

Oracle® Database

Global Data Services Concepts and Administration Guide

12c Release 1 (12.1)

E22100-11

June 2015

Oracle Database Global Data Services Concepts and Administration Guide, 12c Release 1 (12.1)

E22100-11

Copyright © 2013, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Virginia Beecher

Contributing Authors: Janet Stern, Richard Strohm

Contributors: Bob McQuirk, Cris Pedregal-Martin, Jean Zeng, Leonid Novak, Mark Dilman, Nourdine Benadjaoud, Shahab Hamid, Srinagesh Battula, Steve Ball, Wei Hu

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii
Changes in This Release for Oracle Database Global Data Services Concepts and Administration Guide	ix
Changes in Oracle Database 12c Release 1 (12.1.0.2)	ix
1 Introduction to Global Data Services	
Introduction to Global Data Services	1-1
Global Data Services Architecture	1-3
Global Data Services Pool	1-3
Global Data Services Region	1-4
Global Service Manager	1-4
Global Data Services Catalog	1-5
Oracle Notification Service Servers	1-5
Overview of Global Services	1-5
Global Service Attributes	1-6
Global Services in an Oracle RAC Database	1-7
Global Services in an Oracle Data Guard Broker Configuration	1-7
Database Placement of a Global Service	1-8
Replication Lag and Global Services	1-9
Global Services and Distributed Transaction Processing	1-10
Global Services in Multitenant Architecture	1-11
Global Connection Load Balancing	1-11
Client-Side Load Balancing	1-11
Server-Side Load Balancing	1-12
Region Affinity for Global Services	1-12
Any-Region Affinity	1-12
Affinity to a Local Region	1-12
Affinity to a Local Region with Interregion Failover	1-13
Global Run-Time Connection Load Balancing	1-13
Affinity	1-14

Disk I/O and CPU Thresholds	1-14
Global Services Failover	1-14
Global Service Manager Process Suite Architecture.....	1-15
Global Data Services Use Cases	1-17
Load Balancing for Replicated Databases	1-17
Service Failover for Replicated Databases.....	1-18
Region Affinity in Oracle GoldenGate Multi-Master	1-19
Load Balancing in Oracle GoldenGate Multi-Master	1-20
Balancing Oracle Active Data Guard and Oracle GoldenGate Reader Farms.....	1-21

2 Configuring the Global Data Services Framework

Planning an Installation.....	2-1
Installing a Global Service Manager.....	2-2
Upgrading Global Data Services.....	2-3
Creating the Global Data Services Catalog.....	2-5
Adding a Global Service Manager to the Global Data Services Catalog.....	2-6
Connecting to the Global Data Services Catalog.....	2-7
Adding a Global Data Services Pool.....	2-8
Adding a Global Data Services Region	2-9
Adding a Database to a Global Data Services Pool.....	2-9
Valid Node Checking for Registration.....	2-10
Adding a Service to a Global Data Services Pool	2-10
Starting a Global Service	2-11
Database Client Configuration.....	2-11
Exporting the GDS Catalog Data for Logical Backups	2-13
Restoring Logical Backup of the GDS Catalog into the Same Catalog Database	2-13
Restoring Logical Backup of the GDS Catalog into a new Catalog Database.....	2-14
Changing the GSMCATUSER Password	2-14

3 Administering Global Data Services Configurations

Overview of Global Data Services Administration	3-1
Managing Database Pools	3-3
Adding Oracle Data Guard Broker Managed Databases to a Database Pool	3-3
Managing Global Services	3-3
Creating a Global Service.....	3-4
Starting a Global Service	3-4
Stopping a Global Service	3-5
Disabling a Global Service	3-6
Enabling a Global Service	3-6
Modifying Global Service Attributes	3-6
Deleting a Global Service	3-7

4 Troubleshooting

Troubleshooting Oracle Error Codes.....	4-1
ORA-01045: user GSMADMIN_INTERNAL lacks CREATE SESSION privilege; logon denied ..	4-1

ORA-12514: TNS:listener does not currently know of service requested in connect descriptor....	4-2
ORA-12516: TNS:listener could not find available handler with matching protocol stack	4-2
ORA-12541: TNS:no listener	4-2
GSM-45034: Connection to GDS catalog is not established	4-2
GSM-45054: GSM error or NET-40006: unable to start GSM	4-2
Solutions for General Issues	4-2
Connecting to GDS Configuration Databases When No Global Service Managers Are Running	4-2
Connecting to Catalog Databases When No Global Service Managers Are Running	4-3
Obtaining the Running Status of Global Data Services Components	4-3
Viewing Static Configuration Information for Global Data Services Components	4-4
Enabling and Disabling Tracing on a Global Service Manager	4-4
Using Global Service Manager Log and Trace Files	4-4
Using SYS_CONTEXT Parameters in a GDS Environment	4-5

A Global Data Services Control Utility (GDSCTL) Reference

About GDSCTL	A-2
GDSCTL Overview	A-3
Operational Notes	A-4
Special Topic Discussion	A-6
GDSCTL Reference	A-7
add.....	A-10
add brokerconfig.....	A-10
add database	A-11
add gdspool	A-12
add gsm	A-13
add invitednode (add invitedsubnet)	A-15
add region	A-16
add service	A-16
config.....	A-22
config database.....	A-22
config gdspool	A-23
config gsm	A-24
config region	A-24
config service	A-25
config vncr.....	A-26
configure.....	A-28
connect	A-29
create catalog.....	A-30
delete catalog	A-32
disable service.....	A-33
enable service.....	A-34
export catalog.....	A-35
import catalog.....	A-36
modify	A-37
modify catalog.....	A-37

modify database	A-38
modify gdsPOOL	A-39
modify gsm	A-40
modify region	A-41
modify service	A-41
relocate service.....	A-49
remove	A-51
remove brokerconfig	A-51
remove database.....	A-51
remove gdsPOOL.....	A-52
remove gsm.....	A-53
remove invitednode (remove invitedsubnet)	A-53
remove region.....	A-54
remove service.....	A-54
services.....	A-56
set.....	A-58
set gsm	A-58
set inbound_connect_level	A-58
set log_level.....	A-59
set outbound_connect_level	A-59
set trace_level.....	A-60
start.....	A-62
start gsm	A-62
start service	A-62
status	A-64
status database	A-64
status gsm.....	A-65
status service.....	A-66
stop	A-67
stop gsm	A-67
stop service.....	A-67
sync brokerconfig.....	A-69
sync database	A-70
validate catalog.....	A-71

Glossary

Index

Preface

Oracle Database Global Data Services provides dynamic load balancing, failover, and centralized service management for a set of replicated databases that offer common services. The set of databases can include Oracle Real Application Clusters (Oracle RAC) and noncluster Oracle databases interconnected through Oracle Data Guard, Oracle GoldenGate, or any other replication technology.

This Preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for database administrators, Architects, and engineers who manage replicated databases.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Database 12c release 1 (12.1) documentation set:

- *Oracle Data Guard Broker*
- *Oracle Data Guard Concepts and Administration*
- *Oracle Database Net Services Administrator's Guide*

- *Oracle Database Net Services Reference*
- *Oracle Real Application Clusters Administration and Deployment Guide*

Conventions

The following text conventions are used in this document:

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

Changes in This Release for Oracle Database Global Data Services Concepts and Administration Guide

This preface contains:

- [Changes in Oracle Database 12c Release 1 \(12.1.0.2\)](#)

Changes in Oracle Database 12c Release 1 (12.1.0.2)

The following are changes in *Oracle Database Global Data Services Concepts and Administration Guide* for Oracle Database 12c Release 1 (12.1.0.2).

New Features

The following features are new in this release:

- Support for administrator-managed Oracle RAC databases

In this release Global Data Services supports both administrator-managed and policy-managed Oracle RAC databases.

- Support for Oracle GoldenGate

Global Data Services has had support for Oracle GoldenGate starting from the initial release of Oracle Database 12c; however, at that time there was no version of Oracle GoldenGate that could work with Oracle Database 12c. The limitation with using Oracle GoldenGate with Global Data Services is that role-based services and tracking of replication lag are not supported.

See "[Global Data Services Use Cases](#)" on page 1-17, for new use cases including Oracle GoldenGate.

- Support for export and import of data stored in the GDS catalog.

The Global Data Services administrator can create a logical backup of the GDS catalog by exporting the catalog to a file. The backup can be used to restore the catalog after a failure or to move the catalog to another database.

Introduction to Global Data Services

This chapter provides an overview of the Global Data Services architecture.

This chapter includes the following topics:

- [Introduction to Global Data Services](#)
- [Global Data Services Architecture](#)
- [Overview of Global Services](#)
- [Global Connection Load Balancing](#)
- [Global Run-Time Connection Load Balancing](#)
- [Global Services Failover](#)
- [Global Data Services Use Cases](#)

Introduction to Global Data Services

Many enterprises consolidate their information technology infrastructure to improve business efficiency. Database consolidation is a critical part of this process. However, most organizations must still maintain local and remote replicas of their databases for reasons that include:

- Business continuity and disaster recovery
- High availability
- Performance optimization for local clients
- Content localization and caching
- Compliance with local laws

In any set of database replicas, some database servers may have a slow query response time because of high load or high network latency, while other servers capable of offering a faster response time may be under-utilized. Optimal query performance and resource utilization across a set of database replicas requires a workload management solution that provides dynamic load balancing of client connections and workload requests across the replicas.

Many enterprises use a home-grown solution for workload management across database replicas. These solutions cannot provide critical functionality such as run-time load balancing and reliable database service failover between replicas because they are not fully integrated with Oracle software.

Oracle Database provides a powerful workload management feature called database services. Database services are named representations of one or more database

instances. Database services allow you to group database workloads, ensure that client requests are routed to the optimal instance that offers a service, and provide high availability by transparently failing client connections over to surviving instances when a planned or unplanned instance outage occurs.

Oracle Global Data Services (GDS) implements the Oracle Database service model across a set of replicated databases known as a **Global Data Services configuration**.

A Global Data Services configuration looks like a virtual multi-instance database to database clients. It provides client access through **global services**, which are functionally similar to the local database services provided by single-instance or Oracle Real Application Clusters (Oracle RAC) databases. Local and global services both provide load balancing, high availability, and resource management. The essential difference between global services and local services is that global services span the instances of multiple databases, whereas local services span the instances of a single database.

A Global Data Services configuration and its global services are created and managed using the **GDSCTL** command-line interface, which is similar to the **SRVCTL** command-line interface used to manage an Oracle RAC database and its services.

A Global Data Services configuration can consist of any combination of multi-instance or single-instance Oracle databases hosted on heterogeneous or non-heterogeneous server platforms. Oracle Data Guard, Oracle GoldenGate, or any other database replication technology, can be used to synchronize the databases in a Global Data Services configuration.

Global Data Services is a highly effective solution for automatic workload management across a set of replicated databases, whether used with many widely distributed databases and clients or with a single database, a local replica and a few clients.

Note: Global Data Services is primarily intended for applications that are replication-aware. A replication-aware application is one that has any of the following characteristics:

- Uses read-only global services exclusively
- Uses read/write global services, but is programmed to resolve update conflicts if those services are simultaneously offered by more than one database
- Can tolerate replicated data that is potentially stale due to replication lag

Applications that are not replication-aware can still benefit from the improved high availability and disaster recovery capabilities provided by Global Data Services.

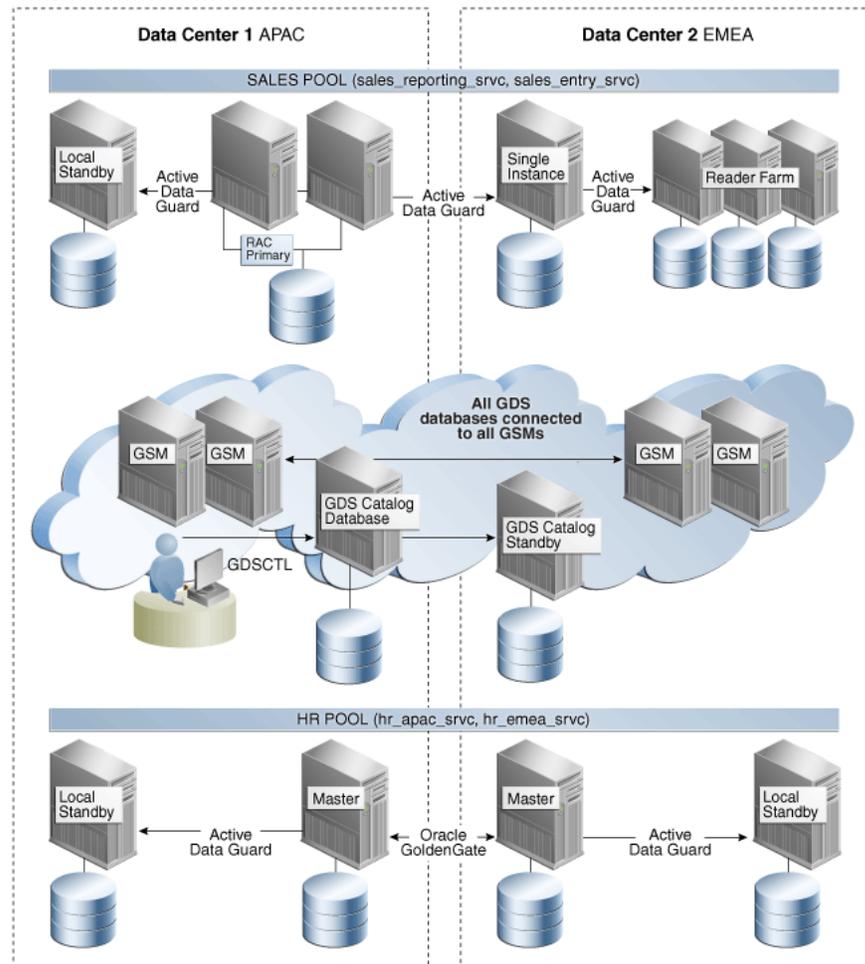
See Also:

- [Appendix A, "Global Data Services Control Utility \(GDSCTL\) Reference,"](#) for more information about GDSCTL
- *Oracle Database Administrator's Guide* for more info about database services

Global Data Services Architecture

Figure 1–1 shows an example of a Global Data Services (GDS) configuration and the GDS components that are present in every GDS configuration. The sections that follow describe these components.

Figure 1–1 Global Data Services Components



This section describes the Global Data Services components:

- [Global Data Services Pool](#)
- [Global Data Services Region](#)
- [Global Service Manager](#)
- [Global Data Services Catalog](#)
- [Oracle Notification Service Servers](#)

Global Data Services Pool

A **Global Data Services pool** is a named subset of databases within a GDS configuration that provides a unique set of global services and belongs to the same administrative domain. Partitioning of GDS configuration databases into pools

simplifies service management and provides higher security by allowing each pool to be administered by a different administrator.

A database can only belong to a single Global Data Services pool. All databases in a pool need not provide the same set of global services. However, all databases that provide the same global service must belong to the same pool.

Global Data Services Region

A **Global Data Services region** is a named subset of databases in a GDS configuration and database clients that share network proximity such that the network latency between members of a region is typically lower than between members of different regions. A region usually corresponds to a local area or metropolitan area network (LAN or MAN). For example, a data center hosting one or more GDS configuration databases and database clients in geographical proximity to the data center might belong to the same region.

