You must start instance "sales1" manually.
Operation requires startup of instance "dr_sales1" on database "DR_Sales".
You must start instance "dr_sales1" manually.
Switchover succeeded. New primary is "DR_Sales"
```

Note: For DGMGRL to restart instances automatically, you must connect to the database as SYSDBA using the username and password you specified in the remote password file before you begin the switchover. The username and password must be the same on the primary and standby databases.

You must manually issue the SHUTDOWN and STARTUP commands to restart the new primary and standby database instances in this configuration.

8 Database Properties

Database properties help you to view and control the behavior of databases, log transport services, and log apply services in a broker configuration. This chapter provides the following sections about the monitorable and configurable properties:

- Section 8.1, "Monitorable (Read-Only) Database Properties"
- Section 8.2, "Configurable Database Properties"

The scope of some properties is said to be database-wide. If the database (primary or standby) is a RAC database consisting of multiple instances, the value of such a property applies uniformly across all of the instances sharing that database. The scope of other properties is said to be instance-specific. Such a property exists for all instances of the RAC database, but its value may differ from one specific instance to another.

> **Note:** This chapter presents properties primarily from the point of view of the DGMGRL CLI. Using the CLI, the properties described in this chapter may be viewed or modified using discrete CLI commands.
>
> The Data Guard GUI explicitly presents some of these properties on the Edit Properties page. Information from other properties may be implicitly incorporated into other pages displayed by the GUI. Each property's description in this chapter indicates how the GUI presents that property to the user.

8.1 Monitorable (Read-Only) Database Properties

Monitorable properties allow you to view information related to the database or the instance, but you cannot change the values of these properties. You can view all of the monitorable properties using CLI `SHOW` commands.

> **Note:** Information for monitorable properties can be seen only when broker management of the database is enabled and the database is in an online state. The Data Guard GUI displays the information obtained from these properties on the Property page.

If the database is a RAC database, the output values of some properties may also show instance-specific information. For example if the primary database is a RAC database, `LogXptStatus` may show `Instance1` transmitting redo data to `Standby2` has an error and `Instance2` transmitting redo data to `Standby4` has an error.

The following sections describe the database monitorable properties:

- InconsistentLogXptProps (Inconsistent Log Transport Properties)
- InconsistentProperties (Inconsistent Database Properties)
- LatestLog
- LogXptStatus (Log Transport Status)
- LsbyFailedTxnInfo (Logical Standby Failed Transaction Information)
- LsbyParameters (Logical Standby Parameters)
- LsbySkipTable (Logical Standby Skip Table)
- LsbySkipTxnTable (Logical Standby Skip Transaction Table)
- RecvQEntries (Receive Queue Entries)
- SendQEntries (Send Queue Entries)
- StatusReport (Status Report)
- TopWaitEvents

8.1.1 InconsistentLogXptProps (Inconsistent Log Transport Properties)

The `InconsistentLogXptProps` property returns a table that shows all properties related to log transport services whose values are inconsistent between the broker configuration file and the runtime value in the database.

The properties reported in this table can be either database-specific properties or instance-specific properties. A database-specific property only ensures that there is one value in the broker's configuration file for all instances sharing the database, but the runtime values among the instances can be different. This means that a database-specific property may be inconsistent only on some instances.

This property pertains to the primary database. The table contains the following columns:

- `INSTANCE_NAME`

  The value matching the SID for the instance.

- `STANDBY_NAME`

  The database unique name (`DB_UNIQUE_NAME`) of the standby database to which this log transport services property pertains.

- `PROPERTY_NAME`

  The name of the log transport services property with an inconsistent value.

- `MEMORY_VALUE`

  The runtime value being used in the database.

- `BROKER_VALUE`

  The value of the log transport services property saved in the broker configuration file.

8.1.2 InconsistentProperties (Inconsistent Database Properties)

The `InconsistentProperties` property returns a table that shows all database properties whose values contained in the broker configuration file are inconsistent with the values in the corresponding server parameter file or the runtime values in the database.

The properties reported in this table can be either database-specific properties or instance-specific properties. A database-specific property only ensures that there is one value in the broker's configuration file for all instances sharing the database, but the runtime memory values or SPFILE values among the instances can be different. This means that a database-specific property may be inconsistent only on some instances.

Each individual database has this property. The table contains the following columns:

- `INSTANCE_NAME`

  The value matching the SID for the instance.

- `PROPERTY_NAME`

  The name of the database property with the inconsistent value.

- `MEMORY_VALUE`

  The corresponding runtime value being used in the database.

- `SPFILE_VALUE`

  The corresponding value saved in the server parameter file (SPFILE).

- `BROKER_VALUE`

  The value of the database property saved in the broker configuration file.

8.1.3 LatestLog

Specifies the last 20 lines in the Data Guard broker log file of the specified instance. The `LatestLog` property is an instance level monitorable property. The table name is the full path of the Data Guard broker log file on the host on which the instance is running. Each instance in the configuration has this property. The table contains the following columns in the order shown:

- Line No.

  The line number of the log in the Data Guard broker log file.

- Timestamp

  The time when the log is generated.

- LogText

  The detailed text of the log.

For example, the following example shows output from a `SHOW INSTANCE` command:

```
DGMGRL> SHOW INSTANCE sales1 'LatestLog' /home/oracle10g/log/drcsales1.log
  Line No.            Timestamp             Log Text
```

```
      431  2003-06-09-15:51:28 DMON: Entered rfm_release_chief_lock for CTL_GET_STATUS
      432  2003-06-09-15:51:28 DMON: Releasing lock
      433  2003-06-09-15:52:18 DMON: ENUM_DRC: success. (len = 610)
      434  2003-06-09-15:52:18 DMON: ENUM_DRC operation completed
      435  2003-06-09-15:52:27 DMON: Entered rfm_get_chief_lock() for CTL_GET_STATUS, reason 0
      436  2003-06-09-15:52:27 DMON: acquiring lock 105 in 5 mode
      437  2003-06-09-15:52:27 DMON: chief lock convert for healthcheck
      438  2003-06-09-15:52:27 DMON: status from rfi_post_instances() = ORA-00000
      439  2003-06-09-15:52:27 INSV: message for req_id 1.1.496252113, opcode CTL_GET_STATUS,
phase 1, flags 5
      440  2003-06-09-15:52:27 DMON: Entered rfmhcexinst
      441  2003-06-09-15:52:27 DMON: rfmhcexinst calling RSMs
      442  2003-06-09-15:52:28 DMON: Setting db status after H/C aggregation
      443  2003-06-09-15:52:28 INSV: message for req_id 1.1.496252113, opcode CTL_GET_STATUS,
phase 5, flags 5
      444  2003-06-09-15:52:28 DMON: Setting db status after H/C aggregation
      445  2003-06-09-15:52:28 INSV: message for req_id 1.1.496252113, opcode CTL_GET_STATUS,
phase 3, flags 65541
      446  2003-06-09-15:52:28 DMON: Entered rfm_release_chief_lock for CTL_GET_STATUS
      447  2003-06-09-15:52:28 DMON: Releasing lock
      448  2003-06-09-15:52:43 DMON: ENUM_DRC: success. (len = 610)
      449  2003-06-09-15:52:43 DMON: ENUM_DRC operation completed
      450  2003-06-09-15:52:43 RSM 0 received GETPROP request: rid=0x01010001, pid=11
```

This property supplants the obsolete `SHOW LOG` command in Oracle 9iR2.

8.1.4 LogXptStatus (Log Transport Status)

The `LogXptStatus` property returns a table that contains the error status of log transport services for each of the enabled standby databases. This property pertains to the primary database.

The table contains the following columns:

- `PRIMARY_INSTANCE_NAME`

  The value matching the SID for the instance on the primary database.

- `STANDBY_DATABASE_NAME`

  The database unique name (`DB_UNIQUE_NAME`) of the standby database.

- `ERROR`

  The text of the log transport error. If there is no error, the field is empty.

Each entry in the table indicates the status of log transport services on one primary instance to one standby database.

The format of the error status is as follows:

```
"standby1_sitename=error_status, standby2_sitename=error_status,..."
```

The error status can be an empty string, which indicates there is no error.

In the following example, the STATUS from DR_Sales is empty because there is no error for the DR_Sales destination. The South_Report destination returned the ORA-01034 message.