A region can contain multiple Global Data Services pools, and these pools can span multiple regions.

For high availability purposes, each region in a GDS configuration should have a designated **buddy region**, which is a region that contains **global service managers** that can provide continued access to a GDS configuration if the global services managers in the local region become unavailable.

Global Service Manager

A **global service manager** is the central software component of Global Data Services, providing service-level load balancing, failover, and centralized management of services in the GDS configuration. Global Data Service clients use a global service manager to perform all GDS configuration operations.

A global service manager is analogous to the remote listener in an Oracle RAC database, except that a global service manager serves multiple databases. A global service manager does the following:

- Acts as a regional listener that clients use to connect to global services
- Provides connect-time load balancing for clients
- Manages global services across the regions of a GDS configuration
- Collects performance metrics from databases in the GDS configuration and measures network latency between regions of a configuration
- Creates a run-time load balancing advisory and publishes it to client connection pools
- Monitors availability of database instances and global services and notifies clients if they fail.

A global service manager is associated with one and only one GDS configuration. Each region in the GDS configuration must have at least one global service manager. It is recommended that multiple global service managers be configured in each region to improve availability and performance. Every global service manager in a GDS configuration manages all global services supported by the configuration.

Global Data Services Catalog

A **Global Data Services catalog** is a repository that stores configuration data for a Global Data Services configuration and all global services provided by that configuration.

A catalog is associated with one and only one GDS configuration. A catalog must reside in an Oracle Database 12c or later database, and that database may be inside or outside the associated GDS configuration. For large-scale GDS configurations, it is recommended that the GDS catalog be hosted outside the databases in the GDS configuration.

GDS catalog may be co-hosted along with catalogs of RMAN or Oracle Enterprise Manager.

Oracle strongly recommends that high availability technologies such as Oracle RAC, Oracle Data Guard, and Oracle Clusterware be used to enhance the availability of the database where the Global Data Services catalog resides.

Oracle Notification Service Servers

GDS clients use **Oracle Notification Service (ONS)** to receive run-time load balancing advisory and high availability events from global service managers.

An Oracle Notification Service (ONS) server is co-located with each global service manager. All such ONS servers within a region are interconnected. Clients of global services subscribe to the ONS server network within their region and its buddy region, and receive FAN notifications from those ONS server networks.

Note: An Oracle RAC database in a Global Data Service configuration may also contain ONS servers running on the cluster nodes. These ONS servers generate FAN notifications related to local services and are not connected to ONS server networks in the GDS regions.

Overview of Global Services

For database clients, a Global Data Services configuration is represented by a set of global services. A global service manager serving a Global Data Services configuration is aware of all global services that the configuration provides and acts as a mediator for the database clients and databases in the configuration. A client program connects to a regional global service manager and requests a connection to a global service. The client does not need to specify which database or instance it requires. The global service manager forwards the client's request to the optimal instance in the configuration that offers the global service. Database clients that share a global service must have the same service level requirements.

The functionality of *local* services, defined as traditional database services provided by a single database, is not changed by global services. Oracle Database 12c can provide local and global services, simultaneously. A client application can also work with global and local services, simultaneously.

Note: Database versions earlier than Oracle Database 12c can provide local services, but only Oracle Database 12c, and later, can provide global services.

The configuration and run-time status of global services are stored in the Global Data Services catalog. Each database that offers global services also maintains information about those services in a local repository (such as a system dictionary or Oracle Cluster Registry), with data on local services. Global services that are stored in a local repository have a flag to distinguish those global services from traditional local services.

If you are locally connected to a particular database, then you can query data on global services provided by that database. You can configure, modify, start, or stop a global service using the Global Data Services Control utility (GDSCTL) when you are connected to the Global Data Services catalog. This ensures centralized coordinated management of global services. You cannot configure, modify, start, or stop a global service using either the Server Control utility (SRVCTL) or the Oracle Clusterware Control utility (CRSCTL).

Note: Under certain circumstances (such as patching a database or clusterware software), you can stop or start global services using SRVCTL with the `-force` option with the appropriate command. You must have the appropriate system privileges.

After you configure global services for a Global Data Services configuration, the global service manager manages the global services on GDS configuration databases according to service properties that you specify when you create the services.

When a database joins the configuration or restarts after a shutdown, the database registers with all global service managers in the configuration. After receiving the registration request, one of the global service managers queries the GDS catalog and checks whether all global services which this database is supposed to provide are created there and have the correct attributes. If there is a discrepancy between the information in the catalog and database, the global service manager may create, delete, or modify some global services in the database, or change their attributes to synchronize them with the information in catalog. Then the global service manager determines which global services need to be running on the database and starts them if necessary.

The global service manager can start or stop a global service in a database. However, if it is an Oracle RAC database, the global service manager does not control which particular instances within the database offer the service. This is controlled by the clusterware and the administrator of the Oracle RAC database.

When a database instance in a Global Data Services configuration fails, all global service managers in the configuration get notified about the failure and stop forwarding requests to the instance. If this instance belongs to a noncluster database, or the instance is the last instance that was available in an Oracle RAC database, then, depending on the configuration, a global service manager can automatically start the service on another database in the Global Data Services pool where the service is enabled. If you decide to manually move a global service from one database to another using the appropriate GDSCTL command, then the global service manager stops and starts the service on the corresponding databases.

Global Service Attributes

Global services have a set of attributes associated with them that control starting the global services, load balancing connections to the global services, failing over those connections, and more. Attributes applicable to local services, including those services

specific to Oracle RAC and Oracle Data Guard broker environments, are also applicable to global services.

The following attributes are unique to global services:

- Preferred or available databases
- Replication lag
- Region affinity

You can modify global services just as you can modify local services. You can enable and disable a global service, you can move the global service to a different database, and you can change the properties of the global service.

Note: You cannot upgrade a local service to a global service.

Global Services in an Oracle RAC Database

Some properties of a global service are only applicable to Oracle RAC databases and are unique for each Oracle RAC database included in a Global Data Services configuration. These properties are related to the placement of global services with instances within an Oracle RAC database, including server pools and service cardinality.

All other existing global service attributes, such as load balancing, role, transparent application failover parameters, and database edition must be the same for all databases offering a global service. Specify these attributes at the global service level.

Notes: By default, in an Oracle RAC environment, a SQL statement executed in parallel can run across all of the nodes in the cluster. The cross-node parallel execution is not intended to be used with GDS load balancing. For an Oracle RAC database in a GDS configuration Oracle recommends that you restrict the scope of the parallel execution to an Oracle RAC node by setting the `PARALLEL_FORCE_LOCAL` initialization parameter to `TRUE`.

Global Services in an Oracle Data Guard Broker Configuration

Oracle Data Guard broker permits a primary database to have up to 30 standby databases. Oracle Data Guard broker logically groups primary and standby databases into a *broker configuration* that enables the broker to manage and monitor the databases as an integrated unit. When you include a broker configuration in a GDS configuration, you manage the broker configuration as one unit. Only an entire broker configuration can be included in (or deleted from) a GDS pool, and a broker configuration cannot span multiple pools.

If you attempt to add or remove a database that belongs to a broker configuration to or from a GDS pool, then an error occurs. You can only add a database to the GDS pool by adding the database to the broker configuration using the Oracle Data Guard command-line interface, DGMGRL. After you add a database to the broker configuration, you must run the `gdctl sync brokerconfig` command to synchronize GDS and Oracle Data Guard.

A GDS pool can consist of a set of databases of a given Data Guard broker configuration. Databases that belong to different Data Guard broker configurations must be mapped to different GDS pools. A pool that contains a Data Guard configuration cannot have databases that do not belong to the configuration

See Also: "[sync brokerconfig](#)" on page A-69 for information about this command

Conversely, when you remove a database from a broker configuration, the database is removed from the GDS pool to which this broker configuration belongs. The `gdsctl sync brokerconfig` command must be executed after removing a database. This is the only way to remove a database from a pool that contains a broker configuration.

You can configure global services with a role attribute to be active in a specific database role, such as the role of primary or physical standby database. If you enable fast-start failover, then the Oracle Data Guard broker automatically fails over to a standby database if the primary database fails. The global service managers that you have configured to work with Oracle Data Guard broker ensure that the appropriate database services are active and that the appropriate Fast Application Notification (FAN) events are published after a role change. These FAN events enable Fast Connection Failover (FCF) of client connections to an appropriate database instance within the GDS configuration.

The Global Data Services framework supports the following Oracle Data Guard broker configurations:

- The set of databases in a Global Data Services pool can be either the set of databases that belong to a single broker configuration or a set of databases that do not belong to a broker configuration. You can add a broker configuration only to an empty Global Data Services pool. If a GDS pool already contains a broker configuration, then, to add a database to the pool, you must add the database to the broker configuration contained in the pool.
- Role-based global services are supported only for database pools that contain a broker configuration.

See Also: *Oracle Data Guard Broker* for more information about broker configuration

Database Placement of a Global Service

You can specify which databases support a global service. These databases are referred to as **preferred databases**. The global service manager ensures that a global service runs on all preferred databases for which the service has been specified. The number of preferred databases is referred to as the **database cardinality** of a global service. You must specify at least one preferred database for a global service.

Note: If you set `preferred_all` for which databases support a service, then you do not have to explicitly specify preferred or available databases. The `preferred_all` setting implies that all databases in the pool are preferred.

When you add or modify a global service, you can specify a list of available databases for this global service. If one of the preferred databases fails to provide a global service, then the global service manager relocates the service to an available database to maintain the specified database cardinality for that service.

In a Global Data Services pool that contains an Oracle Data Guard broker configuration, a role-based global service can be started on a database only if the database is listed as preferred or available for the service and the `role` attribute of the database corresponds to the `role` attribute specified for the service. For example, a global service that can run on any database in a broker configuration (as long as the

role attribute of the database is set to `primary`) must have `primary` specified for its role attribute and have all other databases in the broker configuration with role attributes set to `preferred`.

Note: Do not confuse *database cardinality* of global services with their *instance cardinality*. Instance cardinality is specified and maintained separately for each Oracle RAC database and is not maintained across databases of a Global Data Services pool.

For example, consider a case when there are few instances of an Oracle RAC database offering a global service and one of the instances fails, but there are no available instances on which to start the global service. In this case, the service is not started on an available database instance of another Oracle RAC database. However, if the instance that failed was the last instance in the database offering the service, then the service may start on another database that is listed as available for this service.

Replication Lag and Global Services

For performance reasons, distributed database systems often use asynchronous replication of data between databases, which means that there is the possibility of a delay between the time an update is made to data on a primary database and the time this update appears on a replica. When this happens, the replica database lags behind its primary database.

Global Data Services enables applications to differentiate between global services that provide real-time data from services that can return out-of-date data because of replication lag. For applications that can tolerate a certain degree of lag, you can configure a maximum acceptable lag value.

For applications that cannot tolerate replication lag, you can set the lag time for global services to zero. Requests for this global service are forwarded only to a primary database, or to a replica database that is synchronized with the primary database.

See Also: [Chapter 3, "Administering Global Data Services Configurations"](#) for more information

Note: Only an Oracle Data Guard standby database can be synchronized with the primary database, if necessary, so that you can enable real-time, read-only services on both the primary and synchronized standby databases.

For many applications it is acceptable to read stale data as long as the data is consistent. Such applications can use global services running on any database replica without regard to the length of the replication lag time.

If you configure the lag time for a service to a value other than zero, then a client request can be forwarded only to a replica database that is not lagging behind the primary database by longer than the configured lag time for the service. If there is no such database, then the connection request fails.

Note: Specification of the maximum replication lag is only supported for Active Data Guard configurations.

If a service that is currently running starts to exceed the specified maximum lag, then the service is brought down after all current requests are completed. New requests for the service are routed to a database that meets the configured lag value for the service.

Note: If you use Oracle Data Guard with the Oracle Active Data Guard option, then you can use the `STANDBY_MAX_DATA_DELAY` session parameter to specify a session-specific lag tolerance. When you set this parameter to a nonzero value, a query issued to a physical standby database is executed only if the application lag time is less than or equal to the `STANDBY_MAX_DATA_DELAY` parameter value. Otherwise, the database returns an error to alert the client that the lag time is too long. If both session-level and service-level parameters are set and the `STANDBY_MAX_DATA_DELAY` parameter value is less than the service level parameter value, the client remains connected to the database, but queries would return an error.

Caution: If a database fails, then the global service manager can route a client connection to another database that meets the maximum lag value specified for the service, even if it lags behind the failed database. This solution, however, can create data consistency problems for some applications.

See Also: *Oracle Data Guard Concepts and Administration* for more information about the `STANDBY_MAX_DATA_DELAY` session parameter

Global Services and Distributed Transaction Processing

When performing XA (distributed) transactions against a GDS configuration, tightly-coupled branches of an XA transaction must be directed to the same database. This can be accomplished by ensuring that each global service used in a distributed transaction is, that is runs exactly on one database at a time. It is responsibility of the GDS pool administrator to configure and maintain the property of global services used in distributed transactions.

If a global service used in an XA transaction is running on a multi-instance Oracle RAC database, tightly-coupled branches of the XA transaction by default can be sent to separate database instances. This may result in sub-optimal performance. To avoid the performance degradation in this case, it is highly recommended to set the DTP parameter (`-dtp`) to TRUE for any service used by an XA transaction that can run in an Oracle RAC database. Setting the parameter guarantees that all distributed transactions performed through the service have their tightly-coupled branches running on a single Oracle RAC instance.

Note: The DTP parameter of a global service only affects processing of XA transactions on an Oracle RAC database that provides this service. It has no effect on the database cardinality of the global service. A global service used by a distributed transaction must be manually configured to be active on a single database at a time.

Global Services in Multitenant Architecture

Multitenant container databases (CDB) can be part of the GDS configuration. A global service can be defined at root level of the CDB (CDB\$ROOT) or at the pluggable database (PDB) level. A global service that runs on a PDB can be created by setting the PDB property to the PDB name at creation time. The PDB property of a global service cannot be modified once the service is created. If the PDB property is not set, the service is created at the CDB root level (CDB\$ROOT).

When the service is created at the PDB-level all of the service's preferred and available databases should contain the PDB it belongs to. Global services defined at the PDB level are managed by the same operations and rules as the services defined at any other level.

Global Connection Load Balancing

When a client connects to an Oracle RAC database using a service, the client can use the Oracle Net connection load balancing feature to spread user connections across all the instances that support that service. Similarly, in a Global Data Services configuration, clients connecting to a global service are load balanced, as necessary, across different databases and regions.

This section includes the following topics:

- [Client-Side Load Balancing](#)
- [Server-Side Load Balancing](#)
- [Region Affinity for Global Services](#)

Client-Side Load Balancing

Client-side load balancing balances the connection requests across global service managers. Connection failover is also supported. With connection failover, if an error is returned from the chosen global service manager, then Oracle Net Services tries the next address in the address list until either a successful connection is made or Oracle Net Services has exhausted all the addresses in the list.

Client-side load balancing and failover in a Global Data Services configuration is similar to that for an Oracle RAC database. In a Global Data Services configuration, however, a client in a Global Data Services region first tries to connect to any of the global service managers in its local region. If a global service manager from the local region does not respond, then the client tries a global service manager in another region.

To enable client-side load balancing and failover across multiple regions in a Global Data Services configuration, clients must use an Oracle Net connect descriptor that contains one list of addresses of local regional global service managers for load balancing and intraregion failover, and one (or more) list of addresses of remote regional global service managers for interregion failover. If a region is not specified, it defaults to the region name of the global service manager to which the client is connected. You can also configure timeout and retry attempts for each list to enable multiple connection attempts to the current global service manager before moving to another global service manager in the list.

See Also: ["Database Client Configuration"](#) on page 2-11 for more information

Server-Side Load Balancing

Server-side load balancing for an Oracle RAC database has the listener directing connection requests to the best Oracle RAC database instance. Some applications have long-lived connections, while other applications have short-lived connections.

For global services, server-side load balancing works similarly, except that, instead of being limited to a single database, workloads are balanced across multiple databases in the Global Data Services configuration. In most cases, a global service manager directs a client request for a global service to a database server in the same region, unless all local servers are overloaded and a remote server can provide significantly better quality of service.

In some cases, to take advantage of data caching on a local server, you can direct requests to the local region. Global Data Services enables you to specify a desired level of client/server affinity for a global service.

Region Affinity for Global Services

You can configure global services to operate within specific regions or in any region in the Global Data Services configuration. This is called **region affinity**.

This section includes descriptions for the following types of region affinity:

- [Any-Region Affinity](#)
- [Affinity to a Local Region](#)
- [Affinity to a Local Region with Interregion Failover](#)

Any-Region Affinity

Any-region affinity (the default) for a global service routes a client connection request to the best database in the Global Data Services configuration, regardless of region, that can meet the connection load balancing goal for the requested service. The choice of the database is based on its performance and network latency between the regions where the client and database reside. If databases in different regions are equally loaded, then this policy gives preference to a local region. An interregion connection is made only if the difference in database performance between regions outweighs network latency.

If you specify preferred, available databases for a global service with any-region affinity, then service cardinality is maintained at the Global Data Services pool level. If a service fails on a preferred database, then the service is started on any available database in the Global Data Services pool, and the number of service providers in the pool remains the same. When starting the service on an available database, databases in the region where the service failed have preference. If there is no available database for this service in the local region, then an available database is chosen from a nearby region.

Affinity to a Local Region

Affinity to a local Global Data Services region for a global service routes a client connection request to the best database in the client's region that can meet the connection load balancing goal for the requested service. The global service manager chooses the database based only on the database's performance. A global service with affinity to a local region can be provided in multiple Global Data Services regions at the same time, but a client from one region is never connected to a database in another region.

If you specify preferred or available databases for a global service with local region affinity, then service cardinality is maintained at the regional level. If a service fails on a preferred database, then the service starts on an available database in the same region, so the number of service providers in the region remains the same. If there is no available database for this global service in the local region, then no action is taken and the service cardinality decreases. If there is no database in the local region offering the global service, then the client connection request is denied. For global services that have affinity to a local region, database cardinality of the service is maintained independently in each region.

Affinity to a Local Region with Interregion Failover

This type of affinity is similar to that of affinity to a local Global Data Services region, except that, if there are no databases in the local region offering a global service, then, instead of denying a client request, the request is forwarded to the best database in another region where the requested global service is running. This service failover does not trigger the service to be started on an available database in another region because, with affinity to a local region, database cardinality is maintained independently in each region, and should not change because of service failure in another region. If regional databases become overloaded because of interregion failover, then you can manually add a preferred database for the service.

Global Run-Time Connection Load Balancing

Global run-time connection load balancing distributes client work requests across persistent connections that span the instances of an Oracle RAC database, based on load-balancing information from the database. This feature is supported with connection pools that can receive load balancing recommendations. The database uses the run-time connection load balancing goal for a service and the relative performance of the database instances to generate a recommendation about where to direct service requests.

There are two types of service-level goals for run-time connection load balancing:

- **SERVICE_TIME**: Attempts to direct work requests to instances according to response time. Load-balancing data is based on elapsed time for work done in the service plus available bandwidth to the service.
- **THROUGHPUT**: Attempts to direct work requests according to throughput. The load-balancing data is based on the rate that work is completed in the service plus available bandwidth to the service.

The Global Data Services framework also supports the balancing of work requests at run time to a global service. In the Global Data Services framework the requests are spread across connections to instances in multiple databases. Work is routed to provide the best service times globally and routing responds gracefully to changing system conditions.

To provide global run-time connection load balancing, a global service manager receives performance data for each service from all database instances in the Global Data Services configuration. The global service manager also measures interregion network latency by periodically exchanging messages with global service managers in other Global Data Services regions.

If the load-balancing goal for a global service is set to `SERVICE_TIME`, then a global service manager considers interregion network latency and instance performance data when deciding how to distribute work requests. For example, clients in Region A have run-time load balancing metrics that are weighted toward Region A, and clients in

Region B have metrics that are weighted toward Region B. This implies that, even though the service may be the same, clients in different regions receive different run-time load balancing metrics.

If the load-balancing goal for a global service is set to `THROUGHPUT`, then run-time load balancing metrics are calculated only based on the performance of database instances.

In addition to calculating run-time load balancing metrics for local clients, a global service manager may also need to calculate run-time load balancing metrics of remote regions and publish the metrics for clients residing in a region where all global service managers are not available.

Affinity

The global service manager provides advice about how to direct incoming work to the databases in a Global Data Services configuration that provide the optimal quality of service for that work. The load balancing advisory also sends additional affinity information that indicates to the clients, subscribers to the Oracle Net Service events (as Universal Connection Pool (UCP) and ODP.NET Connection Pool, and so on), whether it should re-connect to the same database. Reconnecting a session to the same database can improve buffer cache efficiency and lower CPU usage and transaction latency. The affinity information indicates whether affinity is active or inactive for a particular database. It is always inactive for a single instance database and can be either active or inactive for a particular instance if the database is an Oracle RAC.

Disk I/O and CPU Thresholds

Disk I/O and CPU thresholds are resource consumption limits that a database administrator can set. The disk I/O threshold is configured by setting the single-block read latency limit, and the CPU threshold is set by setting the CPU usage limit. The global service manager monitors the resource usage and checks whether the I/O and CPU thresholds have been reached or not. When a database reaches one of the thresholds, the load balancing advisory readjusts its advice about how to direct incoming work to this database.

Global Services Failover

When a global service or database fails, a global service that was running on the database fails over to another database where the global service is enabled, but not yet running (if the database role matches the service role). The global service manager considers preferred databases as the failover target before available databases.

If you stop a global service using `GDSCCTL`, then the service does not fail over to another database. However, the database where the service was stopped remains a failover target for this service. If the service fails on another database, then the service can start on that database.

When a global service fails over to an available database, the Global Data Services framework does not move the service back to the preferred database when the preferred database restarts because of the following:

- The service has the desired cardinality
- Maintaining the service on the current instance or database provides a higher level of service availability
- Not moving the service back to the initial preferred instance or database prevents a second outage

If necessary, you can manually relocate the global service back to the preferred database after the service has restarted, without terminating active sessions.

In a Global Data Services pool that contains an Oracle Data Guard broker configuration, the Global Data Services framework supports role-based global services. Valid roles are `PRIMARY`, `PHYSICAL_STANDBY`, `LOGICAL_STANDBY`, and `SNAPSHOT_STANDBY`. The Global Data Services framework automatically starts a global service only when the database role matches the role specified for the service.

If a database switches roles or fails, then the Oracle Data Guard broker notifies the Global Data Services framework about the role change or failure, and the global service manager ensures that the services start according to the new database roles.

Note: A global service cannot fail over from a database in one Global Data Services region to a database in another region if the locality parameter is set to `LOCAL_ONLY`, and interregion failover is not enabled.

When a global service fails over, fast connection failover, if enabled on Oracle clients, provides rapid failover of the connections to that global service. The Global Data Services framework, similar to Oracle RAC, uses Fast Application Notification (FAN) to notify applications about service outages. Instead of waiting for the application to poll the database and detect problems, clients receive FAN events and react, immediately. Sessions to the failed instance or node are terminated, and new connections are directed to available instances providing the global service.

All global service managers monitor service availability on all databases in a Global Data Services configuration. A global service is started on an available database when a global service manager detects that a global service becomes unavailable due to a failure.

Note: A global service manager cannot automatically fail over a service if it cannot connect to the Global Data Services catalog.

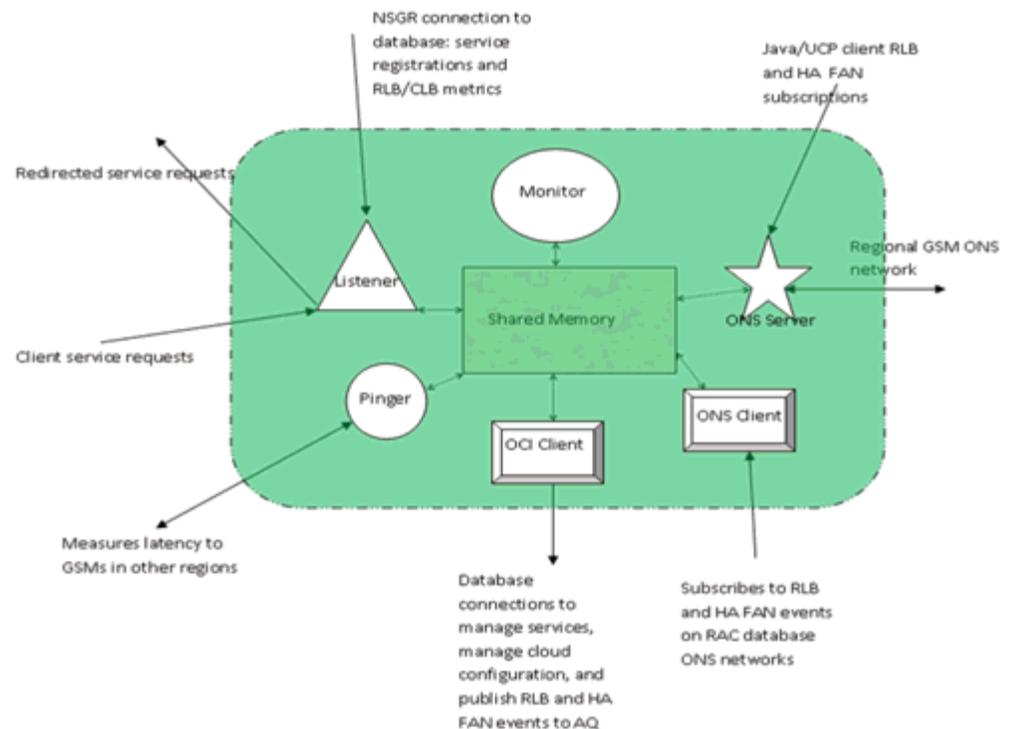
Note: When the target database for failover is under Oracle Clusterware control, the global service manager failover request to start a service on this database may lead to a Oracle Clusterware “pull up” event that starts another instance. This will occur when at least one instance is registered with the global service manager listener, allowing the global service manager to note that the database is running, and is therefore a valid target for failover, but that the service is not already running on the database. This can occur when the registered instance is not a primary or valid instance for the service, either because it is an available instance on an administrator-managed Oracle RAC database, or because it is an instance of a policy-managed Oracle RAC database that is not in the same server pool as the service.

Global Service Manager Process Suite Architecture

The global service manager process suite architecture is a process suite containing multiple processes that communicate using shared memory. Global service manager contains the following processes:

- **Monitor** - Monitors the health of all other processes in the process suite. Responsible for starting the other global service manager processes, creating the shared memory area and re-spawning a global service manager process after detecting its death.
- **Listener** - Listens for incoming client connection requests and forwards requests to appropriate database instances. Each instance in Global Data Services configuration establishes a communication channel to all global service manager listeners. This channel is used for registration of global services and sending performance metrics to the listener. These metrics are later used for connection-time and run-time load balancing. Detects when a database instance goes down. The listener process provides the full functionality of the regular database listener. Therefore a global service manager plays the role of a remote listener for all databases in a Global Data Services configuration. A global service manager listener routes connection requests directly to local database listeners, bypassing SCAN listeners on an Oracle RAC database.
- **Pinger** - Measures network latency between regions by exchanging data with pinger processes of global service managers in the other regions. Measurements are required for connection-time and run-time load balancing in a globally distributed database cloud. Monitors other global service managers and tells its own global service manager when it has to become the regional or master
- **OCI Process** - Responsible for establishing SQLNET connections to Global Data Services databases for managing services, and to Global Data Services catalog for managing the Global Data Services configuration.
- **ONS Server** - Publishes RLB and HA fan events to regional ONS network so that clients can subscribe to the events.
- **ONS Client** - Responsible for subscribing to HA events that are published by Oracle RAC databases in Global Data Services configuration.

Figure 1–2 shows the internal process architecture and the information flow in and out of the process suite.

Figure 1–2 Global Service Manager Process Suite Architecture

Global Data Services Use Cases

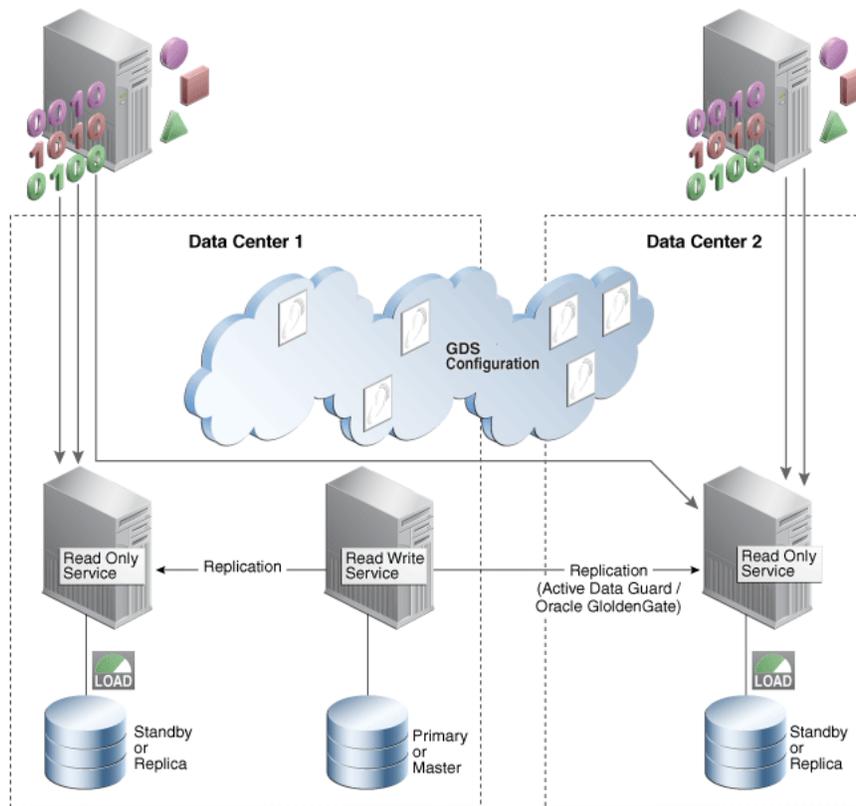
The following are some use cases for Oracle Global Data Services:

- [Load Balancing for Replicated Databases](#)
- [Service Failover for Replicated Databases](#)
- [Region Affinity in Oracle GoldenGate Multi-Master](#)
- [Load Balancing in Oracle GoldenGate Multi-Master](#)
- [Balancing Oracle Active Data Guard and Oracle GoldenGate Reader Farms](#)

Load Balancing for Replicated Databases

Global Data Services extends Oracle RAC-style connect-time and run-time load balancing capabilities (within and across data centers) to a set of replicated databases. The algorithm takes into account load metrics, region affinity, network latency, and load balancing goals. It maximizes performance and achieves efficient resource utilization by enabling Global Data Services on Active Data Guard.