```
DGMGRL> SHOW DATABASE 'North_Sales' 'LogXptStatus' ;
LOG TRANSPORT STATUS
PRIMARY_INSTANCE_NAME STANDBY_DATABASE_NAME               STATUS
             sales1            DR_Sales
             sales1        South_Report      ORA-01034: ORACLE not available
```

8.1.5 LsbyFailedTxnInfo (Logical Standby Failed Transaction Information)

The LsbyFailedTxnInfo property identifies a failed transaction that caused log apply services to stop. This property contains a string with the following values from the DBA_LOGSTDBY_EVENTS view:

- XIDUSN: Transaction ID undo segment number
- XIDSLT: Transaction ID slot number
- XIDSQN: Transaction ID sequence number
- STATUS_CODE: Status (or Oracle error code) belonging to the STATUS message
- STATUS: Description of the current activity of the process or the reason why log apply services stopped

The transaction IDs and status information are separated by a string of number signs (###). This property pertains to a logical standby database.

8.1.6 LsbyParameters (Logical Standby Parameters)

The LsbyParameters property contains a string that identifies the value of MAX_SGA (maximum system global area) and MAX_SERVERS (maximum number of parallel query servers) specifically reserved for log apply services. These values are separated by a string of number signs (###) in the LsbyParameters property.

This property pertains to a logical standby database.

8.1.7 LsbySkipTable (Logical Standby Skip Table)

The LsbySkipTable property lists the logical apply skip specifications. These skip specifications specify filters for log apply services to skip applying a certain class of online redo log files on the logical standby database. This property returns a table with the following columns from the DBA_LOGSTDBY_SKIP view:

- ERROR

  Indicates if the statement should be skipped (Y) or if errors should be returned for the statement (N)

- STATEMENT_OPT

  Indicates the type of statement that should be skipped

- SCHEMA

  The schema name for which this skip option should be used

- NAME

  Name of the object for which this skip option should be used

- PROCEDURE

  Name of the stored procedure to execute when processing the skip option

8.1.8 LsbySkipTxnTable (Logical Standby Skip Transaction Table)

The LsbySkipTxnTable lists the skip settings chosen. This property returns a table with following columns:

- XIDUSN: Transaction ID undo segment number
- XIDSLT: Transaction ID slot number
- XIDSQN: Transaction ID sequence number
- ACTIVE: Description of the current activity of the process or the reason why log apply services stopped

This property pertains to a logical standby database.

8.1.9 RecvQEntries (Receive Queue Entries)

The RecvQEntries property returns a table indicating all log files that were received by the standby database but have not yet been applied. If no rows are returned, it implies all log files received have been applied. This property pertains to a standby database.

The table contains the following columns in the order shown:

- STATUS

  The STATUS column is set to one of the following values for a log file on a logical standby database:

  - NOT_APPLIED: No redo records in this log file have been applied.
  - PARTIALLY_APPLIED: Some of the redo records in this log file have been applied while others have not.
  - COMMITTED_TRANSACTIONS_APPLIED: This status value only applies to a logical standby database. All redo records belonging to the committed transactions have been applied. Redo records belonging to uncommitted transactions have not been read by logminer and may still be needed when the transactions are committed in the future. Therefore, it is not safe yet to discard this online redo log file.

- RESETLOGS_ID

  Resetlogs identifier associated with the archived redo log file

- THREAD

  The redo thread number

- LOG_SEQ

  The online redo log file sequence number

- TIME_GENERATED

  The first time when the online redo log file was written to the primary database

- TIME_COMPLETED

  The next time when the log file was archived on the primary database (corresponds to the NEXT_CHANGE# column)

- FIRST_CHANGE#

  First change number in the archived redo log file

- NEXT_CHANGE#

  First change in the next log file

- SIZE (KBs)

The SIZE of the online redo log file in kilobytes

For example:

```
DGMGRL> SHOW DATABASE 'DR_Sales' 'RecvQEntries' ;
              STATUS     RESETLOGS_ID           THREAD
LOG_SEQ          TIME_GENERATED       TIME_COMPLETED    FIRST_CHANGE#
NEXT_CHANGE#        SIZE (KBs)
         NOT_APPLIED        497198843                1                    5
06/20/2003 14:55:38  06/20/2003 16:31:26           202138
210718             7364
         NOT_APPLIED        497198843                1                    6
06/20/2003 16:31:26  06/20/2003 16:31:39           210718
210753               13
         NOT_APPLIED        497198843                1                    7
06/20/2003 16:31:39  06/20/2003 16:31:54           210753
210758                1
         NOT_APPLIED        497198843                1                    8
06/20/2003 16:31:54  06/20/2003 16:31:59           210758
210789               11
```

Note: In the Data Guard GUI, this information is displayed on the Log File Details page.

8.1.10 SendQEntries (Send Queue Entries)

The SendQEntries property returns a table that shows all log files on the primary database that were not successfully archived to one or more standby databases. This property pertains to the primary database.

The table contains the following columns:

- STANDBY_NAME

  The value can be empty or it can contain the database unique name (DB_UNIQUE_NAME) of the standby database. If empty, the STATUS column will contain a value of CURRENT or NOT_ARCHIVED.

- STATUS

  The STATUS column is set to one of the following values:

  – CURRENT: A log file to which online redo is currently being written.

  - NOT_ARCHIVED: A completed online redo log file that has not been archived locally.
  - ARCHIVED: A completed log file that has been archived locally but has not been transmitted to the standby database specified in the STANDBY_NAME column.

  The table contains exactly one row with the value of STATUS=CURRENT. There can be multiple rows with the value STATUS=ARCHIVED or STATUS=NOT_ARCHIVED.

- RESETLOGS_ID

  Resetlogs identifier associated with the archived redo log file

- THREAD

  The redo thread number.

- LOG_SEQ

  The log sequence number. Multiple rows may have the same LOG_SEQ value (for different STANDBY_NAME values).

- TIME_GENERATED

  The first time when the online redo log file was written to the primary database.

- TIME_COMPLETED

  The next time when the log file was archived on the primary database (corresponds to the NEXT_CHANGE# column).

- FIRST_CHANGE#

  First change number in the archived redo log file.

- NEXT_CHANGE#

  First change in the next log file.

- SIZE (KBs)

  The SIZE of the online redo log file in kilobytes.

For example, the following shows output from a SHOW DATABASE command:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'SendQEntries' ;
PRIMARY_SEND_QUEUE
        STANDBY_NAME       STATUS     RESETLOGS_ID
THREAD              LOG_SEQ       TIME_GENERATED       TIME_COMPLETED
FIRST_CHANGE#     NEXT_CHANGE#       SIZE (KBs)
```

```
       reportdbade52     ARCHIVED        497198843
1                    9  06/20/2003 16:31:59  06/20/2003 16:39:57
210789          211411              186
       reportdbade52     ARCHIVED        497198843
1                   10  06/20/2003 16:39:57  06/20/2003 16:40:01
211411          211415                1
       reportdbade52     ARCHIVED        497198843
1                   11  06/20/2003 16:40:01  06/20/2003 16:40:07
211415          211418                1
                          CURRENT        497198843
1                   12  06/20/2003 16:40:07
211418                                1
```

Note: In the Data Guard GUI, this information is displayed on the Log File Details page.

8.1.11 StatusReport (Status Report)

The StatusReport property returns a table that provides a list of errors or warnings about the status of the database. In a RAC database environment, it also includes the status of all running instances. Each individual database has this property. The table contains the following columns in the order shown:

- INSTANCE_NAME

  The value matching the SID for the instance.

- ERROR_TEXT

  Formatted error text.

- SEVERITY

  The severity of the error message. The value is either WARNING or ERROR.

For example, the following shows output from a SHOW DATABASE command:

```
DGMGRL> SHOW DATABASE 'North_Sales' 'StatusReport' ;
STATUS REPORT
       INSTANCE_NAME   SEVERITY ERROR_TEXT
             sales1      ERROR ORA-16737: The log transport service for
standby "DR_Sales" has an error.
                   *      ERROR ORA-16745: unable to add DB_UNIQUE_NAME DR_Sales
into The DG_CONFIG list because it is full
```

8.1.12 TopWaitEvents

Specifies the 5 events with the longest waiting time in the specified instance. The events and their waiting time are retrieved from V$SYSTEM_EVENT. Each instance in the configuration has this property. This property is an instance level monitorable property. The table contains the following columns in the order shown:

- Event

  The system wait event.

- Wait Time

  The total amount of time waited for this event in hundredths of a second.

> **See Also:** See *Oracle Database Reference* for detailed explanations of all system wait events.

The following example shows output from a SHOW INSTANCE command:

```
DGMGRL> show instance dg10 'TopWaitEvents';
TOP SYSTEM WAIT EVENTS
               Event            Wait Time
   rdbms ipc message               671350
SQL*Net message from client         62390
          pmon timer                47897
  Queue Monitor Wait                43016
 wakeup time manager                38508
```

8.2 Configurable Database Properties

Configurable database properties control the behavior of databases in a broker configuration. You can view and dynamically update the values of these properties using either the CLI or the Data Guard GUI. However, some properties can only be updated through the CLI.

In most cases, the configurable database property is said to have **database scope**; meaning the value you set for the property applies uniformly to each instance of the database. However, in a few cases, the configurable database property is said to have **instance scope**; meaning, for a multiple-instance database environment, it is possible that the values of some particular properties may differ from one instance of the database to the next. Table 8–1 lists each configurable database property and indicates if the scope of the property is database-wide or instance-specific. If the Scope column contains:

- Database—The value of the property is database wide, not instance specific.
- Instance—The value of the property is instance specific, not database wide.

Table 8–1 Configurable Properties

Configurable Property Name	Scope	Pertains To
AlternateLocation	Instance	Log transport services
ApplyInstanceTimeout	Database	Log apply services
ApplyNext	Database	Log apply services for physical standby databases
ApplyNoDelay	Database	Log apply services for physical standby databases
ApplyParallel	Database	Log apply services for physical standby databases
ArchiveLagTarget	Database	Log transport services
AsyncBlocks	Database	Log transport services
Binding	Database	Log transport services
DbFileNameConvert	Database	Log transport services
DelayMins	Database	Log apply services
Dependency	Database	Log transport services
HostName	Instance	Instance identification
InitialConnectIdentifier	Database	Broker communication
LocalListenerAddress	Instance	Broker communication
LogArchiveFormat	Instance	Log transport services
LogArchiveMaxProcesses	Database	Log transport services
LogArchiveMinSucceedDest	Database	Log transport services
LogArchiveTrace	Instance	Diagnosis
LogFileNameConvert	Database	Log transport services
LogShipping	Database	Log transport services
LogXptMode	Database	Log transport services
LsbyASkipCfgPr	Database	Log apply services (SQL redo) for logical standby databases

Table 8–1 (Cont.) Configurable Properties

Configurable Property Name	Scope	Pertains To
LsbyASkipErrorCfgPr	Database	Log apply services (SQL redo) for logical standby databases
LsbyASkipTxnCfgPr	Database	Log apply services (SQL redo) for logical standby databases
LsbyDSkipCfgPr	Database	Log apply services (SQL redo) for logical standby databases
LsbyDSkipErrorCfgPr	Database	Log apply services (SQL redo) for logical standby databases
LsbyDSkipTxnCfgPr	Database	Log apply services (SQL redo) for logical standby databases
LsbyMaxEventsRecorded	Database	Log apply services (SQL redo) for logical standby databases
LsbyMaxSga	Instance	Log apply services (SQL redo) for logical standby databases
LsbyMaxServers	Instance	Log apply services (SQL redo) for logical standby databases
LsbyRecordAppliedDdl	Database	Log apply services (SQL redo) for logical standby databases
LsbyRecordSkipDdl	Database	Log apply services (SQL redo) for logical standby databases
LsbyRecordSkipErrors	Database	Log apply services (SQL redo) for logical standby databases
LsbyTxnConsistency	Database	Log apply services (SQL redo) for logical standby databases
MaxFailure	Database	Log transport services
NetTimeout	Database	Log transport services
PreferredApplyInstance	Database	Log apply services
RealTimeApply	Database	Log apply services
ReopenSecs	Database	Log transport services
SidName	Instance	Instance identification
StandbyArchiveLocation	Instance	Log transport services

Table 8–1 (Cont.) Configurable Properties

Configurable Property Name	Scope	Pertains To
StandbyFileManagement	Database	Log apply services

See Also: Chapter 3 for more information about database property management

Note: When a broker configuration with its primary database is created and standby databases are added to the configuration, the broker imports existing settings from the databases to set many of the properties. If importing an existing setting fails, or if a property value is not imported, then the broker uses a broker default value. The default values and whether or not a property is imported is indicated within each property description.