Figure 1–3 shows load balancing across replicated databases, both local and remote, in a Global Data Services configuration. The Read Write Service runs on the Primary (or Master) database. Read Only Services run on the two Standby (or Replica) databases. Client connections are load balanced between the read-only services running on the Standby databases (across the two data centers).

Figure 1–3 Load Balancing for Replicated Databases

Service Failover for Replicated Databases

Global Data Services extends Oracle RAC-style service failover (within and across data centers) and management capabilities to replicated databases, and takes into account service placement policies. It achieves higher availability and improved manageability by enabling Global Data Services with Active Data Guard and Oracle GoldenGate.

Figure 1–4 shows databases replicated across two data centers, in a Global Data Services Configuration. A Read Write Service runs on the Primary (or Master) database. Read Only Services are load balanced across the two Standby (or Replica) databases.

Figure 1–4 depicts the failure of the Standby in Data Center 1. In Figure 1–5 Global Data Services fails over the Read Only Service to another available database (in this case the Primary) and load balances it with the Read Only Service running on the remote Standby in Data Center 2.

Figure 1-4 Read Only Service Failure

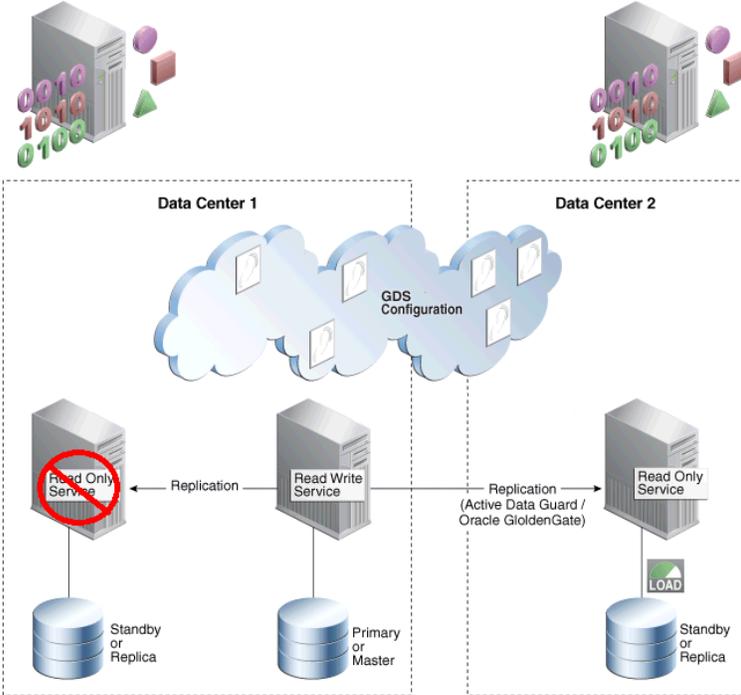
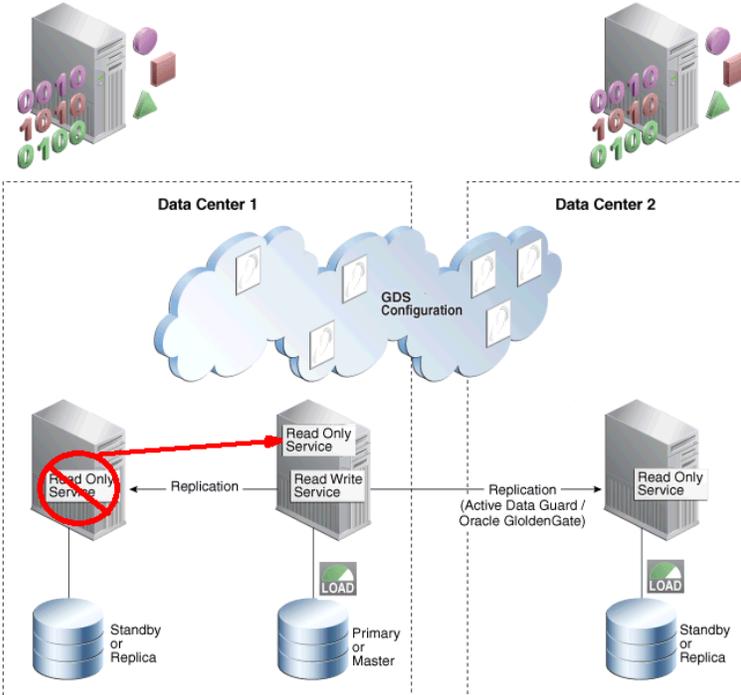


Figure 1-5 Read Only Service Failover



Region Affinity in Oracle GoldenGate Multi-Master

Figure 1-6 depicts Oracle GoldenGate multi-mastered replicas in a Global Data Services configuration. In this use case the application needs to avoid multi-master

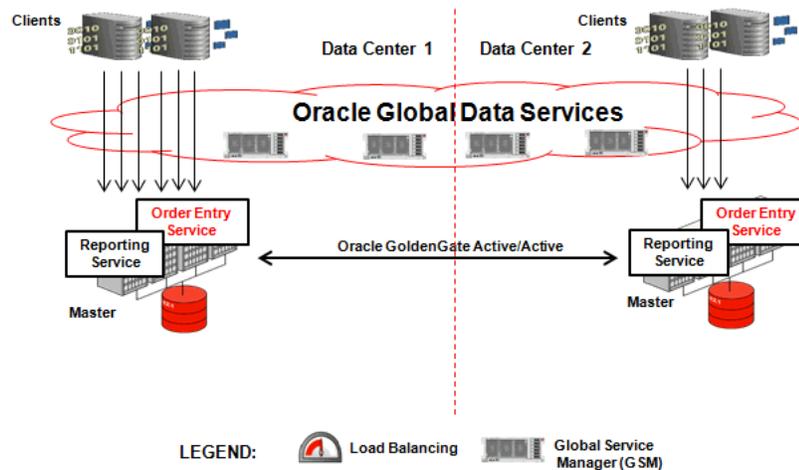
read/write conflicts, and Global Data Services can route all of the workloads to nearest and best replica in the client's region.

```
GDSCTL>add service -service reporting_srvc -gdspool sales
      -preferred_all -locality LOCAL_ONLY -region_failover
GDSCTL>add service -service order_entry_srvc -gdspool sales
      -preferred_all -locality LOCAL_ONLY -region_failover
```

This use case features the following:

- The application handles multi-master conflict resolutions.
- Read/write and read-only global services run on both replicas.
- Client connections are routed to the nearest and the best replica database in the client's region that meets the connection load balancing goal for the requested Order Entry and Reporting global services.
- Takes into account load metrics, region affinity, and load balancing goals.
- Global Data Services provides workload routing for read-only and read/write services for Oracle GoldenGate multi-master configuration.
- Routes all database workloads to the nearest datacenter by enabling Global Data Services for Oracle GoldenGate multi-master configuration.

Figure 1–6 Region Affinity in Oracle GoldenGate Multi-Master



Load Balancing in Oracle GoldenGate Multi-Master

Figure 1–7 depicts Oracle GoldenGate multi-mastered replicas in a Global Data Services configuration. In this use case the application needs to handle multi-master conflict resolutions, and Global Data Services provides connect-time and run-time load balancing (within and across data centers) for all work requests.

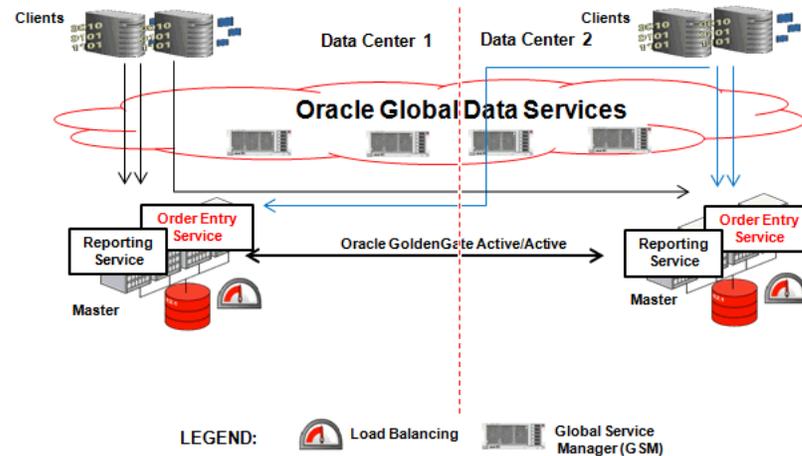
```
GDSCTL>add service -service order_entry_srvc -gdspool sales
      -preferred_all -clbgoal LONG
GDSCTL>add service -service reporting_srvc -gdspool sales
      -preferred_all -clbgoal LONG
```

This use case features the following:

- Application handles multi-master conflict resolutions.
- Read Write and Read Only global services run on both replicas.

- Client connections are load balanced between the READ WRITE global services running on both the masters (across data centers). Same is the case for the Read only service.
 - Takes into account load metrics, region affinity, network latency and load balancing goals.
 - Load balance all workloads by enabling GDS on Oracle GoldenGate multi-master configuration.

Figure 1–7 Load Balancing in Oracle GoldenGate Multi-Master



Balancing Oracle Active Data Guard and Oracle GoldenGate Reader Farms

Figure 1–8 depicts Global Data Services enabled for an Oracle Active Data Guard or Oracle GoldenGate reader farm with physical standby/replicas located in both local and remote data centers. In this use case with Global Data Services, route read/write workload to primary/master, balance read-only workload on the reader farm, and you can expect improved resource utilization and higher scalability for read workloads.

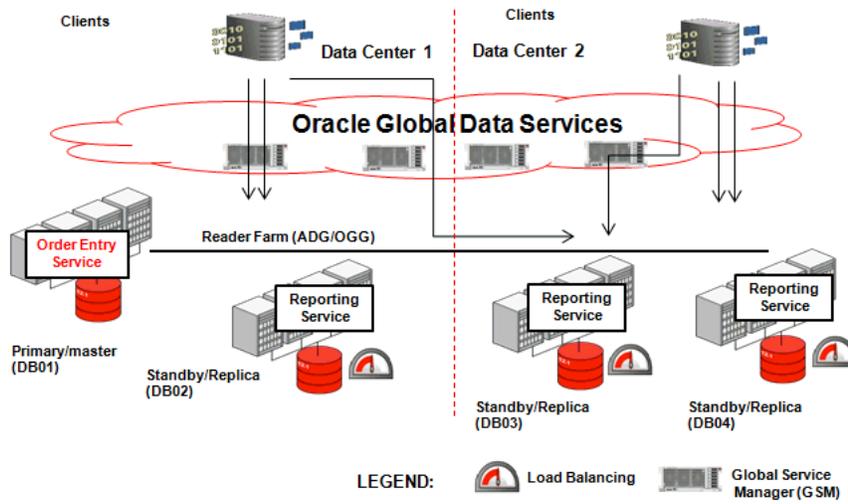
Example 1–1 Oracle GoldenGate

```
GDSCTL>add service -service reporting_srvc -gds pool sales
      -preferred DB02, DB03, DB04 -clbgoal LONG -rlbgoal SERVICE_TIME
```

This use case features the following:

- Read/write global service runs on the Primary/Master database.
- Read-only global services run on the reader farm.
- Client connections are load balanced between the read-only global services running on the reader farm (across data centers).
- Global Data Services provides connect-time and run-time load balancing (within and across data centers) on a reader farm and better resource utilization and higher scalability by balancing read-only workload on an Oracle Active Data Guard or Oracle GoldenGate reader farm.

Figure 1–8 Balancing Oracle Active Data Guard and Oracle GoldenGate Reader Farms



Configuring the Global Data Services Framework

The Global Data Services framework consists of at least one global service manager, a Global Data Services catalog, and the GDS configuration databases. Some components of the framework are installed when you install Oracle Database 12c. Other components require that you perform certain tasks using the Global Data Services control utility (GDSCTL).

See Also: [Appendix A, "Global Data Services Control Utility \(GDSCTL\) Reference"](#) to familiarize yourself with GDSCTL

This chapter includes the following topics:

- [Planning an Installation](#)
- [Installing a Global Service Manager](#)
- [Upgrading Global Data Services](#)
- [Creating the Global Data Services Catalog](#)
- [Adding a Global Service Manager to the Global Data Services Catalog](#)
- [Connecting to the Global Data Services Catalog](#)
- [Adding a Global Data Services Pool](#)
- [Adding a Global Data Services Region](#)
- [Adding a Database to a Global Data Services Pool](#)
- [Adding a Service to a Global Data Services Pool](#)
- [Starting a Global Service](#)
- [Database Client Configuration](#)
- [Exporting the GDS Catalog Data for Logical Backups](#)
- [Changing the GSMCATUSER Password](#)

Planning an Installation

Before you install any software, review these hardware, network, operating system, and other software requirements for Linux.

Global Data Services components require:

- At least 256MB of total physical memory

- At least 20MB of available physical memory
- At least 6 GB of total swap space
- At least 1.5 GB of free disk space
- Certified architecture, for example x86_64
- Linux system kernel version is at least "2.6.18"
- Package make-3.81 is available on the system
- Package binutils-2.17.50.0.6 is available on the system
- Package gcc-4.1.2 (x86_64) is available on the system
- Package libaio-0.3.106 (x86_64) is available on the system
- Package libaio-devel-0.3.106 (x86_64) is available on the system
- Package libstdc++-4.1.2 (x86_64) is available on the system
- Package sysstat-7.0.2 is available on the system
- Package compat-libstdc++-33-3.2.3 (x86_64) is available on the system
- Package libgcc-4.1.2 (x86_64) is available on the system
- Package libstdc++-devel-4.1.2 (x86_64) is available on the system
- Package glibc-devel-2.5 (x86_64) is available on the system
- Package gcc-c++-4.1.2 (x86_64) is available on the system
- Package glibc-2.5-58 (x86_64) is available on the system
- Package ksh-... is available on the system

Installing a Global Service Manager

The global service manager is the central component of the Global Data Services framework, and you must install the global service manager using separate media. No other Oracle software is required to install and run the global service manager.

You can install the global service manager on a system where you have other Oracle products installed, but you must install the global service manager in a separate Oracle home directory. You can install more than one global service manager on a single system, but each global service manager must have a separate Oracle home directory. For performance reasons, depending on the number of databases in your Global Data Services configuration, you may want to deploy the global service manager on a dedicated host.

You must install at least one global service manager for each Global Data Services region. Global service managers can be hosted on physical or virtual environments. For high availability, Oracle recommends installing multiple (typically 3) global service managers in each region running on separate hosts.

Note: Oracle Universal Installer does not currently support installing software on multiple hosts. You must install each global service manager on its respective host.

The Global Data Services administrator installs the global service manager. The Global Data Services administrator's responsibilities include:

- Administering global service managers
- Administering the Global Data Services catalog
- Administering regions and database pools

Note: The Global Data Services administrator must have an operating system user account on all hosts where global service managers are deployed, and you must run the installation under that user account. The installation must *not* be run by a root user.