8.2.1 AlternateLocation

Specifies an alternate disk location to store the archived redo log files in the standby when the location specified by the `StandbyArchiveLocation` fails. The property has an instance scope, and the location it specifies has to be accessible by the instance.

Category	Description
Datatype	String
Valid values	Directory specification on system where the standby instance is located
Broker default	Empty string
Imported?	No
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical or logical

Category	Description
Corresponds to ...	- On the standby instance, the LOCATION attribute for the LOG_ARCHIVE_DEST_*n* initialization parameter that represents an alternate destination of the local destination that matches the property StandbyArchiveLocation - On the primary database, the TEMPLATE attribute for the LOG_ARCHIVE_DEST_*n* initialization parameter that represents an alternate destination
Scope	Instance
GUI name	Alternate Standby Location

[1] Although this property is set for the standby instance, it is indirectly related to log transport services for the primary database. The broker sets up both an alternate local destination on the standby instance and an alternate remote destination on the primary database.

> **Note:** On a logical standby database, Oracle recommends the LOCATION attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter for the local destination be different from the value of AlternateLocation property.

8.2.2 ApplyInstanceTimeout

Specifies the number of seconds the broker waits after detecting the current apply instance failed and before initiating the apply instance failover.

Category	Description
Datatype	Integer
Valid values	>=0 (seconds)
Broker default	120
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Physical or logical
Corresponds to ...	Not applicable
Scope	Database

Category	Description
GUI name	Not applicable

8.2.3 ApplyNext

Specifies the number of archived redo log files that log apply services should apply immediately to the physical standby database, temporarily overriding any apply delay interval previously specified by the `DelayMins` property. The `ApplyNext` property value is applied only at the point when you explicitly specify that value. Once the value is applied, the property no longer has any effect until the next time that its value is explicitly specified.

Specifying a value for this property has no effect and will be ignored if broker management of the standby database is disabled or if log apply services are offline at the time that a value is specified. Specifying a zero value for this property always has no effect on the current behavior of log apply services.

The last value specified for this property may be displayed using the CLI `SHOW` command.

Category	Description
Datatype	Integer
Valid values	`>=0` log files (no action taken if zero specified)
Broker default	`0`
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Physical
Corresponds to ...	`NEXT` *`n`* clause of the `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE` statement
Scope	Database
GUI name	Not applicable

8.2.4 ApplyNoDelay

Specifies whether or not to cancel the effect of the `DelayMins` property that was set on the standby database:

- If log apply services are online and you set ApplyNoDelay=YES, log apply services apply the archived redo log files as soon as they are archived to the standby database. This property is equivalent to using the following SQL statements.

  On a physical standby database:

  ```
  ALTER DATABASE RECOVER MANAGED STANDBY DATABASE NODELAY;
  ```

  On a logical standby database:

  ```
  ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
  ```

- If log apply services are online and you set ApplyNoDelay=NO, log apply services respect the delay settings specified by the DelayMins property of the standby database. This property is equivalent to using the following SQL statements.

  On a physical standby database:

  ```
  ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DEFAULT DELAY;
  ```

  On a logical standby database:

  ```
  EXECUTE DBMS_LOGSTDBY.APPLY_UNSET('APPLY DELAY');
  ```

- If log apply services are offline, then setting the property has no immediate effect. However, when log apply services are online again, the value of the property is used to determine the apply delay behavior of log apply services.

Category	Description
Datatype	String
Valid values	YES or NO
Broker default	NO
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Physical or logical

Category	Description
Corresponds to ...	▪ YES corresponds to the NODELAY clause of the ALTER DATABASE RECOVER MANAGED STANDBY DATABASE statement ▪ NO corresponds to the DEFAULT DELAY clause of the ALTER DATABASE RECOVER MANAGED STANDBY DATABASE statement ▪ ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE statement
Scope	Database
GUI name	Not applicable

The value of the ApplyNoDelay property persists through role changes. For example, if the ApplyNoDelay property is set to YES and the database undergoes a series of switchover operations, transitioning the database from the standby role to the primary role and then back again, the ApplyNoDelay property will continue to be set to YES throughout all of the role changes.

8.2.5 ApplyParallel

Specifies the number of concurrent processes log apply services can use on the physical standby database for Redo Apply. If log apply services are offline, then setting the property has no immediate effect. However, when log apply services are online again, the value of the property is used to determine the parallel apply behavior of log apply services.

Category	Description
Datatype	String
Valid values	▪ AUTO—the number of parallel processes used for Redo Apply is automatically determined by Oracle based on the number of CPUs that the system has. ▪ NO—no parallel apply. ▪ '2', '3', and so on—manually specify the number of parallel processes used for Redo Apply. (Specifying '0' is the same as specifying NO; specifying '1' is the same as specifying AUTO.)
Broker default	AUTO
Imported?	No

Category	Description
Parameter class	Not applicable
Role	Standby
Standby type	Physical
Corresponds to ...	■ AUTO corresponds to the PARALLEL clause of the ALTER DATABASE RECOVER MANAGED STANDBY DATABASE statement ■ NO corresponds to the NOPARALLEL clause of the ALTER DATABASE RECOVER MANAGED STANDBY DATABASE statement ■ '2','3', and so on corresponds to the PARALLEL *n* clause of the ALTER DATABASE RECOVER MANAGED STANDBY DATABASE statement
Scope	Database
GUI name	Not applicable

8.2.6 ArchiveLagTarget

Limits the amount of data that can be lost and effectively increases the availability of the standby database by forcing a log switch after the amount of time you specify (in seconds) elapses. That way, the standby database will not miss redo records generated from a time range longer than the value set for the ARCHIVE_LAG_TARGET initialization parameter.

Category	Description
Datatype	Number
Valid values	Seconds (either 0 seconds, or any number from 60 to 7200 seconds)
Broker default	0 (disabled)
Imported?	Yes, from the ARCHIVE_LAG_TARGET initialization parameter
Parameter class	Dynamic
Role	Primary
Standby type	Not applicable
Corresponds to ...	ARCHIVE_LAG_TARGET=*seconds* initialization parameter
Scope	Database
GUI name	Archive Lag Target

8.2.7 AsyncBlocks

Specifies the size of the SGA buffer to be used when network I/O operations are to be done asynchronously using the log writer process (LGWR). The value you set for `AsyncBlocks` property takes effect only when the `LogXptMode` property is set to `ASYNC`.

Category	**Description**
Datatype	Integer
Valid values	0 to 1,024,000 blocks
Broker default	61,440
Imported?	Yes, from the `ASYNC_BLOCKS` column of the `V$ARCHIVE_DEST` view of the primary database
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical and logical
Corresponds to ...	▪ `ASYNC` attribute for the `LOG_ARCHIVE_DEST_`*n* initialization parameter of the primary database ▪ `ASYNC_BLOCKS` column of the `V$ARCHIVE_DEST` view of the primary database
Scope	Database
GUI name	Not applicable

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the `LOG_ARCHIVE_DEST_`*n* value of the primary database.

> **See Also:** *Oracle Data Guard Concepts and Administration* for information about the `ASYNC` attribute of the `LOG_ARCHIVE_DEST_`*n* initialization parameter

8.2.8 Binding

Specifies whether or not the standby destination is `MANDATORY` or `OPTIONAL`.

Category	**Description**
Datatype	String

Category	Description
Valid values	- MANDATORY You can specify a policy for reuse of online redo log files using the MANDATORY value. If the archiving operation of a mandatory destination fails, online redo log files cannot be overwritten. - OPTIONAL You can specify a policy for reuse of online redo log files using the OPTIONAL value. If the archiving operation of an optional destination fails, the online redo log files are overwritten.
Broker default	OPTIONAL
Imported?	Yes, from the BINDING column of the V$ARCHIVE_DEST view of the primary database
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical and logical
Corresponds to ...	- MANDATORY and OPTIONAL attributes for the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database - BINDING column of the V$ARCHIVE_DEST view of the primary database
Scope	Database
GUI name	Not applicable

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

8.2.9 DbFileNameConvert

Distinguishes standby datafile filenames from primary datafile filenames. You must set this property on all standby databases. If you add a datafile to the primary database, this property converts the datafile name on the primary database to the datafile on the standby database.

This property is used in the following situations:

- At standby mount time, it is used to rename primary datafile filenames to standby datafile filenames if the datafile file path on the standby system is different from the primary database system.

- When a new datafile is created on the primary database, a corresponding new datafile will be created on the standby database if the `StandbyFileManagement` property is set to `'AUTO'`. Oracle uses the datafile file path mapping information from the `DbFileNameConvert` property to determine the standby file path of the new standby datafile. If the `StandbyFileManagement` property is set to `'MANUAL'`, you must add a corresponding file to the standby database.

Category	Description
Datatype	String
Valid values	Set the value of this parameter to a list of string pairs: 1. The first string is the substring found in the datafile names on the primary database. 2. The second string is the substring found in the datafile names on the standby database. For example, `('string1', 'string2', 'string3', 'string4',...)` Where: - `string1` is the substring of the primary database filename. - `string2` is the substring of the standby database filename. - `string3` is the substring of the primary database filename. - `string4` is the substring of the standby database filename.
Broker default	`''`
Imported?	Yes, from the `DB_FILE_NAME_CONVERT` initialization parameter
Parameter class	Static
Role	Standby
Standby type	Physical
Corresponds to ...	`DB_FILE_NAME_CONVERT` initialization parameter
Scope	Database
GUI name	`DB File Name Convert`

8.2.10 DelayMins

Specifies the number of minutes log apply services will delay applying the archived redo log files on the standby database.

Category	Description
Datatype	Integer
Valid values	>=0 (minutes)
Broker default	0
Imported?	Yes, from the DELAY_MINS column of the V$ARCHIVE_DEST view of the primary database
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical and logical
Corresponds to ...	■ DELAY attribute for the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database ■ DELAY_MINS column of the V$ARCHIVE_DEST view of the primary database
Scope	Database
GUI name	Apply Delay (mins)

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

8.2.11 Dependency

Specifies the database unique (DB_UNIQUE_NAME) name (can be the primary or a standby database name) on which this database depends for receiving archived redo log files.

Category	Description
Datatype	String
Valid values	Database unique name (DB_UNIQUE_NAME), except for the standby database itself, or you can set this property to an empty string.
Broker default	''
Imported?	No
Parameter class	Dynamic
Role	Standby[1]

Category	Description
Standby type	Physical or logical
Corresponds to ...	DEPENDENCY attribute for the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database
Scope	Database
GUI name	Not applicable

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

See Also: *Oracle Data Guard Concepts and Administration*

8.2.12 HostName

Specifies the name of the host on which the instance is running. The property can only be updated when broker management of the database is disabled. You should only update the value when the host is renamed, in which case you need to disable broker management of the database, update the HostName property to match with the new host name, and then reenable broker management of the database.

Note: If the value of the HostName property does not match the actual name of the host, broker management of the database cannot be enabled.

Category	Description
Datatype	String
Valid values	Name of the host on which the instance is running
Broker default	Not applicable
Imported?	Yes
Parameter class	Not applicable
Role	Primary and standby
Standby type	Physical or logical
Corresponds to ...	HOST_NAME column of the V$INSTANCE view

Category	Description
Scope	Instance
GUI name	Not applicable

8.2.13 InitialConnectIdentifier

Specifies the initial connection identifier the broker uses to make the first connection to a database. If using the CLI, you supply the value when you enter the `CREATE CONFIGURATION` or `ADD DATABASE` command. If using the GUI, the value is supplied by the GUI automatically. You should only update the value of this property when the original value has an error and the broker cannot connect to the database when creating the configuration or adding a database.

Category	Description
Datatype	String
Valid values	A connect identifier that can be used to connect to this database
Broker default	Not applicable
Imported?	No
Parameter class	Not applicable
Role	Primary and standby
Standby type	Physical or logical
Corresponds to ...	Not applicable
Scope	Database
GUI name	Not applicable

8.2.14 LocalListenerAddress

Specifies the listener address at which the instance is registered. The property can only be updated when broker management of the database is disabled. You should only update the value when the `LOCAL_LISTENER` initialization parameter value is being changed, in which case you need to:

1. Disable broker management of the database
2. Update the `LOCAL_LISTENER` initialization parameter value

3. Update the LocalListenerAddress property in listener ADDRESS format to match with the new LOCAL_LISTENER address
4. Reenable broker management of the database

In the event that the LOCAL_LISTENER initialization parameter value of instances belonging to more than one database is being changed, it is recommended to:

1. Disable the configuration
2. Make the LOCAL_LISTENER initialization parameter value changes at all of the instances
3. Make the LocalListenerAddress property changes for all of the affected instances
4. Enable the configuration

> **Note:** If the value of the LocalListenerAddress property does not match the actual address of the listener at which the instance is registered, broker management of the database cannot be enabled.

Category	Description
Datatype	String
Valid values	Listener address, in ADDRESS format
Broker default	(ADDRESS=(PROTOCOL=tcp) (HOST=<host_name>) (PORT=1521))
Imported?	Yes, from the LOCAL_LISTENER initialization parameter and translated into ADDRESS format if the initialization parameter value is a new service name
Parameter class	Dynamic
Role	Primary and standby
Standby type	Physical or logical
Corresponds to ...	LOCAL_LISTENER initialization parameter
Scope	Instance
GUI name	Not applicable

8.2.15 LogArchiveFormat

Specifies the format for filenames of archived redo log files using a database ID (%d), thread (%t), sequence number (%s), and resetlogs ID (%r).

Category	Description
Datatype	String
Valid values	%d_%t_%s_%r
Broker default	Empty string
Imported?	Yes, from the LOG_ARCHIVE_FORMAT initialization parameter on the primary database
Parameter class	Static
Role	Primary and standby
Standby type	Physical and logical
Corresponds to ...	LOG_ARCHIVE_FORMAT initialization parameter
Scope	Instance
GUI name	Not applicable

8.2.16 LogArchiveMaxProcesses

Specifies the initial number of archiver background processes (ARC0 through ARC9) the Oracle database invokes. The actual number of archiver processes in use may increase subsequently based on archive workload.

Category	Description
Datatype	Integer
Valid values	1 to 10
Broker default	2
Imported?	Yes, from the LOG_ARCHIVE_MAX_PROCESSES initialization parameter
Parameter class	Dynamic
Role	Primary and standby
Standby type	Physical and logical
Corresponds to ...	LOG_ARCHIVE_MAX_PROCESSES initialization parameter

Category	Description
Scope	Database
GUI name	`Archiver Processes`

8.2.17 LogArchiveMinSucceedDest

Controls when online redo log files are available for reuse. For the online redo log files to be available for reuse, archiving must succeed to a minimum number of destinations.

Category	Description
Datatype	Integer
Valid values	1 to 10
Broker default	1
Imported?	Yes, from the `LOG_ARCHIVE_MIN_SUCCEED_DEST` initialization parameter
Parameter class	Dynamic
Role	Primary
Standby type	Not applicable
Corresponds to ...	`LOG_ARCHIVE_MIN_SUCCEED_DEST` initialization parameter
Scope	Database
GUI name	Not applicable

8.2.18 LogArchiveTrace

Set this parameter to an integer value to see the progression of the archiving of online redo log files on the primary and the standby databases. The Oracle database writes an audit trail of the archived redo log files received from the primary database into process trace files.

Category	Description
Datatype	Integer

Category	Description
Valid values	A valid value is any combination of any of the following values: 0: Disable archive redo log tracing 1: Track archiving of online redo log file 2: Track archiving status of each archive redo log destination 4: Track archiving operational phase 8: Track ARCHIVELOG destination activity 16: Track detailed ARCHIVELOG destination activity 32: Track ARCHIVELOG destination parameter modifications 64: Track ARCn process state activity 128: Track FAL (fetch archive log) server related activities 256: Supported in a future release 512: Tracks asynchronous LGWR activity 1024: RFS physical client tracking 2048: ARCn/RFS heartbeat tracking
Broker default	255
Imported?	Yes, from the LOG_ARCHIVE_TRACE initialization parameter
Parameter class	Dynamic
Role	Primary and standby
Standby type	Physical and logical
Corresponds to...	LOG_ARCHIVE_TRACE initialization parameter
Scope	Instance
GUI name	Log Archive Trace

8.2.19 LogFileNameConvert

Converts the filename of an online redo log file on the primary database to the filename of a corresponding online redo log file on the standby database.

Category	Description
Datatype	String

Category	Description
Valid values	Set the value of this parameter to a list of an even number of string pairs, separated by commas. 1. The first string is the substring found in the datafile names on the primary database. 2. The second string is the substring found in the datafile names on the standby database. For example, `('string1', 'string2', 'string3', 'string4',...)` Where: - `string1` is the substring of the primary database filename. - `string2` is the substring of the standby database filename. - `string3` is the substring of the primary database filename. - `string4` is the substring of the standby database filename.
Broker default	`''`
Imported?	Yes, from the `LOG_FILE_NAME_CONVERT` initialization parameter
Parameter class	Static
Role	Standby
Standby type	Physical
Corresponds to ...	`LOG_FILE_NAME_CONVERT` initialization parameter
Scope	Database
GUI name	`Log File Name Convert`

8.2.20 LogShipping

Specifies whether or not log transport services can send archived redo log files to the particular standby database. The broker uses the value of the `LogShipping` property only when the primary database is in the `ONLINE` state:

- If the primary database is in the `LOG-TRANSPORT-OFF` state, then log transport services are offline to all standby databases, regardless of whether or not the `LogShipping` property is set to `ON` or `OFF`.

- If the primary database is in the `ONLINE` state and the value of the `LogShipping` property is `ON`, then log transport services are enabled to send archived redo log files to the particular standby database. If the `LogShipping` property is OFF, then log transport services are disabled to send archived redo log files to the particular standby database.