Note: When you install the global service manager on Windows platforms, Oracle Universal Installer provides you with the option to use a Windows Built-in Account or to specify a standard Windows user account as the Oracle home user for the Oracle home. This account is used for running the Windows Services for the Oracle home and it cannot be an administrator account.

Note that LocalService account is used for running the services if Windows Built-in Account option is chosen. For administrative tasks, including global service manager installation, upgrade, and patching, you should log on using a Windows account that has administrative privileges.

See Chapter 3 of the *Oracle Database Platform Guide for Microsoft Windows* for details and recommendations regarding the use of Oracle home user.

To install a global service manager:

1. Start Oracle Universal Installer from the root directory of the software media and follow the prompts.

When the installation completes, the global service manager home directory contains binaries required to run the global service manager and the Global Service Manager Control utility (GDSCTL).

2. Set the ORACLE_HOME environment variable to the directory you specified during installation.
3. Add the \$ORACLE_HOME/bin directory created for the global service manager to the PATH environment variable.
4. Set the TNS_ADMIN environment variable set to \$ORACLE_HOME/network/admin.

Upgrading Global Data Services

There are four components that comprise the distributed Global Data Services infrastructure, and each component may reside in a separate installation and may be upgraded independently using the usual upgrade procedure; however, there are certain rules about component versioning that should be followed. The components and rules are as follows:

- **Catalog database:** The catalog database is the central repository for the GDS metadata; it is a standard Oracle Database installation. The version of the catalog database must always be greater than or equal to the version of any GDSCTL session that connects to it, and exactly equal to the version of any global service

manager server that connects to it, with one exception: to ease migration, the catalog may temporarily have a version greater than some global service manager servers that are connected to it, until any change is made to the catalog, at which time any connected global service manager that is not at the same version will be disconnected, and any stopped global service manager that is not at the same version will not be allowed to connect.

Note: GDSCTL sessions running release 12.1.0.1 cannot make changes to a later versioned catalog; when running commands that will update the catalog, the GDSCTL client should be at a minimum version of 12.1.0.2.

- **GDSCTL client:** This component is run in an ad-hoc manner from a terminal session on any system that contains a global service manager installation. The version of the GDSCTL client is the version of the global service manager installation that it runs from.

Note: When connecting a 12.1.0.1 GDSCTL client to a later versioned catalog only a limited set of commands are allowed, and any command that may cause catalog changes will result in a compatibility error. Commands that update the catalog metadata in a catalog at version 12.1.0.2 or later should be executed from a GDSCTL client running at least release 12.1.0.2.

- **Global service manager server:** The version of the global service manager server is the version of the global service manager installation from which the server runs. A global service manager server cannot connect to any catalog database that is at a lower version than itself. A global service manager server cannot connect to any catalog database that is at a higher version than itself in which changes have already been made to the catalog at that higher version. A global service manager can connect to a pool database running any version of Oracle Database 12c or later.
- **Pool database:** A pool database is any database added to a GDS pool which runs a global service. A pool database may be at any version of Oracle 12c or later, including versions later than the catalog version. You may upgrade or downgrade pool databases at any time.

Given these rules, it is possible to perform a rolling upgrade of the distributed GDS infrastructure with zero service downtime. The advised order of upgrade is:

1. Upgrade the catalog database. For best results this should be done using a rolling database upgrade; however, global services will remain available during the upgrade if the catalog is unavailable, although service failover will not occur.

Note: When upgrading Oracle Database 12c Release 1 with the 12.1.0.2 patch release, you must execute the following command after upgrading the catalog database:

```
modify catalog -pwd password -newkeys
```

This command generates encryption keys and encrypt existing GSMUSR passwords stored in the GDS catalog.

2. Upgrade global service manager installations that are used to run GDSCTL clients, which do not also run a global service manager server (if any).

Note: Global service manager upgrades should be done in-place; however, an in-place upgrade will cause erroneous error messages unless permissions on the following files for the following platforms are updated to 755:

On Linux/Solaris64/Solaris Sparc64:

```
$ORACLE_HOME/QOpatch/qopiprep.bat
$ORACLE_HOME/jdk/bin/jcontrol
$ORACLE_HOME/jdk/jre/bin/jcontrol
```

On AIX:

```
$ORACLE_HOME/QOpatch/qopiprep.bat
$ORACLE_HOME/jdk/jre/bin/classic/libjvm.a
$ORACLE_HOME/jdk/bin/policytool
```

On HPI:

```
$ORACLE_HOME/jdk/jre/lib/IA64N/server/Xusage.txt
$ORACLE_HOME/jdk/jre/bin/jcontrol
$ORACLE_HOME/QOpatch/qopiprep.bat
```

On Windows, no error messages are expected.

3. Stop, upgrade, and restart all global service manager servers one-at-a-time. In order to ensure zero downtime, at least one global service manager server should always be running. Global service manager servers at an earlier version than the catalog will continue to operate fully until catalog changes are made.
4. Upgrade pool databases in any order, either before or after the global service manager and catalog database upgrades, at the discretion of the pool database administrator.

Creating the Global Data Services Catalog

Every Global Data Services configuration must have a Global Data Services catalog. The Global Data Services catalog can reside on the same host as a GDS configuration database, but Oracle does not recommend this scenario for large configurations. Oracle recommends that you use Oracle high availability features such as Oracle Real Application Clusters (Oracle RAC) and Oracle Data Guard to protect the Global Data Services catalog against outages.

Global Data Services Catalog Requirements

- The Global Data Services catalog must reside on an Oracle Database 12c (or later) database that uses a server parameter file (SPFILE).

If you create the Global Data Services catalog in an Oracle RAC database, then Oracle recommends that you set up Single Client Access Name (SCAN) for that database.

- The Global Data Services administrator who creates the Global Data Services catalog must have a user account on the **catalog database**, and must have `GSMADMIN_ROLE` privileges and an account password. For example, the following SQL statements can be executed on the catalog database.

```
CREATE USER gsm_admin IDENTIFIED BY password;
GRANT gsmadmin_role TO gsm_admin;
```

- The Global Data Services catalog must be protected for high availability and disaster recovery.

Note: Oracle recommends that the Global Data Services administrator does not directly connect to the catalog database, despite having a user account on the catalog database. Global Data Services administrators can use the GDSCTL utility to manage Global Data Services. GDSCTL connects to the Global Data Services catalog with the credentials that the Global Data Services administrator provides when running GDSCTL commands.

Use GDSCTL on any host where GDSCTL is installed and configured to create a Global Data Services catalog, as follows:

```
GDSCTL> create catalog -database db_name -user user_name
```

For example:

```
GDSCTL> create catalog -database serv1:1521:catdb.example.com -user gsm_admin
```

In the preceding example, `serv1:1521:catdb.example.com` is an Easy Connect string that contains the host name and port number of the listener that is used to connect to the database, and `catdb.example.com` is the service name for the Global Data Services catalog database.

You designate one database as the primary repository for the Global Data Services catalog. You can use existing high availability technologies, such as Oracle RAC, Oracle Data Guard, and Oracle Clusterware, to protect the Global Data Services catalog.

See Also: ["create catalog"](#) on page A-30 for complete usage information

Adding a Global Service Manager to the Global Data Services Catalog

Before a global service manager can be started, the global service manager should be registered in the Global Data Services catalog.

To add a global service manager:

1. Alter the `GSMCATUSER` account.

Every global service manager in a Global Data Services configuration maintains a direct Oracle Net Services connection to the catalog database under the `GSMCATUSER` account, which is created by default during Oracle Database 12c installation. The database administrator (DBA) of the catalog database must unlock the account and give the account password to the Global Data Services administrator.

```
ALTER USER gsmcatuser ACCOUNT UNLOCK;
```

```
ALTER USER gsmcatuser IDENTIFIED BY password;
```

2. Run the following command on the host where you want the global service manager to run:

```
GDSCTL> add gsm -gsm gsm_name -listener listener_port -catalog catalogdb_name
```

For example:

```
GDSCTL> add gsm -gsm east_gsm1 -listener 1523 -catalog
serv1:1521:catdb.example.com
```

In the preceding example, `serv1:1521:catdb.example.com` is the connect identifier of the catalog database. The Global Data Services administrator is prompted for the `GSMCATUSER` password during the execution of the command.

See Also: "add gsm" on page A-13 for complete usage information

3. After you add the global service manager to the Global Data Services framework, start the global service manager, as follows:

```
GDSCTL> start gsm -gsm gsm_name
```

During startup, the global service manager creates or modifies the `ONS.CONFIG` file and populates the file with configuration data from the Oracle Notification Service server that belongs to the global service manager.

By default, the file is created in the `$ORACLE_HOME/opmn/conf` directory. The location can be changed to `$ORACLE_CONFIG_HOME/opmn/conf` if the environment variable `ORACLE_CONFIG_HOME` is set.

Note: The `ONS.CONFIG` file cannot be shared, and there must be a unique `ONS.CONFIG` file for each global service manager installation.

Connecting to the Global Data Services Catalog

Many GDSCTL commands require a connection to the Global Data Services catalog. You can connect to the Global Data Services catalog using one of the following two methods:

Method 1

Connect to the Global Data Services catalog, as follows:

```
GDSCTL> connect user_name@connect_identifier
Enter password: ***
Catalog connection is established
GDSCTL>
```

In the preceding example, `catalog` is a connect identifier that resolves to one or more global service manager endpoints. For high availability, Oracle recommends that the connect identifier resolves to the list of all global service managers in the configuration. For example, if there are two global service managers in the Global Data Services configuration and you use `TNSNAMES` for name resolution, then the `tnsnames.ora` file must contain an entry similar to the following:

```
catalog = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=xyz) (PORT=1523))
          (ADDRESS=(PROTOCOL=tcp) (HOST=abc) (PORT=1523))
          )
```

In the preceding example, the first global service manager runs on the host named *xyz*, and the second global service manager runs on the host named *abc*.

See Also: ["connect"](#) on page A-29 for complete usage information

Method 2

To run GDSCTL commands from the operating system prompt, append the `-catalog` parameter to any of the commands that require you to be connected to the Global Data Services catalog.

For example:

```
$ gdsctl add gdspool -gdspool hr -catalog mygdscatlog
```

```
username: Robert
```

```
password:
```

GDSCTL must use a global service manager as a listener to connect to the Global Data Services catalog because the location of the Global Data Services catalog can change. The global service manager records location changes and can route connection requests.

Adding a Global Data Services Pool

Notes:

- If you require only one Global Data Services pool, then you do not need to add one using these instructions. A default Global Data Services pool, `DBPOOLORA`, is created for you when you create the Global Data Services catalog.
 - The Global Data Services administrator has permissions to run GDSCTL commands to manage a Global Data Services pool and, if there is only a single pool, then the Global Data Services administrator also administers the pool.
 - If you specify a user when you run the `gdsctl add gdspool` command, then the local DBA where the Global Data Services catalog resides must first add the user to the catalog database.
-
-

Large database clouds can require multiple Global Data Services pools that are managed by different administrators. Ensure that you are connected to the Global Data Services catalog and add a pool, administered by a specific user, as follows:

```
GDSCTL> add gdspool -dbpool database_pool_list [-users user_list]
```

For example:

```
GDSCTL> add gdspool -dbpool hr -users rjones
```

The preceding example adds a Global Data Services pool called `hr`, and adds the user `rjones`, who is assigned the privileges to administer the `hr` pool. The privileges enable the pool administrator to add databases to the pool and manage global services on the databases in the pool.

A Global Data Services pool must have a unique name within its GDS configuration. If you do not specify a name for the pool when you create it, then the name defaults to

`oradbpool`. The pool name can be up to 30 bytes long and can be any valid identifier (an alphabetical character followed by zero or more alphanumeric ASCII characters or the underscore (_)).

Adding a Global Data Services Region

Note: If you require only one Global Data Services region, then you do not need to add a region using these instructions. A default Global Data Services region, `REGIONORA`, is created for you when you create the Global Data Services catalog.

Ensure that you are connected to the Global Data Services catalog and add a Global Data Services region, as follows:

```
GDSCTL> add region -region region_list
```

For example:

```
GDSCTL> add region -region west,east
```

The preceding example adds two regions, `east` and `west`, to the Global Data Services framework.

A Global Data Services region should have a name that is unique within the corresponding Global Data Services configuration. If no name is specified at the first region creation time, the default name, `oraregion`, is given to the region. The region name can be up to 30 characters long and can be any valid identifier - an alphabetical character followed by zero or more alphanumeric ASCII characters or '_'.

Adding a Database to a Global Data Services Pool

To provide global services, a database must be added to a Global Data Services pool. Before adding a database to a pool, the database administrator should unlock the `GSMUSER` account and give the password to the Global Data Services pool administrator, as shown in the following example:

```
ALTER USER gsmuser ACCOUNT UNLOCK;
ALTER USER gsmuser IDENTIFIED BY password;
```

To be part of a Global Data Services pool, a database must use a server parameter file (SPFILE). An Oracle RAC database should also have `SCAN` set up.

To add a database, the user should connect to the Global Data Services catalog using the Global Data Services pool or Global Data Services administrator credentials, for example:

```
GDSCTL>connect rjones@catalog
Enter password:
```

and run the `gdsctl add database` command:

```
GDSCTL>add database -connect edc007:1521/db14.east.example.com -region east
-gdspool hr
Enter password:
```

In this example `edc007:1521/db14.east.example.com` is the connect identifier of the database and when prompted, enter the `GSMUSER` account password for this database.

If the pool already contains databases and there are global services associated with the pool, then the services are automatically created on the new database.

Valid Node Checking for Registration

The valid node checking for registration (VNCR) feature provides the ability to configure and dynamically update a set of IP addresses, host names, or subnets from which registration requests are allowed by the global service manager. Database instance registration with a global service manager succeeds only when the request originates from a valid node.

By default, the Global Data Services framework automatically adds a VNCR entry for the host on which a remote database is running each time the `gdctl add database` command is run. The automation (called auto-VNCR) requires that the host name entry exists in either the local hosts file or in the name server. If the remote host is identified by a different name on any of the nodes on which the global service manager runs, then the Global Data Services administrator must manually add VNCR entry to the Global Data Services catalog by running the `gdctl add invitednode` command.

See Also: "`add invitednode (add invitedsubnet)`" on page A-15 for complete usage information

Adding a Service to a Global Data Services Pool

The `gdctl add service` command is used to create a service on the Global Data Services pool databases. A simple example of the command is as follows:

```
GDSCTL>add service -gdspool hr -service emp_report1 -preferred_all
```

In this example `emp_report1` is the service name and the `-preferred_all` option means that the service should normally run on all of the databases in the pool.

The service name specified in the 'add service' command can be domain qualified (for example, `sales.example.com`) or not (for example, `sales`). If the specified name is not domain qualified, the service is created with the default domain name "`<GDS_pool_name>.<GDS_configuration_name>`", however the shorter non-domain qualified name can be used with subsequent `GDSCTL` commands to manage the service. If the specified name is domain qualified, the full domain qualified service name must be used in all subsequent `GDSCTL` commands used to manage the service.

A global service name must be unique within a GDS pool and when qualified by domain, must also be unique within a GDS configuration. A global service cannot be created at a database if a local or global service with the same name already exists at that database.