Category	Description
Datatype	String
Valid values	ON or OFF
Broker default	ON
Imported?	No
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical and logical
Corresponds to ...	ENABLE and DEFER values for the LOG_ARCHIVE_DEST_STATE_*n* initialization parameter of the primary database
Scope	Database
GUI name	Log Shipping

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

8.2.21 LogXptMode

Enables you to set the log transport mode. You set the log transport services on each standby database to one of the following modes:

- SYNC

  Configures log transport services for this standby database using the LGWR, SYNC, and AFFIRM attributes of the LOG_ARCHIVE_DEST_*n* initialization parameter. Standby redo log files are required. This mode is required for the maximum protection or maximum availability data protection modes. This log transport mode enables the highest grade of data protection to the primary database, but also incurs the highest performance impact.

- ASYNC

  Configures log transport services for this standby database using the LGWR, ASYNC, and NOAFFIRM attributes of the LOG_ARCHIVE_DEST_*n* initialization parameter. Standby redo log files are required. This mode enables a moderate grade of data protection to the primary database, and incurs a lower performance impact than SYNC.

- ARCH

  Configures log transport services for this standby database using the ARCH attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter. Standby redo log files are not required. This mode enables the lowest grade of data protection to the primary database, and incurs the lowest performance impact. This is the default setting.

Category	**Description**
Datatype	String
Valid values	SYNC or ASYNC or ARCH
Broker default	■ ASYNC for standby databases with standby redo log files ■ ARCH for standby databases without standby redo log files
Imported?	Yes, from the ARCHIVER, TRANSMIT_MODE, and AFFIRM columns of V$ARCHIVE_DEST view of the primary database
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical or logical
Corresponds to ...	■ ARCH, LGWR, SYNC, ASYNC, AFFIRM, and NOAFFIRM attributes for the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database ■ ARCHIVER, TRANSMIT_MODE, and AFFIRM columns of V$ARCHIVE_DEST view of the primary database
Scope	Database
GUI name	Log Transport Mode

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

See Also: Chapter 3 for more information about setting data protection modes for log transport services

8.2.22 LsbyASkipCfgPr

Adds a skip specification to log apply services. It provides a way to control the apply service to skip (ignore) SQL statements that you do not want to apply to the logical standby database. The `SKIP` operation:

- Sets the criteria for identifying the SQL statements that will not be applied to the standby database
- Specifies any additional processing that will be done, if necessary

Specifying a value for this property has no effect and will be ignored if management of the standby database is disabled.

Category	Description
Datatype	String
Valid values	A valid set of arguments to the `DBMS_LOGSTDBY.SKIP` procedure
Broker default	Empty string
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	`DBMS_LOGSTDBY.SKIP` procedure
Scope	Database
GUI name	`Add Skip Table Entries`

See Also: *PL/SQL Packages and Types Reference*

8.2.23 LsbyASkipErrorCfgPr

Adds a skip error specification to log apply services. It provides criteria to determine if an error should cause log apply services to stop. All errors to be skipped are stored in system tables that describe how exceptions should be handled.

Specifying a value for this property has no effect and will be ignored if management of the standby database is disabled.

Category	Description
Datatype	String
Valid values	A valid set of arguments to the DBMS_LOGSTDBY.SKIP_ERROR procedure. The string must contain comma separators between the arguments.
Broker default	Empty string
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	DBMS_LOGSTDBY.SKIP_ERROR procedure
Scope	Database
GUI name	Add Skip Table Entries

See Also: *PL/SQL Packages and Types Reference*

8.2.24 LsbyASkipTxnCfgPr

Skips over a transaction that caused the log apply services to stop applying transactions to the logical standby database. This property enables you to specify the transaction ID (XIDSQN NUMBER) of the problematic transaction that you want log apply services to ignore. Before you restart log apply services, you should issue a SQL transaction that will correctly update the logical standby database in place of the skipped transaction. Applying a compensating transaction will help keep the logical standby database transactionally consistent with the primary database.

Specifying a value for this property has no effect and will be ignored if management of the standby database is disabled.

Category	Description
Datatype	String
Valid values	A valid set of arguments to the DBMS_LOGSTDBY.SKIP_TRANSACTION procedure. Use comma separators between the arguments.
Broker default	Empty string

Category	Description
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	`DBMS_LOGSTDBY.SKIP_TRANSACTION` procedure
Scope	Database
GUI name	`Skip Edit Properties`

Note: Data Guard GUI indirectly supports skipping a transaction using the Skip Edit Properties page.

See Also: *PL/SQL Packages and Types Reference*

8.2.25 LsbyDSkipCfgPr

Deletes an existing skip specification from log apply services. It reverses or removes the actions of the LsbyASkipCfgPr property by finding the record, matching all the parameters, and removing the record from the system table. The match must be exact, and multiple skip actions can be removed only by a matching number of unskip actions. You cannot remove multiple skip actions by using wildcard characters as a value to this property.

Specifying a value for this property has no effect and will be ignored if management of the standby database is disabled.

Category	Description
Datatype	String
Valid Values	A valid set of arguments to the `DBMS_LOGSTDBY.UNSKIP` procedure
Broker Default	Empty string
Imported?	No
Parameter Class	Not applicable
Role	Standby

Category	Description
Standby type	Logical
Corresponds to ...	`DBMS_LOGSTDBY.UNSKIP` procedure
Scope	Database
GUI name	`Remove Skip Table Entries`

See Also: *PL/SQL Packages and Types Reference*

8.2.26 LsbyDSkipErrorCfgPr

Deletes an existing skip error specification from log apply services. It reverses or removes the actions of the LsbyASkipErrorCfgPr property by finding the record, matching all of the parameters and removing the record from the system table. The match must be exact, and multiple skip actions can be removed only by a matching number of unskip actions. You cannot remove multiple skip actions by using wildcard characters as a value to this property.

Specifying a value for this property has no effect and will be ignored if management of the standby database is disabled.

Category	Description
Datatype	String
Valid values	A valid set of arguments to the `DBMS_LOGSTDBY.UNSKIP_ERROR` procedure. The string must contain comma separators between the arguments.
Broker default	Empty string
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	`DBMS_LOGSTDBY.UNSKIP_ERROR` procedure
Scope	Database
GUI name	`Remove Skip Table Entries`

See Also: *PL/SQL Packages and Types Reference*

8.2.27 LsbyDSkipTxnCfgPr

Reverses or removes the actions of the LsbyASkipTxnCfgPr property. The transaction IDs must match exactly, and multiple skip transaction actions can be removed only by a matching number of unskip transaction actions. You cannot remove multiple skip transaction actions by using wildcard characters as a value to this property.

Specifying a value for this property has no effect and will be ignored if management of the standby database is disabled.

Category	Description
Datatype	String
Valid values	A valid set of arguments to the `DBMS_LOGSTDBY.UNSKIP_TRANSACTION` procedure
Broker default	Empty string
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	`DBMS_LOGSTDBY.UNSKIP_TRANSACTION` procedure
Scope	Database
GUI name	Not applicable

See Also: *PL/SQL Packages and Types Reference*

8.2.28 LsbyMaxEventsRecorded

Specifies the number of events that will be stored in the `DBA_LOGSTDBY_EVENTS` table, which stores logical standby event information.

Category	Description
Datatype	Integer
Valid values	>=0

Category	Description
Broker default	0
Imported?	Yes, from the MAX_EVENTS_RECORDED row of SYSTEM.LOGSTDBY$PARAMETERS
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	DBMS_LOGSTDBY.APPLY_SET('MAX_EVENTS_RECORDED') and the DBMS_LOGSTDBY.APPLY_UNSET('MAX_EVENTS_RECORDED') procedures
Scope	Database
GUI name	Max Events Recorded

8.2.29 LsbyMaxSga

Specifies the number of megabytes for the allocation of log apply services cache in the system global area (SGA). If the value is 0, log apply services use one quarter of the value set for the SHARED_POOL_SIZE initialization parameter.

Category	Description
Datatype	Integer
Valid values	>=0
Broker default	0
Imported?	Yes, from the MAX_SGA row of SYSTEM.LOGSTDBY$PARAMETERS
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	DBMS_LOGSTDBY.APPLY_SET('MAX_SGA') and the DBMS_LOGSTDBY.APPLY_UNSET('MAX_SGA') procedures
Scope	Instance
GUI name	Max SGA (MB)

8.2.30 LsbyMaxServers

Specifies the number of parallel query servers specifically reserved for log apply services. If the value is 0, log apply services use all available parallel query servers to read the log files and apply changes.

Category	Description
Datatype	Integer
Valid values	>=0
Broker default	0
Imported?	Yes, from the `MAX_SERVERS` row of `SYSTEM.LOGSTDBY$PARAMETERS`
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	`DBMS_LOGSTDBY.APPLY_SET('MAX_SERVERS')` and the `DBMS_LOGSTDBY.APPLY_UNSET('MAX_SERVERS')` procedures
Scope	Instance
GUI name	`Max Servers`

8.2.31 LsbyRecordAppliedDdl

Controls whether or not DDL statements that were applied to the logical standby database are recorded in the `DBA_LOGSTDBY_EVENTS` table. Specify one of the following values:

- `TRUE`: DDL statements applied to the logical standby database are recorded in the `DBA_LOGSTDBY_EVENTS` table. This is the default parameter setting.
- `FALSE`: Applied DDL statements are not recorded.

Category	Description
Datatype	String
Valid values	`TRUE` or `FALSE`
Broker default	`TRUE`

Category	Description
Imported?	Yes, from the RECORD_APPLIED_DDL row of SYSTEM.LOGSTDBY$PARAMETERS
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	DBMS_LOGSTDBY.APPLY_SET('RECORD_APPLIED_DDL') and the DBMS_LOGSTDBY.APPLY_UNSET('RECORD_APPLIED_DDL') procedures
Scope	Database
GUI name	Record Applied DDL

8.2.32 LsbyRecordSkipDdl

Controls whether or not skipped DDL statements are recorded in the DBA_LOGSTDBY_EVENTS table. Specify one of the following values:

- TRUE: Skipped DDL statements are recorded in the DBA_LOGSTDBY_EVENTS table. This is the default parameter setting.
- FALSE: Skipped DDL statements are not recorded in the DBA_LOGSTDBY_EVENTS table.

Category	Description
Datatype	String
Valid values	TRUE or FALSE
Broker default	TRUE
Imported?	Yes, from the RECORD_SKIP_DDL row of SYSTEM.LOGSTDBY$PARAMETERS
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	DBMS_LOGSTDBY.APPLY_SET('RECORD_SKIP_DDL') and the DBMS_LOGSTDBY.APPLY_UNSET('RECORD_SKIP_DDL') procedures

Category	Description
Scope	Database
GUI name	`Record Skip DDL`

8.2.33 LsbyRecordSkipErrors

Controls whether or not skipped errors (as described by the `DBMS_LOGSTDBY.SKIP_ERROR` procedure) are recorded in the `DBA_LOGSTDBY_EVENTS` table. Specify one of the following values:

- `TRUE`: Skipped errors are recorded in the `DBA_LOGSTDBY_EVENTS` table.
- `FALSE`: Skipped errors are not recorded in the `DBA_LOGSTDBY_EVENTS` table.

Category	Description
Datatype	String
Valid values	`TRUE` or `FALSE`
Broker default	`TRUE`
Imported?	Yes, from the `RECORD_SKIP_ERRORS` row of `SYSTEM.LOGSTDBY$PARAMETERS`
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to	`DBMS_LOGSTDBY.APPLY_SET('RECORD_SKIP_ERRORS')` and the `DBMS_LOGSTDBY.APPLY_UNSET('RECORD_SKIP_ERRORS')` procedures
Scope	Database
GUI name	`Record Skip Errors`

8.2.34 LsbyTxnConsistency

Controls the level of transaction consistency maintained between the primary and standby databases. Specify one of the following values:

- `FULL`: Transactions are applied to the logical standby database in the exact order in which they were committed on the primary database. This option may affect performance.

- READ_ONLY: Transactions are committed out of order (which provides better performance). SQL SELECT statements return read-consistent results. This is particularly beneficial when the logical standby database is being used to generate reports.

> **Note:** DML statements involving standby tables are not allowed in this mode.

- NONE: Transactions are committed out of order and no attempt is made to provide read-consistent results. This results in the best performance of the three modes. If applications reading the logical standby database make no assumptions about transaction order, this option works well.

Category	Description
Datatype	String
Valid values	FULL or READ_ONLY or NONE
Broker default	FULL
Imported?	Yes, from the TRANSACTION_CONSISTENCY row of SYSTEM.LOGSTDBY$PARAMETERS
Parameter class	Not applicable
Role	Standby
Standby type	Logical
Corresponds to ...	DBMS_LOGSTDBY.APPLY_SET('TRANSACTION_CONSISTENCY') and the DBMS_LOGSTDBY.APPLY_UNSET('TRANSACTION_CONSISTENCY') procedures
Scope	Database
GUI name	Transaction Consistency Level

8.2.35 MaxFailure

Specifies the maximum number of contiguous archiving failures before the log transport services stop trying to transport archived redo log files to the standby database. A value of zero indicates that an unlimited number of failures are allowed.

Category	Description
Datatype	Integer
Valid values	>=0
Broker default	0
Imported?	Yes, from the MAX_FAILURE column of V$ARCHIVE_DEST view of the primary database
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical and logical
Corresponds to ...	▪ MAX_FAILURE attribute for the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database ▪ MAX_FAILURE column of the V$ARCHIVE_DEST view of the primary database
Scope	Database
GUI name	Not applicable

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

8.2.36 NetTimeout

Specifies the number of seconds the LGWR waits for Oracle Net Services to respond to a LGWR request. It is used to bypass the long connection timeout in TCP. This property is only used when the LogXptMode property of the same database is set to SYNC or ASYNC.

Category	Description
Datatype	Integer
Valid values	0, 15 to 1200
Broker default	30
Imported?	Yes, from the NET_TIMEOUT column of V$ARCHIVE_DEST view of the primary database
Parameter class	Dynamic
Role	Standby[1]

Category	Description
Standby type	Physical and logical
Corresponds to ...	■ NET_TIMEOUT attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database ■ NET_TIMEOUT column of V$ARCHIVE_DEST view of the primary database
Scope	Database
GUI name	Not applicable

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

8.2.37 PreferredApplyInstance

Indicates that a particular instance is the preferred choice for serving log apply services. It is only used when the database is a standby RAC database. The value could be an empty string (default) which means the broker chooses the apply instance.

Category	Description
Datatype	String
Valid Values	The instance name (SID) or empty string
Broker Default	Empty string
Imported?	No
Parameter Class	Not applicable
Role	Standby
Standby Type	Physical and logical
Corresponds to	Not applicable
Scope	Database
GUI Name	Apply Instance

See Also: Section 3.5.8 for more information

8.2.38 RealTimeApply

Turns on and off the real-time apply feature of the physical or logical standby database. You need to set up standby redo log files to use the real-time apply feature. Once real-time apply is turned on, the apply delay feature is automatically turned off.

Category	Description
Datatype	String
Valid values	`ON` or `OFF`
Broker default	`OFF`
Imported?	No
Parameter class	Not applicable
Role	Standby
Standby type	Physical and logical
Corresponds to ...	▪ `USING CURRENT LOGFILE` clause of the `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE` command for a physical standby database ▪ `IMMEDIATE` clause of the `ALTER DATABASE START LOGICAL STANDBY APPLY` command for a logical standby database
Scope	Database
GUI name	`Real Time Apply`

See Also: Section 3.5.1 and *Oracle Data Guard Concepts and Administration* for information about managing real-time apply

8.2.39 ReopenSecs

Specifies the minimum number of seconds before the archiver process (ARC*n*, foreground, or log writer process) should try again to access a previously failed destination.

Category	Description
Datatype	Integer
Valid values	>=0 seconds

Category	Description
Broker default	0
Imported?	Yes, from the REOPEN_SECS column of V$ARCHIVE_DEST view of the primary database
Parameter class	Dynamic
Role	Standby[1]
Standby type	Physical and logical
Corresponds to ...	■ REOPEN attribute for the LOG_ARCHIVE_DEST_*n* initialization parameter of the primary database ■ REOPEN_SECS column of the V$ARCHIVE_DEST view of the primary database
Scope	Database
GUI name	Not applicable

[1] Although this property is set for the standby database, it is indirectly related to the log transport services for the primary database. The broker propagates the setting you specify on the standby database to the corresponding attributes of the LOG_ARCHIVE_DEST_*n* value of the primary database.

8.2.40 SidName

Specifies the SID of the instance. The property can only be updated when broker management of the database is disabled. You should only update the value when the SID is changed, in which case you need to disable broker management of the database, update the SidName property to match with the new SID, and reenable broker management of the database.

Note: If the value of the SidName property does not match the actual value of the SID, broker management of the database cannot be enabled.

Category	Description
Datatype	String
Valid values	SID of the instance
Broker default	Not applicable
Imported?	Yes

Category	Description
Parameter class	Not applicable
Role	Primary and standby
Standby type	Physical or logical
Corresponds to ...	INSTANCE_NAME column of the V$INSTANCE view
Scope	Instance
GUI name	Not applicable

8.2.41 StandbyArchiveLocation

Specifies the location of archived redo log files arriving from a primary database. Oracle recommends that you always explicitly set the value (if flash recovery area is not in use).

Category	Description
Datatype	String
Valid values	Nonempty file specification of the location of archived redo log files on the standby database
Broker default	dgsby_<db_unique_name>
Imported?	Yes, from the DESTINATION column of the V$ARCHIVE_DEST fixed view of the standby database where the destination is a local destination and where the VALID_FOR attribute is compatible with the string (STANDBY_ROLE, STANDBY_LOGFILE); if no such destination exists, import is from the STANDBY_ARCHIVE_DEST initialization parameter
Parameter class	Dynamic
Role	Standby
Standby type	Physical or logical
Corresponds to ...	■ LOCATION attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter of the standby database with VALID_FOR compatible with (STANDBY_ROLE, STANDBY_LOGFILE) ■ DESTINATION column of the V$ARCHIVE_DEST view of the standby database
Scope	Instance
GUI name	Standby Archive Location

Note: On a logical standby database, Oracle recommends the LOCATION attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter for the local destination be different from the value of StandbyArchiveLocation property.

8.2.42 StandbyFileManagement

Affects how the add datafile operation on the primary database is applied on the standby database. If this property is set to AUTO, in conjunction with valid settings in the DbFileNameConvert property, a corresponding new datafile is automatically created on the standby database. The location of this new standby datafile is determined by the value of the DbFileNameConvert property.

If this property is set to MANUAL, you have to create the correct new datafile on the standby database manually.

Category	Description
Datatype	String
Valid values	AUTO or MANUAL
Broker default	AUTO
Imported?	Yes, from the STANDBY_FILE_MANAGEMENT initialization parameter
Parameter class	Dynamic
Role	Standby
Standby type	Physical or logical
Corresponds to ...	STANDBY_FILE_MANAGEMENT initialization parameter
Scope	Database
GUI name	Not applicable

9

Troubleshooting Data Guard

This chapter describes various errors and how to solve them. This chapter contains the following topics:

- Section 9.1, "Sources of Diagnostic Information"
- Section 9.2, "General Problems and Solutions"
- Section 9.3, "Troubleshooting Problems During a Failover Operation"
- Section 9.4, "Troubleshooting Problems During a Switchover Operation"

9.1 Sources of Diagnostic Information

The Data Guard broker provides information about its activities in several forms:

- Database status information (see Section 3.7)
- Oracle alert log files

  The broker records key information in the alert log file for each database. You can check the alert log files for such information when troubleshooting Data Guard.
- Data Guard "broker log" files

  For each database in a broker configuration, the broker monitor processes record important behavior and status information in a "broker log" file. The broker log file is maintained as follows for a UNIX environment:

  ```
  $ORACLE_HOME/rdbms/log/drc*.log
  ```

  > **See Also:** Section 8.1.3 for additional information about the LatestLog property

9.2 General Problems and Solutions

This section describes general problems and solutions. This section contains the following topics:

- ORA-16596: Database is Not a Member of the Data Guard Configuration
- Log Files Are Being Accumulated on the Primary and Not Archived to Some Standby Databases
- Many Log Files Are Received on a Standby Database But Not Applied
- The Primary Database is Flashed Back

9.2.1 ORA-16596: Database is Not a Member of the Data Guard Configuration

A request was issued to the broker, but the database instance through which you have connected is no longer a part of the broker configuration. You may see this error when the broker monitor (DMON) fails to locate a broker configuration profile for the database upon which it is running.