A global service name can contain alphanumeric characters, `'_'` and `'.'`. The first character must be alphanumeric. The maximum length of a global service name is 64 characters. The maximum length of a domain qualified global service name is 250 characters.

An Oracle Net connect descriptor used to connect to a global service must contain a domain qualified service name

See Also: "add service" on page A-16 for complete usage information

Starting a Global Service

The `gdsctl start service` command is used to start an existing service on the Global Data Services pool databases.

```
GDSCTL>start service -service emp_report1 -gdspool hr
```

If the `-role` parameter is specified for the service, the service only starts on the databases in which the role matches the specified value. If the `-lag` parameter is specified for the service, the service only starts on the databases for which replication lag does not exceed the specified value. Unless `-preferred_all` is specified for the service, the service only starts on the databases that are listed as preferred for the service.

Database Client Configuration

Database clients connect to database services using an Oracle Net connect string. The connect string used for a global service differs from the connect string used for a local service in the following ways:

- The service name parameter in the connection data section specifies a global service
- Multiple connection endpoints are specified, and these endpoints are global service managers rather than local, remote, or single client access name (SCAN) listeners
- The database client's region may be specified in the connection data section

Consider the following connect string:

```
(DESCRIPTION=
  (FAILOVER=on)
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(host=sales-east1) (port=1522))
    (ADDRESS=(host=sales-east2) (port=1522))
    (ADDRESS=(host=sales-east3) (port=1522)))
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(host=sales-west1) (port=1522))
    (ADDRESS=(host=sales-west2) (port=1522))
    (ADDRESS=(host=sales-west3) (port=1522)))
  (CONNECT_DATA=
    (SERVICE_NAME=sales)
    (REGION=east)))
```

This connect string contains three global service managers (`sales-east1`, `sales-east2`, and `sales-east3`) in the client's local region (`east`), and three global service managers (`sales-west1`, `sales-west2`, and `sales-west3`) in the client's buddy region (`west`).

Client-side load balancing is enabled across the global service managers within each region by setting the `LOAD_BALANCE` parameter to `ON` in the address list for each region. Connect-time failover between regions is enabled by setting the `FAILOVER` parameter to `ON`.

It is a best practice to have three global service managers in each region, for each region to have a buddy region, and for client-side load balancing and connect-time failover to be configured as shown in the example connect string.

The `REGION` parameter is optional if only global service managers from the local region are specified in the connect string. This is the case when there is only one region in the GDS configuration, or could be the case when there are multiple regions, but it is not feasible to change the connect string of an existing client designed to work with a single database. If the `REGION` parameter is not specified, the client's region is assumed to be the region of the global service manager used to connect to the global service.

Note: The pre-12c Thin JDBC client can only be used with a GDS configuration that has a single region, unless the region parameter is specified in the connect string.

The Oracle Database 12c and later integrated clients use Oracle Notification Services (ONS) to receive the Fast Application Notification (FAN) events that support load balancing and Fast Connection Failover (FCF). The integrated clients automatically subscribe, without user-configuration, to up to 3 of the ONS servers co-located with the global service managers in each of the client's local and buddy regions.

Note: Automatic ONS configuration is not supported if connections to an ONS server have to be secured using SSL. You must configure ONS manually to enable SSL. See client-specific guides for information on how to configure ONS manually.

Note for Pre-12c Clients:

The pre-12c OCI and ODP.NET clients do not support global services.

The pre-12c JDBC client supports global services, but you must manually configure it to subscribe to the ONS servers co-located with the global service managers in the client's local and buddy regions. It is a best practice to subscribe to three ONS servers in each of the client's local and buddy regions.

See the *Oracle Database JDBC Developer's Guide* for information about how to configure ONS subscriptions.

Configuring Integrated Clients for FAN and FCF

Load balancing and Fast Connection Failover (FCF) of client connections across the databases in a GDS configuration is supported by the Oracle database integrated clients, and requires that those clients be configured for FAN and FCF.

See one of these client-specific Programmer's Guides for information about that client's FAN and FCF configuration requirements:

- *Oracle Database JDBC Developer's Guide*
- *Oracle Universal Connection Pool for JDBC Developer's Guide*
- *Oracle Database Development Guide*
- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows*

Exporting the GDS Catalog Data for Logical Backups

Because the GDS catalog stores metadata for the entire GDS configuration, loss or corruption of the catalog data may require significant efforts to manually recreate it. While unavailability of the GDS catalog does not impact core GDS functionality such as load balancing, service failover, and application notification, no changes can be made to the GDS configuration until the catalog is restored. Therefore it is important to develop a strategy for protecting the GDS catalog.

Even when the GDS catalog is protected by HA technologies such as Oracle RAC and Data Guard, it is highly recommended that you regularly back up the GDS catalog. You can create a logical backup of the catalog by exporting the catalog data to a file. This backup of the GDS catalog can help you in a disaster recovery scenario, and when there is need to undo changes made to the catalog since the last backup was made. You can also use the backup when moving the GDS catalog to a new database.

To export the GDS catalog data to a file, ensure that you are connected to the catalog and execute the following command:

```
GDSCTL> export catalog file_name_with_full_path
```

The catalog configuration will be saved to the specified file on the system where GDSCTL is running. Access to the file should be limited to Global Data Services administrators since it may contain sensitive information such as connection strings for the pool databases.

Notes:

- It is strongly recommended that the catalog be validated before exporting it to ensure that there are no inconsistencies in the catalog data. Any errors reported by the `validate catalog` command should be corrected before exporting the catalog data.
 - You must not make any change to the file with exported catalog data. Any changes to the file may prevent using of this file for the catalog restore, or may cause catalog corruption after restore. It is recommended to store the file checksum along with the backup file. Do not try to restore the catalog configuration if the file has been modified.
-
-

Restoring Logical Backup of the GDS Catalog into the Same Catalog Database

To restore GDS catalog data from a backup file, connect to the catalog database and issue the following command:

```
GDSCTL> import catalog file_name_with_full_path
```

After the import of the catalog data is finished, pool databases will be automatically synchronized (see the `sync database` command description in ["sync database"](#) on page A-70.). If there are no global service managers available, this action will be deferred until a global service manager registers with the catalog.

It is recommended that you validate the catalog after the import is done and all the databases are synchronized.

Note: Trying to restore the GDS catalog from the file that has been modified may result in a corrupt catalog. It is the responsibility of the GDS administrator to check consistency of the backup file (for example, by using the checksum.)

Restoring Logical Backup of the GDS Catalog into a new Catalog Database

When moving a catalog to a new database, you must first create an empty catalog on the database (see "[Creating the Global Data Services Catalog](#)" on page 2-5.). After that the `import catalog` command may be executed as described in the previous section.

If the new catalog database has a different connection string, it is the administrator's responsibility to change the connection string on global service manager systems. It is also required to restart all global service managers in this case. The synchronization procedure will not be completed, and thus the restore procedure will not be finished, until at least one global service manager registers with the catalog.

Changing the GSMCATUSER Password

If you want to change GSMCATUSER password, first run the following command in SQL*Plus while connected to the GDS catalog:

```
ALTER USER gsmcatuser IDENTIFIED BY password
```

Then in GDSCCTL run the following command:

```
GDSCCTL> modify catalog -oldpwd oldpassword -newpwd newpassword
```

Administering Global Data Services Configurations

The `GDSCTL` utility is used to create, manage and monitor a Global Data Services configuration and all of its components. This utility is very similar to the `SRVCTL` utility used to manage an Oracle Real Application Cluster (Oracle RAC).

This chapter includes the following topics:

- [Overview of Global Data Services Administration](#)
- [Managing Database Pools](#)
- [Managing Global Services](#)

Overview of Global Data Services Administration

Global Data Services is managed by the Global Data Services administrator whose responsibilities include the following tasks:

- Installing and upgrading the global service manager software
- Creation and maintenance of the Global Data Services catalog
- Starting, stopping, and configuring global service managers
- Creation and administration of Global Data Services regions and pools
- Management of global services
- Monitoring of the Global Data Services framework components

Each Global Data Services configuration requires at least one Global Data Services administrator. A small configuration can be administered by a single person who performs all the administrative duties. For a large configuration with many regions and pools it may be necessary to have a group of Global Data Services administrators who share responsibilities. All Global Data Services administrators have privileges to perform all the listed administrative tasks for a given Global Data Services configuration.

An operating system account should exist for the Global Data Services administrator on all computers where global service managers are expected to run. The account user should have privileges to install and run global service manager software. Only Global Data Services administrators should be granted these privileges.

A Global Data Services administrator must also be added as a user to the Global Data Services catalog database and granted the `GSMADMIN_ROLE` role. The database account for a Global Data Services administrator should be created by a database administrator of the catalog database. The Global Data Services administrator might create this

account by himself if he happens to have local database administrator privileges on this database.

If a Global Data Services configuration contains multiple pools, then in addition to Global Data Services administrators who manage the entire configuration, each pool can have one or more Global Data Services pool administrators. Responsibilities of a pool administrator are limited to the administration of a particular pool and include the following tasks:

- Adding and removing databases in the pool
- Management of global services in the pool

To perform these tasks a Global Data Services pool administrator must be a user of the Global Data Services catalog database with the appropriate privileges. Creation of the database user for a pool administrator and granting of the privileges is performed automatically when a Global Data Services pool is created with the `-USER` option. A pool administrator can also be added to a pool after its creation using the `gdsctl modify gds_pool` command. A Global Data Services administrator always has privileges to administer any pool in the database configuration.

All administrative operations should be performed using the appropriate GDSCTL commands. Execution of the most GDSCTL commands requires access to the Global Data Services catalog. For such commands, credentials for the catalog database must be specified using the appropriate command options.

Many administrative operations, such as adding a database to a Global Data Services pool, or enabling a global service, require making changes not only to the Global Data Services catalog, but also to databases in the Global Data Services configuration. The generic workflow for such commands is as follows:

- GDSCTL connects to the catalog database with credentials provided by the administrator and makes appropriate changes to the catalog.
- The catalog database notifies all global service managers in the Global Data Services configuration about the changes. The notification is sent using an Oracle Net Services connection that each global service manager maintains with the catalog database.
- After receiving the notification one of the global service managers connects to the configuration databases that need to be configured and makes the appropriate changes.

To support this workflow a global service manager should be able to connect to the catalog and configuration databases. The connection to the catalog database is established using `GSMCATUSER` account, which is created by default on any Oracle 12c database during database installation. The account must be unlocked by the database administrator of the catalog database and its password given to the Global Data Services administrator. Whenever a new global service manager is added to the GDS configuration, the Global Data Services administrator has to specify the password for the `GSMCATUSER` account. The password is then encrypted and stored in the global service manager wallet for future use by the global service manager.

The global service manager connects to the pool databases using the `GSMUSER` account, which also exists by default on any Oracle 12c database. The account is locked after the database installation. It should be unlocked by the local database administrator before the database can be added to a Global Data Services pool. The password for the `GSMUSER` account is given to the pool or Global Data Services administrator who adds the database to a Global Data Services pool and must be specified in the `gdsctl add database` command. The password is stored in the Global Data Services catalog for future use by all global service managers.

The GSMUSR passwords are stored in the GDS catalog in an encrypted form using the PKCS 1 encryption/decryption schema. You can encrypt GSMUSR passwords stored in the GDS catalog with a newly generated key by executing the `modify catalog` command. For example:

```
GDSCTL> modify catalog -newkeys
```

GSMCATUSER and GSMUSER accounts are shared by all global service managers in the Global Data Services framework and used for all management operations performed by global service managers, including automatic operations such as service failover. Human users should never connect to databases using these accounts.

Managing Database Pools

This section describes the administration tasks associated with managing database pools in the global data services framework. It contains the following topics:

- [Adding Oracle Data Guard Broker Managed Databases to a Database Pool](#)

Adding Oracle Data Guard Broker Managed Databases to a Database Pool

When you include an Oracle Data Guard broker configuration in a Global Data Services configuration, you manage the broker configuration as one unit. Only an entire Oracle Data Guard broker configuration can be added to (or deleted from) a database pool. A configuration cannot span multiple pools. An attempt to add or remove an individual database to or from a pool that belongs to a broker configuration results in an error.

The only way to add a database to the pool is to add the database to the broker configuration (using the `DGMGRL` utility). Adding a database to the broker configuration causes its automatic addition to the database pool to which this configuration belongs. Removing a database from a broker configuration causes its removal from the pool that contains the configuration. This is the only way to remove a database from a pool that contains a broker configuration.

Also, note the following limitations:

- The set of databases in a database pool can be either:
 - The set of databases that belong to a single broker configuration
 - A set of databases that belong to no broker configuration

You can add a broker configuration only to an empty database pool and, if a pool already contains a broker configuration, then, to add a database to a database pool, you must add the database to the broker configuration contained in the database pool.

- Role-based global services are supported only for database pools that contain a broker configuration.

See Also: *Oracle Data Guard Broker* for more information about the `DGMGRL` utility

Managing Global Services

This section describes the administration tasks associated with global services. It contains the following topics:

- [Creating a Global Service](#)

- [Starting a Global Service](#)
- [Stopping a Global Service](#)
- [Disabling a Global Service](#)
- [Enabling a Global Service](#)
- [Modifying Global Service Attributes](#)
- [Deleting a Global Service](#)

Creating a Global Service

A global service is created by execution of the `ADD SERVICE` command. This command associates the global service with a Global Data Services pool and stores attributes of the service in the Global Data Services catalog. If databases are specified using the `PREFERRED` or `AVAILABLE` options, the service is created on those specified databases. If the `PREFERRED_ALL` option is used, the service is created on all databases in the Global Data Services pool.

A service that already exists in a Global Data Services pool is also automatically created on a database in the following cases:

- The service is modified to add a database that is part of the pool.
- The service has the `PREFERRED_ALL` attribute and a new database is added to the pool.

See Also: ["add service"](#) on page A-16

Starting a Global Service

A global service is automatically enabled immediately after it has been created. The term *enabled* means that the service can be started on a database if the database is eligible for running the service, namely, when the following conditions are met:

- The database is open and registered with a global service manager.
- The service has not been disabled on that database.
- The database role matches the role attribute of the service.
- The replication lag on the database does not exceed the maximum value specified for the service.
- The service has not reached its cardinality defined by the number of preferred databases.
- No other database in the pool is a better candidate for starting the service, for example, the service can be started on an available database only if there is no eligible preferred database.

A newly created global service never gets started automatically until the `START SERVICE` command is executed by the user. This gives the pool administrator control over the initial service startup which may be important in the case when multiple services are being added to the pool and a certain sequence of service startups is required.

A service with the automatic management policy (the default option) must be initially started by executing the `START SERVICE` command without the `-database` option. This command not only starts the service on all eligible databases in the pool, but also enables the automatic service startup in the following cases:

- After the service is automatically created on a database that is eligible to run it. (The two cases of automatic service creation are listed in the previous section.)
- A database that was down gets restarted and is eligible for the service.
- A database becomes eligible to run the service. This can happen, for example, because the replication lag on a database has decreased to an acceptable value, or the service cardinality has been increased by the user.

The `START SERVICE` command with `-database` option can be used to start a service with the automatic management policy on particular databases if the service was shut down there by the `STOP SERVICE` command described in "[Stopping a Global Service](#)" on page 3-5.