Solution Reconnect to the configuration through another database instance. Confirm that the value of the `DB_UNIQUE_NAME` initialization parameter for the database and the database object name in the broker configuration are identical.

This problem can also occur if you attempt to enable a configuration, but the broker configuration file for one of its databases was accidentally removed or is outdated. In this case, remove the database from the broker configuration, manually delete the configuration file for that standby database (not for the primary database), and try again to enable the configuration. After the configuration is enabled, run the Add Standby Database wizard and choose the **Add existing standby database** option to create a new database profile for the previously deleted standby database.

9.2.2 Log Files Are Being Accumulated on the Primary and Not Archived to Some Standby Databases

By viewing the Log File Details page for the Data Guard GUI, you have determined that log files are accumulating on the primary database and are not being archived to some of the standby databases in the broker configuration.

Solution To narrow down the problem, do the following:

- Verify that the state of the primary database is `ONLINE` (not `LOG-TRANSPORT-OFF`).

- Verify that the value of the LogShipping property of the standby database in question is ON.
- Check the status of the primary database log transport services. If log transport services have an error, use the error message to determine further checking and resolution action. For example:
  - If the error indicates the standby database is not available, you need to restart the standby database.
  - If the error indicates no listener, you need to restart the listener.
  - If the error indicates the standby database has no local destination, you need to set up a standby location to store archived redo log files from the primary database.

9.2.3 Many Log Files Are Received on a Standby Database But Not Applied

By viewing the Performance page or Log File Details page for the Data Guard GUI, you have determined that the standby database accumulates too many log files without applying them.

Solution There are many possible reasons why archived redo log files might not be applied to the standby database. Investigate why the log files are building up and rule out the valid reasons.

If the current status of the standby database is not normal:

- Determine whether or not the log apply services might be unexpectedly offline. See the ORA-16766 (for physical standby databases) or ORA-16768 (for logical standby databases) problem and solution statement for more help.
- If this is a logical standby database, check to see if a failed transaction has occurred.
- If you want to suppress the error while you investigate the problem, you can temporarily disable broker management of the database.

  See Also: Chapter 7 for additional information about disabling the database using the CLI

If the current status of the standby database is normal:

- Verify the state of the standby database is online (not in the LOG-APPLY-OFF, OFFLINE, or READ-ONLY states).

- Verify the state of the primary database is online (not in the LOG-TRANSPORT-OFF state).

  See Also: Chapter 8 for additional information about the LogShipping property

- Check to see if log files are building up because the value of the DelayMins property is set too large. (Log apply services will delay applying the archived redo log files on the standby database for the number of minutes specified.)

  See Also: Chapter 8 for additional information about the DelayMins property

- If you cannot see any errors, compare the archive rate to the apply rate on the Performance page in the Data Guard GUI to see if the apply rate is lower than the archive rate.

9.2.4 The Primary Database is Flashed Back

If the primary database is flashed back, the standby databases in the configuration are disabled by the broker because they are no longer viable as standbys in their current state. If the standby database has applied logs beyond the point to which the primary was flashed back, and the flashback feature is enabled on this standby database, you can restore their viability as standby databases by flashing them back to the same point or to an earlier point. Then, reenable them within the broker configuration as follows:

```
DGMGRL> ENABLE DATABASE <db_unique_name>;
```

For more information about restoring the viability of a standby database that was disabled by the broker, see Section 4.2.5.

9.3 Troubleshooting Problems During a Failover Operation

Although it is possible for a failover to stop, it is unlikely. If an error occurs, it is likely to happen when the standby database is transitioning to the primary role. If the error status indicates that this problem occurred, use these general guidelines to fix the problem:

1. Investigate the error information provided by the broker to find the source of the problem and correct it.
2. Perform the failover again.

The following problems may occur during a failover operation:

Problem: The primary database is still running. The Data Guard GUI tries to detect if the primary database is still running. However, if the primary database is still running and the Data Guard GUI cannot detect that it is running, the failover will not be successful.

Solution: If the primary database is still running but can no longer function as a primary database, shut it down and retry the failover.

9.4 Troubleshooting Problems During a Switchover Operation

If the switchover fails due to problems with the configuration, the broker reports any problems it encounters in the alert log files or in the broker log files (see Section 9.1). In general, you can choose another database for the switchover or restore the configuration to its pre-switchover state and then retry the switchover. The following subsections describe how to recover from the most common problems.

Problems Transitioning the Primary Database to the Standby Role

If the error status indicates a problem when transitioning the original primary database to the standby role (including stopping log transport services and starting log apply services), use these general guidelines to recover:

1. Disable the configuration using the CLI.

> **Note:** You can enable or disable the configuration using the CLI. You cannot disable the configuration using the Data Guard GUI. You can enable the configuration using the Data Guard GUI in the event that it was previously disabled using the CLI.

2. Investigate the error message returned by the broker to find the source of the problem on the primary database and correct it. Oracle Enterprise Manager provides an Alert Log Content link for viewing alert log file information. You may also examine the contents of the broker log file for the primary database.
3. Reenable the configuration to refresh and restore the databases to their original roles and states.
4. Perform the switchover again.

Problems Transitioning the Standby Database to the Primary Role

If the error status indicates that a problem occurred when transitioning the target standby database to the primary role (including stopping log apply services and starting log transport services), use these general guidelines to restore to the pre-switchover state.

Step 1 Analyze and correct the detected failure.

1. Disable the configuration using the CLI.

> **Note:** You can enable or disable the configuration using the CLI. You cannot disable the configuration using the Data Guard GUI. You can enable the configuration using the Data Guard GUI in the event that it was previously disabled using the CLI.

2. Investigate the error messages returned by the broker to find the source of the problem on the standby database and correct it. Examine the alert log file information and the contents of the broker log file for the target standby database.

Step 2 Convert back to a primary database.

The original primary database was already converted to a standby database by the time this stage of the switchover is reached. The database must be converted back to being a primary database as follows:

1. Ensure the original primary database is at least mounted. A restart may be necessary:

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT
```

2. Execute the following SQL*Plus command to convert the database back to running in the primary database role:

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
```

3. Restart the database if necessary. The database is now in its former role as a primary database.

Additional steps may be needed to allow any logical standby databases in the configuration to accept and apply redo logs from this primary database. See *Oracle Data Guard Concepts and Administration* for these steps.

Step 3 Finish the recovery.

1. Reenable the configuration.
2. Perform the switchover operation again.

Additional Problems that May Occur During a Switchover Operation

Problem: The switchover fails due to problems with log transport services.

Solution:

1. Verify the state and status of the standby database by viewing its information on the Data Guard Overview page.

   See Also: Chapter 5 for additional information about the Data Guard Overview page

2. Run the Verify operation after the switchover completes to examine the alert log file of the new primary database and to verify the status of the configuration.

   See Also: Section 5.7.1 for additional information about verifying the configuration

Problem: The switchover may fail during verification checks done by Data Guard broker (for example, log transport services might return errors on a database that is involved in the switchover).

Solution: Choose another database for the switchover or fix the problem by transporting the archived redo log files.

A

Data Guard Broker Changed and Deprecated Features

This appendix details the Data Guard broker features that were deprecated or changed. This appendix provides the following sections:

- Section A.1, "Data Guard Broker Changed Features"
- Section A.2, "Data Guard Broker Deprecated Features"

A.1 Data Guard Broker Changed Features

This section contains information about Data Guard broker features that were changed.

A.1.1 General Features That Changed

- The concept of a *graceful* failover was renamed to *complete* failover while *forced* failover was renamed to *immediate* failover.

  > **See Also:** Chapter 7 for information about the Oracle Database 10*g* Data Guard broker `FAILOVER` command

- The concept of a primary site was changed to a primary database and standby site was changed to standby database. The former *site* object of a broker configuration was eliminated for Oracle Data Guard.
- The GUI's default behavior has been reversed when removing parts of a broker configuration. The GUI now clears the archive destination parameter upon removal of a database from the broker configuration, or clears all of the

parameters upon removal of the entire configuration. The GUI still provides the option of preserving these values as was done by default in previous releases.

A.1.2 Changed Properties

Table A–1 shows properties that changed:

Table A–1 Changed Properties

Property Name	Description of Change
InconsistentLogXptProps (monitorable property)	■ STANDBY_NAME column replaces STANDBY_SITE_NAME. ■ INSTANCE_NAME is a new column. ■ MEMORY_VALUE column replaces DATABASE_VALUE. ■ BROKER_VALUE column replaces METADATA_VALUE.
InconsistentProperties (monitorable property)	■ INSTANCE_NAME is a new column. ■ MEMORY_VALUE column replaces DATABASE_VALUE. ■ BROKER_VALUE column replaces METADATA_VALUE.
LogXptStatus (monitorable property)	■ Output was changed to a table format.
RecvQEntries (monitorable property)	■ Replaces SbyLogQueue property. ■ THREAD is a new column. ■ FIRST_CHANGE# is a new column. ■ NEXT_CHANGE# is a new column. ■ SIZE is a new column.
SendQEntries (monitorable property)	■ THREAD is a new column. ■ STANDBY_NAME column replaces SITE_NAME. ■ FIRST_CHANGE# is a new column. ■ NEXT_CHANGE# is a new column. ■ SIZE is a new column.
ApplyParallel (configurable property)	■ New default value is AUTO
AsyncBlocks (configurable property)	■ New default value is 61,440

Table A–1 (Cont.) Changed Properties

Property Name	Description of Change
LogXptMode (configurable property)	New default values are: ▪ ASYNC for standby databases with standby redo log files ▪ ARCH for standby databases without standby redo log files
ParallelApply (configurable property)	▪ Allows input values AUTO and NO.