A service with the manual policy must be started manually on each individual database, including when a database gets restarted or becomes eligible to run the service. When executed against a service with the manual policy, the `START SERVICE` command without the `-database` option starts the service on all eligible databases that are currently present in the pool. If used with the `-database` option, the command starts the service only on the specified databases, if they are eligible to run it.

Note: The cases of automatic service startup listed in this section only describe what happens when the `START SERVICE` command is executed against a service with the automatic management policy. They do not include cases when a service is started automatically on a database because of a failover from another database. Service failover is not associated with the `START SERVICE` command, and its behavior is the same for services with automatic and manual management policy.

See Also: "[start service](#)" on page A-62

Stopping a Global Service

A global service running on databases in a Global Data Services pool can be shut down by the `STOP SERVICE` command. If the `STOP SERVICE` command is executed with the `-database` option, then the service is stopped on the specified databases; otherwise it is stopped on all databases in the pool.

Note: A stopped service with the automatic management policy is restarted if the database where it was running gets restarted and is eligible to run the service. Also, stopping a service with the automatic management policy on all databases in a Global Data Services pool does not prevent the automatic service startup on a new database when the service is created there. To completely disable the automatic startup of a service, its management policy should be changed to manual.

When the service is stopped by the user, the Global Data Services framework considers that database to be temporarily unavailable for this service. Stopping a global service does not cause a service failover event; the service cardinality is temporarily decreased and the global service manager does not attempt to start the service on another database in the pool.

However, a database with a stopped service is still considered a failover target for this service; when the service fails on another database, it can be started on this database if it is eligible to run the service. After the service failover to a database, the service on that database is no longer considered to be stopped by the user.

A stopped service can be manually restarted by executing the `START SERVICE` command.

See Also: ["stop service"](#) on page A-67

Disabling a Global Service

A global service can be disabled on a database or a set of databases by executing the `DISABLE SERVICE` command. A disabled service cannot be started until it is reenabled. This includes the service failover from another database; a database with the disabled service is never considered a failover target.

A service has to be stopped to be disabled. An error is returned if `DISABLE SERVICE` is executed against a database where the service is running.

See Also: ["enable service"](#) on page A-34

Enabling a Global Service

A disabled global service can be reenabled on a database by executing `ENABLE SERVICE` command. If the service management policy is `AUTOMATIC` and the database is eligible for running the service, it is started automatically after being enabled. A service with the `MANUAL` management policy must be started manually. A database can become a failover target after a service is enabled there.

See Also: ["add service"](#) on page A-16

Modifying Global Service Attributes

The `MODIFY SERVICE` command is used to modify global service attributes. In addition to specifying service properties (such as role, maximum lag, load balancing method, and so on) service attributes define on which databases the service should run. Therefore `MODIFY SERVICE` can be used to add a database to a service, remove it from a service, or move a service from one database to another. As the result of the command execution, a service may be created, deleted, started, or stopped on one or more databases in a Global Data Services pool.

Most global service attributes are specified at the service creation time in the `ADD SERVICE` command and only need to be modified when some changes have to be made. However, a few service attributes related to Oracle RAC databases, must be set by executing the `MODIFY SERVICE` command right after the `ADD SERVICE` command has been executed. These attributes include the name of the server pool, instance cardinality (`UNIFORM/`) and some other parameters that are specific to particular Oracle RAC databases. Such attributes cannot be set by the `ADD SERVICE` command because this command is only used to specify attributes that have the same values for all databases in a Global Data Services pool.

See Also: ["modify service"](#) on page A-41

Deleting a Global Service

The `REMOVE SERVICE` command deletes a global service from the Global Data Services pool by removing it from the Global Data Services catalog and all databases where it was created. A service should be stopped before being deleted.

See Also: ["remove service"](#) on page A-54

Troubleshooting

This chapter contains topics to help you troubleshoot issues you encounter with Global Data Services.

This chapter include the following topics:

- [Troubleshooting Oracle Error Codes](#)
- [Solutions for General Issues](#)
- [Obtaining the Running Status of Global Data Services Components](#)
- [Viewing Static Configuration Information for Global Data Services Components](#)
- [Enabling and Disabling Tracing on a Global Service Manager](#)
- [Using Global Service Manager Log and Trace Files](#)
- [Using SYS_CONTEXT Parameters in a GDS Environment](#)

Troubleshooting Oracle Error Codes

This section contains information for troubleshooting specific Oracle error messages you might encounter, such as:

- [ORA-01045: user GSMADMIN_INTERNAL lacks CREATE SESSION privilege; logon denied](#)
- [ORA-12514: TNS:listener does not currently know of service requested in connect descriptor](#)
- [ORA-12516: TNS:listener could not find available handler with matching protocol stack](#)
- [ORA-12541: TNS:no listener](#)
- [GSM-45034: Connection to GDS catalog is not established](#)
- [GSM-45054: GSM error or NET-40006: unable to start GSM](#)

ORA-01045: user GSMADMIN_INTERNAL lacks CREATE SESSION privilege; logon denied

The user `GSMADMIN_INTERNAL` is an internal only user, it should never be unlocked or used for any database login. No direct modifications should be made on the Global Data Services schema objects unless directed by Oracle Technical Support.

ORA-12514: TNS:listener does not currently know of service requested in connect descriptor

The global service may be down. Verify that the pool databases are up and the service is started.

The global service may be disabled. Ensure that the pool databases are up and the service is enabled and started.

The GDS pool database may be down. Ensure that the GDS pool databases are up and the service is enabled and started.

ORA-12516: TNS:listener could not find available handler with matching protocol stack

The GDS pool database's local listener may be down. Ensure that the GDS pool database local listener is running.

ORA-12541: TNS:no listener

All global service managers may be down. Verify that the global service managers are running.

GSM-45034: Connection to GDS catalog is not established

The GDS catalog database or its listener may be down. Verify that the GDS catalog database and its local listener are running.

GSM-45054: GSM error or NET-40006: unable to start GSM

The GDS catalog database or its listener may be down. Verify that the GDS catalog database and its local listener are running.

Solutions for General Issues

This section contains information for solving some general issues you might encounter, such as:

- [Connecting to GDS Configuration Databases When No Global Service Managers Are Running](#)
- [Connecting to Catalog Databases When No Global Service Managers Are Running](#)

Connecting to GDS Configuration Databases When No Global Service Managers Are Running

You need multiple address lists; the first list should be exclusively regional global service manager listeners, the second list contains global service manager listeners of the buddy region and the third list contains local listeners.

You can always connect through a global service manager while it is up, and only fail over to local listeners when all global service manager listeners are down.

Template:

```
(DESCRIPTION=
  (FAILOVER=on)
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(global_protocol_address_information))
```

```

        (ADDRESS=(global_protocol_address_information))
        (ADDRESS=(global_protocol_address_information))
    )
    (ADDRESS_LIST=
        (LOAD_BALANCE=ON)
        (ADDRESS=(global_protocol_address_information))
        (ADDRESS=(global_protocol_address_information))
        (ADDRESS=(global_protocol_address_information))
    )
    (ADDRESS_LIST=
        (LOAD_BALANCE=ON)
        (ADDRESS=(local_protocol_address_information))
        (ADDRESS=(local_protocol_address_information))
    )
    (CONNECT_DATA=
        (SERVICE_NAME=global_service_name)
        (REGION=region_name))

```

Example:

```

(DESCRIPTION=
    (FAILOVER=on)
    (ADDRESS_LIST=
        (LOAD_BALANCE=ON)
        (ADDRESS=(HOST=gsmhost1) (PORT=1523) (PROTOCOL=TCP))
        (ADDRESS=(HOST=gsmhost2) (PORT=1523) (PROTOCOL=TCP))
        (ADDRESS=(HOST=gsmhost3) (PORT=1523) (PROTOCOL=TCP))
    )
    (ADDRESS_LIST=
        (LOAD_BALANCE=ON)
        (ADDRESS=(HOST=gsmhost4) (PORT=1523) (PROTOCOL=TCP))
        (ADDRESS=(HOST=gsmhost5) (PORT=1523) (PROTOCOL=TCP))
        (ADDRESS=(HOST=gsmhost6) (PORT=1523) (PROTOCOL=TCP))
    )
    (ADDRESS_LIST=
        (LOAD_BALANCE=ON)
        (ADDRESS=(HOST=server1) (PORT=1521) (PROTOCOL=TCP))
    )
    (CONNECT_DATA=
        (SERVICE_NAME=sales_read_service.dbpoolora.oradbccloud)
        (REGION=WEST))

```

Note: In the case of an Oracle RAC enabled GDS database, the third address list contains the local Oracle RAC database's SCAN listeners.

Connecting to Catalog Databases When No Global Service Managers Are Running

Local listener enables access to the GDS catalog database even when global service managers are down.

This access may be needed for any DB Administration/maintenance activities on the catalog database when global service managers are not running.

Obtaining the Running Status of Global Data Services Components

The status command can be used to obtain the running status of the GDS components.

```
GDSCTL>status gsm

GDSCTL>status service

GDSCTL>status database
```

Viewing Static Configuration Information for Global Data Services Components

The `gdsctl config` command can be used to obtain the static configuration information of various GDS components.

```
GDSCTL>config

GDSCTL>config gsm

GDSCTL>config region

GDSCTL>config gdspool

GDSCTL>config database

GDSCTL>config service

GDSCTL>config invitednode
```

Enabling and Disabling Tracing on a Global Service Manager

You can enable tracing using the `set trace_level` command.

```
GDSCTL>set trace_level -gsm gsm_name SUPPORT
```

The `SUPPORT` option provides trace with troubleshooting information for Oracle Support Services. The other options are `ADMIN` and `USER`.

To disable tracing:

```
GDSCTL>set trace_level -gsm gsm_name OFF
```

Using Global Service Manager Log and Trace Files

The exact location of a given global service manager's log and trace files can be obtained using the `status gsm` command as shown in the following example.

```
GDSCTL>status gsm

Alias                MYGSM
Version              12.1.0.0.2
Start Date           13-OCT-2012 12:20:16
Trace Level          support
Listener Log File    /scratch/oracle/diag/gsm/myhost/mygsm/alert/log.xml
Listener Trace File  /scratch/oracle/diag/gsm/myhost/mygsm/trace/ora_1829_
47542149303936.trc
Endpoint summary     (ADDRESS=(HOST=myhost.com)(PORT=1571)(PROTOCOL=tcp))
GSMOCI Version       0.1.7
Mastership           N
Connected to GDS catalog Y
Process Id           1833
```

```

Number of reconnections 0
Pending tasks. Total 0
Tasks in process. Total 0
Regional Mastership TRUE
Total messages published 34261
Time Zone -07:00
Orphaned Buddy Regions: None
GDS region east
Network metrics:
  Region: euro RTT:34 Bandwidth:40

```

In this example `myhost` is the global service manager host name and `mygsm` is the name of the global service manager.

The text based listener log can be found in `/scratch/oracle/diag/gsm/hostname/gsm_name/trace` directory. The file is called `alert_gsm*.log` (for example, `alert_gsm1.log`)

Using SYS_CONTEXT Parameters in a GDS Environment

For a session established using a connection to a global service, some parameters of namespace `USERENV` have values that are different from values set when connecting to a local service on the same database. The different values for a global service are set to make the database pool appear to clients as a single database with many instances. This was done to provide backward compatibility with pre-12c clients which expect multiple instances of a service to exist only on an Oracle RAC database.

When a client connects to a global service, GDS sets the following in the session context differently.

- `DB_UNIQUE_NAME` and `DB_DOMAIN` are set to `<gdspool_name>.<config_name>`
- `INSTANCE` is set to a system generated number `<inst_num>` which is unique within a GDS configuration
- `INSTANCE_NAME` is set to `<gdspool_name>%<virtual_instance_num>`
- `SERVICE_NAME` is set to `<region_name>%<service_name>`

A

Global Data Services Control Utility (GDSCTL) Reference

This appendix includes a complete reference for the Global Data Services utility (GDSCTL) for Oracle Database.

This appendix includes the following topics:

- [About GDSCTL](#)
 - [GDSCTL Overview](#)
 - [Operational Notes](#)
 - [Special Topic Discussion](#)
- [GDSCTL Reference](#)

About GDSCTL

This section contains topics that about using the GDSCTL utility.

- [GDSCTL Overview](#)
- [Operational Notes](#)
- [Special Topic Discussion](#)

GDSCTL Overview

The GDSCTL utility is a command-line interface for configuring and managing the Global Data Services framework. To run some commands, GDSCTL must establish a connection to a global service manager, a Global Data Services catalog database, or a database in the Global Data Services configuration.

Operational Notes

Note: Unless specified, GDSCTL resolves connect strings with the current name resolution methods (such as TNSNAMES). The exception is the global service manager name. GDSCTL queries the `gsm.ora` file to resolve the global service manager name.

To start GDSCTL, enter the following command at the operating system prompt:

```
$ gdsctl
```

The preceding command starts GDSCTL and displays the GDSCTL command prompt. You can enter GDSCTL commands at either the operating system prompt or the GDSCTL command prompt, as shown in the following examples:

```
$ gdsctl add gsm -gsm gsm1 -catalog 127.0.0.1:1521:db1
```

```
GDSCTL> add gsm -gsm gsm1 -catalog 127.0.0.1:1521:db1
```

Both of the preceding commands achieve the same result. The first command is run at the operating system command prompt while the second command is run at the GDSCTL command prompt. The command syntax examples in this document use the GDSCTL command prompt.

Notes:

- Many GDSCTL commands require you to first connect to the Global Data Services catalog before running the command.

If you run commands from the GDSCTL prompt, then you must execute the `connect` command before the first GDSCTL command that requires connection to the Global Data Services catalog. The `connect` command needs only to be run once in a GDSCTL session.

- A net service name may be specified instead of a connect descriptor when adding a database or broker configuration to a GDS configuration. If a net service name is specified, it must be resolvable at each global service manager in the GDS configuration to a connect descriptor that allows connectivity to the entity that is being added.
-
-

Alternatively, you can gather all the GDSCTL commands in one file and run them as a batch with GDSCTL, as follows:

```
$ gdsctl @script_file_name
```

The preceding command starts GDSCTL and runs the commands contained in the specified script file.

Using GDSCTL Help

You can display help for GDSCTL, as follows:

- GDSCTL> `help`: The `help` command displays a summary of all GDSCTL commands. If you specify a command name after `help`, then the help text for that command displays.
- GDSCTL> `command -h`: This syntax displays help text for the specified command, where `command` is the command name.

The following examples display identical help text for the `start` command:

```
GDSCTL> help start
GDSCTL> start -h
```

Privileges and Security

Only users with the proper privileges can run GDSCTL commands.

See Also: ["Overview of Global Data Services Administration"](#) on page 3-1 for more information about GDSCTL privileges and security

Special Topic Discussion

GDSCTL Connections

For certain operations GDSCTL must connect to a global service manager. To connect to a global service manager, GDSCTL must be running on the same host as the global service manager. When connecting to a global service manager, GDSCTL looks for the `gsm.ora` file associated with the local global service manager. [Table A-1](#) lists the GDSCTL operations that require a connection to a global service manager.