See Also: Chapter 8, "Database Properties"

A.1.3 Changed State Names

Table A–2 shows the state names that changed:

Table A–2 State Name Changes

Database Type	State Name Prior to Oracle Database 10*g*	New State Name as of Oracle Database 10*g*
Primary	READ-WRITE	LOG-TRANSPORT-OFF
Primary	READ-WRITE-XPTON	ONLINE
Physical standby	PHYSICAL-APPLY-READY	LOG-APPLY-OFF
Physical standby	PHYSICAL-APPLY-ON	ONLINE
Logical standby	LOGICAL-APPLY-READY	LOG-APPLY-OFF
Logical standby	LOGICAL-APPLY-ON	ONLINE

See Also: Section 3.2, "Database States"

A.1.4 Changed CLI Features

When removing a standby database from the broker configuration, the CLI provides you with a choice about removing the standby destination from the primary database so that log files are no longer transported to that standby database. The default for this choice in Oracle Database 10*g* is to remove that destination. This is a reversal of the default (and only) behavior as it was implemented in the Oracle9*i* DGMGRL CLI.

A.1.5 Changed Data Guard GUI Features

This section contains information about GUI features that changed.

- You cannot disable broker management of the broker configuration from within the Data Guard GUI as had been allowed in previous releases. You can only do this from the command-line interface.
- In this release of Oracle Database 10*g*, the Test Application feature will not be available. This feature will return in a future release of Data Guard broker.
- View Log does not view alert or broker log files.

  See Also: Section 8.1.3, "LatestLog"

- When removing a standby database from the broker configuration, the Data Guard GUI presents you with a choice about removing the standby destination from the primary database so that log files are no longer transported to that standby database. The default for this choice in Oracle Database 10*g* is to remove that destination. This is a reversal of the default as it was implemented in Oracle9*i* Data Guard Manager.

A.2 Data Guard Broker Deprecated Features

This section contains information about Data Guard broker features that were deprecated or are obsolete.

- The concept of a *site object* is obsolete.
- The concept of a *database resource object* is obsolete.

Both of these objects were replaced by *database objects*.

A.2.1 Deprecated Properties

Table A–3 shows properties that were deprecated and if there is a replacement property.

Table A–3 Deprecated Properties

Deprecated Property	Replacement Property (if any)
`Alternate`	No replacement
`StandbyArchiveDest`	`StandbyArchiveLocation`
`SbyLogQueue`	`RecvQEntries`

See Also: Chapter 8, "Database Properties" for information about the properties that replaced deprecated properties.

A.2.2 Deprecated CLI Commands and Keywords

Table A–4 shows CLI commands or keywords that were deprecated and if there is a replacement command or keyword.

Note: Existing Oracle9*i* command-line scripts are supported in Oracle Database 10*g* for non-RAC databases. Oracle recommends upgrading your scripts to replace deprecated commands and keywords with their Oracle Database 10*g* replacements.

Table A–4 Deprecated Commands or Keywords

Deprecated Command or Keyword	Replacement Command or Keyword
`ALTER CONFIGURATION (protection mode)`	`EDIT CONFIGURATION (protection mode)`
`ALTER CONFIGURATION (state)`	No replacement
`ALTER RESOURCE (property)`	`EDIT DATABASE (property)`
`ALTER RESOURCE (state)`	`EDIT DATABASE (state)`
`ALTER SITE (AUTO PFILE)`	`EDIT INSTANCE (AUTO PFILE)`
`ALTER SITE (state)`	No replacement

`CREATE CONFIGURATION`
`PRIMARY SITE IS site-name`
`RESOURCE IS resource-name`
`HOSTNAME IS host-name`
`INSTANCE NAME IS instance-name`
`SERVICE NAME IS net-service-name`
`SITE IS MAINTAINED AS standby-type`
	`CREATE CONFIGURATION`
	`PRIMARY DATABASE IS database-name`
	`CONNECT IDENTIFIER IS connect-identifier`

`CREATE SITE`	No replacement
`DISABLE RESOURCE`	`DISABLE DATABASE`
`DISABLE SITE`	No replacement
`ENABLE RESOURCE`	`ENABLE DATABASE`

Table A–4 (Cont.) Deprecated Commands or Keywords

Deprecated Command or Keyword	Replacement Command or Keyword
`ENABLE SITE`	No replacement

`FAILOVER TO site-name`
`GRACEFUL`
`FORCED`
	`FAILOVER TO database-name`
	`IMMEDIATE`

`REMOVE SITE`	`REMOVE DATABASE`
`SHOW CONFIGURATION VERBOSE property-name`	`SHOW CONFIGURATION`
`SHOW DEPENDENCY TREE`	No replacement
`SHOW LOG`	No replacement
`SHOW RESOURCE`	`SHOW DATABASE`
`SHOW SITE`	No replacement
`SWITCHOVER TO site-name`	`SWITCHOVER TO database-name`

See Also: Chapter 7, "Data Guard Command-Line Interface Reference" for information about the commands that replaced deprecated commands.

A.2.3 Data Guard GUI Features That Are Deprecated

This section contains information about GUI features that are deprecated.

- The Add Site wizard is obsolete and has been replaced by the Add Standby Database wizard. Use the Add Standby Database wizard to create a broker configuration and to create new or add existing standby databases to the broker configuration.

Glossary

apply instance

In an Oracle Real Application Clusters (RAC) databases environment, the apply instance is the one instance applying archived redo log files to either a physical standby database or logical standby database.

broker

A distributed management framework that automates and simplifies most of the complex operations required to create, control, and monitor a Data Guard configuration.

broker configuration

A logical grouping of the primary and standby databases (including log transport services and log apply services) of a Data Guard configuration.

See also Data Guard configuration.

configuration object

A named collection of database objects. It is an abstraction of an actual Data Guard configuration.

database object

A named object that corresponds to a primary or standby database in a Data Guard configuration. The broker uses this object to manage and control the state of a single database and to associate properties with the database.

Data Guard configuration

A distributed computing system that prevents or minimizes losses due to unplanned events (for example, human errors, environmental disasters, or data corruption) as well as to planned downtime (such as for routine maintenance tasks).

See also broker configuration.

Data Guard environment

The physical configuration of the primary and standby databases. The environment depends on many factors, including the:

- Number of standby databases associated with a primary database
- Number of host systems used by the databases
- Directory structures of the machines used by the databases
- Network configuration
- Log transport services
- Log apply services

The Data Guard environment can be managed manually by a DBA, automatically using the Data Guard GUI or the Data Guard CLI, or a combination of all of these.

default state

The initial runtime state in which the database object will run when you enable broker management of the configuration. The actual default state can vary depending on the role (primary or standby) in which the database is running.

See also intended state.

failover

Failover changes a standby database into the role of a primary database.

instance

A named object; a database object is a collection of one or more named instances. The broker uses this object to manage and control the state of the database with which the instance is associated, and to associate instance-specific properties with this instance of the database.

intended state

The runtime state of a database object while it is enabled for management by the broker.

See also default state.

logical standby database

A logical standby database takes standard Oracle archived redo log files, transforms them back into SQL transactions, and then applies them to an open standby database. Although changes can be applied concurrently with end-user access, the tables being maintained through regenerated SQL transactions allow read-only access to users of the logical standby database. Because the database is open to allow application of reconstructed SQL transactions, it is physically different from the primary database. The database tables can have different indexes and physical characteristics from their primary database peers, but the tables must maintain logical consistency from an application access perspective, to fulfill their role as a standby data source.

physical standby database

A physical standby database is physically identical to the primary database. While the primary database is open and active, a physical standby database is either performing recovery (by applying online redo log files), or open for reporting access. A physical standby database can be queried (read-only mode) when not performing recovery while the primary database continues to archive redo logs.

primary database

A production database from which one or more standby databases is created and maintained. Every standby database is associated with one and only one primary database. A single primary database can, however, support multiple standby databases. The Data Guard broker monitor (DMON) maintains the master copy of the binary configuration file with the primary database, ensuring that each standby database's copy of the file is kept up to date as changes are made.

The broker refers to this database using the value in the `DB_UNIQUE_NAME` initialization parameter which is defined to be globally unique.

profiles

The description of a database object including its current state (on or off), properties, and current status (for example, its health). This description is maintained persistently by the broker in its binary configuration file.

read-only mode

A physical standby database mode that is initiated using the following SQL statement:

```
ALTER DATABASE OPEN READ ONLY;
```

The read-only mode enables you to query a physical standby database, but does not allow you to make changes to it. While in this mode, online redo log files are archived but are not applied until the physical standby database reenters Redo Apply mode.

standby database

A copy of a primary database created using a backup of your primary database. Standby databases are kept synchronized with the primary database by applying archived redo log files over time from the primary database to each standby database. The standby database can take over processing from the primary database, providing nearly continuous database availability. A standby database has its own server parameter file, control file, and datafiles. It also has a copy of the broker's configuration file, kept up to date at the direction of the DMON process running in the primary database instance.

The broker refers to a standby database by its globally unique name that is stored in its `DB_UNIQUE_NAME` initialization parameter.

See also logical standby database and physical standby database.

Index