Table A-1 *Operations That Require a Connection to a Global Service Manager*

GSM Operation	Description
<code>add gsm</code>	Add a global service manager.
<code>start gsm</code>	Start the global service manager.
<code>stop gsm</code>	Stop the global service manager.
<code>modify gsm</code>	Modify the configuration parameters of the global service manager.
<code>status gsm</code>	Obtain the status of a global service manager.
<code>set inbound_connect_level</code>	Set the <code>INBOUND_CONNECT_LEVEL</code> listener parameter.
<code>set trace_level</code>	Set the trace level for the listener associated with the specified global service manager.
<code>set outbound_connect_level</code>	Set the timeout value for the outbound connections for the listener associated with a specific global service manager.
<code>set log_level</code>	Set the log level for the listener associated with a specific global service manager.

For all other operations, GDSCTL uses Oracle Net Services to connect to the Global Data Services catalog database or another database in the Global Data Services configuration. For these connections you can run GDSCTL from any client or host that has the necessary network configuration.

GDSCTL Reference

GDSCTL Command Syntax and Options

GDSCTL commands, objects, and options; database names, instance names, Global Data Services region names, Global Data Services pool names, and service names are all case insensitive. Passwords and server pool names are also case sensitive. GDSCTL uses the following command syntax:

```
$ gdsctl command [object] [options] [argument]
or
GDSCTL> command [object] [options] [argument]
```

In GDSCTL syntax:

- *command*: A verb such as add, start, stop, or remove
- *object* (also known as a noun): The target or object on which GDSCTL performs the command, such as service or database. You can find a list of objects in [Table A-3](#).
- *options*: Optional flags that extend the use of a preceding command combination to include additional parameters for the command. For example, the `-gdspool` option indicates that the name of a specific Global Data Services pool follows. If a comma-delimited list follows an option, then do not use spaces between the items in the list.
- *argument*: Additional variables for the GDSCTL command to specify actions for an object, or to specify actions for GDSCTL without an object.

Table A-2 Summary of GDSCTL Commands

Command	Description
add on page A-10	Add a global service manager, region, database, database pool, invitednode, Oracle Data Guard broker configuration, or service to the Global Data Service management framework.
config on page A-22	Display the static configuration for databases, database pools, global service managers, services, and valid node checking for registration (VNCR).
configure on page A-28	Set the GDSCTL parameters.
connect on page A-29	Specify the credentials to use for the current GDSCTL session.
create catalog on page A-30	Create a Global Data Services catalog in a specified database.
delete catalog on page A-32	Delete a Global Data Services catalog.
disable service on page A-33	Disable a global service.
enable service on page A-34	Enable a global service.
modify on page A-37	Modify catalog, databases, database pools, regions, or services.
relocate service on page A-49	Move a global service from one database to another.

Table A–2 (Cont.) Summary of GDSCTL Commands

Command	Description
remove on page A-51	Remove global service managers, regions, databases, database pools, invitednode, services, or Oracle Data Guard broker configurations.
services on page A-56	Display information about the services registered with a specific global service manager.
set on page A-58	Set the current global service manager for the current session and sets various run-time parameters of the local global service manager.
start on page A-62	Start a global service manager or global services.
status on page A-64	Show status of global service managers, regions, databases, database pools, or services.
stop on page A-67	Stop a global service manager or global services.
sync brokerconfig on page A-69	Synchronize the Oracle Data Guard broker configuration in the global service manager with the configuration in the database pool.
sync database on page on page A-70	Bring attributes of global services and GDS related parameters of a pool database in synchronization with the contents of the GDS catalog.
validate catalog on page A-71	Cross check the Global Data Services catalog, global service manager run-time status, and pool databases, and reports any inconsistencies and errors.

GDSCTL Objects Summary

[Table A–3](#) lists the keywords that you can use for the *object* portion of GDSCTL commands. You can use either the full name or the abbreviation for each object keyword. The **Purpose** column describes the object and the actions that can be performed on that object.

Table A–3 Object Keywords and Abbreviations for GDSCTL

Object	Keyword (Abbreviations)	Purpose
Global Data Services catalog	autovncr	Enables or disables valid node checking for registration (VNCR) list for database registration
Oracle Data Guard broker configuration	brokerconfig	To add, modify, and manage the Oracle Data Guard broker configuration. The Oracle Data Guard broker logically groups primary and standby databases into a broker configuration that enables the broker to manage and monitor them together as an integrated unit.
Global Data Services catalog	catalog	To manage the Global Data Services catalog stored in an Oracle database.
Database	database	To add, modify, and remove database configuration information about databases.

Table A-3 (Cont.) Object Keywords and Abbreviations for GDSCTL

Object	Keyword (Abbreviations)	Purpose
Global Data Services pool	gdspool	To add, modify, and manage a Global Data Services pool. A Global Data Services pool is a set of databases within a GDS configuration that provides a unique set of global services and belongs to a certain administrative domain.
Global service manager	gsm	To add, modify, and manage a global service manager. A global service manager is a software component that provides service-level load balancing and centralized management of services within the Global Data Services configuration.
Global Data Services catalog	invitednode	Adds host address information to the valid node checking for registration (VNCR) list in the Global Data Services catalog.
Global Data Services catalog	invitedsubnet	Adds subnet information to the valid node checking for registration (VNCR) list in the Global Data Services catalog.
Global Data Services Region	region	To add, modify, and manage a Global Data Services Region, which is a logical boundary that contains database clients and servers that are considered to be geographically close to each other.
Service	service	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, and remove global services.

add

The `gdsctl add` command adds elements to the global service manager configuration. [Table A-4](#) describes the `add` command variations.

Table A-4 *gdsctl add command Summary*

Command	Description
add brokerconfig on page A-10	Adds an Oracle Data Guard configuration to a Global Data Services pool
add database on page A-11	Adds a database to a Global Data Services region or Global Data Services pool
add gdspool on page A-12	Adds a Global Data Services pool to the configuration and optionally configures pool administrator users
add gsm on page A-13	Adds a global service manager to the configuration
add invitednode (add invitedsubnet) on page A-15	Adds a host or IP address to a VNCR list in the Global Data Services catalog
add region on page A-16	Adds a Global Data Services region to the configuration
add service on page A-16	Adds a global service to the configuration

add brokerconfig

Adds an Oracle Data Guard broker configuration to a Global Data Services pool.

Syntax and Options

Use the `gdsctl add brokerconfig` command with the following syntax:

```
add brokerconfig [-gdspool gdspool_name] -connect connect_identifier
                [-region region_name] [-pwd password] [-savename]
```

Table A-5 *gdsctl add brokerconfig Options*

Option	Description
<code>-connect <i>connect_identifier</i></code>	Specify an Oracle Net connect descriptor or net service name that resolves to a connect descriptor for a database in the broker configuration.
<code>-gdspool <i>gdspool_name</i></code>	The pool to which the databases of the Oracle Data Guard broker configuration are to be added. If the specified Global Data Services pool already contains databases or another configuration, GDSCTL returns an error.
<code>-pwd <i>password</i></code>	The password for the GSMUSER. If <code>-pwd</code> is not specified, then you are prompted for the password.
<code>-region <i>region_name</i></code>	The Global Data Services region to which the databases belong. If you specify a region, then all the databases are added to that region. If you do not specify a region, then all databases are added with a region of UNASSIGNED. If the region is UNASSIGNED, then you must use the modify database command to change the region.
<code>-savename</code>	Specify this option to store a net service name specified with the <code>-connect</code> option in the Global Data Services catalog, rather than the connect descriptor mapped to that net service name.

Usage Notes

- You must connect to the Global Data Services catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Examples

Add the Oracle Data Guard broker configuration for the DB1 database to the Global Data Services pool MYREADERFARM and the WEST region.

```
GDSCCTL> add brokerconfig -connect 192.168.1.1:1521:sid -region west -gdspool
myreaderfarm
```

Exceptions or Error Codes

GDSCCTL returns the errors listed in [Table A-6](#) if you use this command incorrectly.

Table A-6 *gdsctl add brokerconfig Exceptions or Error Codes*

Exception	Description
ERROR-44866	A pool can only contain one Data Guard broker configuration. If a Global Data Services pool already contains an Oracle Data Guard broker configuration, then GDSCCTL returns error 44866 because a database must be added using Oracle Data Guard in this case.

add database

Adds databases to a Global Data Services region and Global Data Services pool.

Syntax and Options

Use the `gdsctl add database` command with the following syntax:

```
add database -connect connect_identifier [-region region_name]
[-gdspool gdspool_name] [-pwd password] [-savename] [-cpu_threshold cpu]
[-disk_threshold disk]
```

Table A-7 *gdsctl add database Options*

Option	Description
<code>-connect <i>connect_identifier</i></code>	Specify an Oracle Net connect descriptor or net service name that resolves to a connect descriptor for the database being added.
<code>-cpu_threshold <i>cpu</i></code>	Specifies CPU Utilization percentage threshold.
<code>-disk_threshold <i>disk</i></code>	Specifies the average latency in milliseconds of a synchronous single-block read.
<code>-gdspool <i>gdspool_name</i></code>	The Global Data Services pool to which the database belongs.
<code>-pwd <i>password</i></code>	The password for the GSMUSER. If <code>-pwd</code> is not specified, then you are prompted for the password.
<code>-region <i>region_name</i></code>	The Global Data Services region to which the database belongs.
<code>-savename</code>	Specify this option to store a net service name specified with the <code>-connect</code> option in the Global Data Services catalog, rather than the connect descriptor mapped to that net service name.

Usage Notes

- You must connect to the Global Data Services catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.
- If `-savename` is *not* specified, then GDSCtl replaces what you specify for the net service name with the full connection string before saving the configuration to the catalog.
- The default for GDSCtl is for `autovncr` to be enabled for the catalog. If `autovncr` has been disabled for the catalog, before configuring Global Data Services pools and adding databases to the Global Data Services configuration, the nodes where those databases run must be part of the valid node checking for registration (VNCR) list for database registration. Use the `add invitednode` (`add invitedsubnet`) command to define the valid nodes.

Example

Adds database DB1 to the WEST region and Global Data Services pool MYREADERFARM.

```
GDSCtl> add database -connect 127.0.0.1:1521:db1 -region west -gdspool
myreaderfarm
```

Adds a database using *myalias* instead of the IP address connection string.

```
GDSCtl> add database -connect myalias -gdspool myreaderfarm
```

Exceptions or Error Codes

GDSCtl returns the errors listed in [Table A-8](#) if you use this command incorrectly.

Table A-8 *gdscctl add database Exceptions or Error Codes*

Exception	Description
ERROR-44866	If a pool already contains an Oracle Data Guard broker configuration, then GDSCtl returns an error; you must add databases using Oracle Data Guard in this case. That is, if a pool contains an Oracle Data Guard broker configuration, then additional databases can only be added to the pool by adding them to that DG broker configuration.
ERROR-44868	If the database being added is part of a Oracle Data Guard broker configuration, then GDSCtl returns an error; you must use the <code>add brokerconfig</code> command in this case.

add gdspool

Adds a Global Data Services pool to the Global Data Services framework. When the Global Data Services catalog is created, a default pool named DBPOOLORA is created, automatically.

Syntax and Options

Use the `gdscctl add gdspool` command with the following syntax:

```
add gdspool -gdspool database_pool_list [-users user_list]
```

Table A–9 *gdsctl add gdspool Options*

Option	Description
<code>-gdspool database_pool_list</code>	A comma-delimited list of Global Data Services pool names. A Global Data Services pool must have a unique name within its GDS configuration. If you do not specify a name for the pool when you create it, then the name defaults to <code>oradbpool</code> . The pool name can be up to 30 bytes long and can be any valid identifier (an alphabetical character followed by zero or more alphanumeric ASCII characters or the underscore (<code>_</code>)).
<code>-users user_list</code>	A comma-delimited list of users that are granted the pool administrator role.

Usage Notes

- You must connect to the Global Data Services catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command.
- The default for GDSCTL is for `autovncr` to be enabled for the catalog. If `autovncr` has been disabled for the catalog, then before configuring Global Data Services pools and adding databases to the Global Data Services configuration, the nodes where those databases run must be part of the valid node checking for registration (VNCR) list for database registration. Use the `add invitednode` (`add invitedsubnet`) command to define the valid nodes.

Example

Add a Global Data Services pool named MYREADERFARM to the configuration:

```
GDSCTL> add gdspool -gdspool myreaderfarm
```

add gsm

Adds a global service manager to the Global Data Services framework. You must specify the Global Data Services catalog database when using this command.

Syntax and Options

Use the `gdsctl add gsm` command with the following syntax:

```
add gsm -gsm gsm_name -catalog connect_id [-pwd password] [-wpwd password]
      [-region region_name] [-localons ons_port] [-remoteons ons_port]
      [-listener listener_port] [-endpoint gmsendpoint]
      [-remote_endpoint remote_endpoint] [-trace_level level]
```

Table A–10 *gdsctl add gsm Options*

Option	Description
<code>-catalog connect_id</code>	Specify the connect identifier for the Global Data Services catalog database. If a network service name is specified, it must be resolvable by the local naming method to a connect descriptor that allows the global service manager being added to connect to the catalog database.

Table A-10 (Cont.) gdsctl add gsm Options

Option	Description
<code>-endpoint gsmendpoint</code>	Specifies the protocol address that the global services manager listens on for client connection requests. If you use this option, the value that you specify overrides the default endpoint.
<code>-gsm gsm_name</code>	Specify the name of the global service manager that you want to add. If you do not specify a name, then GDSCCTL uses the current global service manager name for the session (specified with the <code>set gsm</code> command).
<code>-listener listener_port</code>	Specify the listener port. The default port is 1522.
<code>-localons ons_port</code>	Specify the local ONS port. If you do not specify this option, then GDSCCTL uses the default ONS port (which is 6123 on most platforms).
<code>-pwd password</code>	Specify the password for the <code>GSMCATUSER</code> . If you do not specify a password, then you are prompted to enter one.
<code>-region region_name</code>	Specify the region to which the global service manager belongs. The value for <code>region_name</code> must match the name of an existing Global Data Services region. If you do not specify a region, then GDSCCTL adds the global service manager without assigning a region.
<code>-remote_endpoint remote_endpoint</code>	Specifies the protocol address that is used by the global service manager to receive database registration requests and communicate with other global service managers in the configuration. If you use this option, the value that you specify overrides the default endpoint.
<code>-remoteons ons_port</code>	Specify the remote ONS port. If you do not specify this option, then GDSCCTL uses the default ONS port (which is 6234 on most platforms).
<code>-trace_level level</code>	Specify the global service manager trace level (to be used as directed by Oracle Support Services).
<code>-wpwd password</code>	Specify a password to protect the global service manager wallet. If a wallet password is not specified, a system-generated password is used instead. Note that if a password is specified with this option, the wallet cannot be modified without supplying that password.

Usage Notes

- You must run this command, locally, on the computer where you want to add the global service manager.
- You must have operating system privileges on the computer where you want to add the global service manager to run this command.
- When you run this command, GDSCCTL connects to the Global Data Services catalog as the `GSMCATUSER` user and prompts you for the `GSMCATUSER` password.

Example

Add a global service manager named `gsm1`, specifying the location of the Global Data Services catalog database, `DB1`.

```
GDSCTL> add gsm -gsm gsm1 -catalog 127.0.0.1:1521:db1
```

add invitednode (add invitedsubnet)

Adds host address or subnet information to the valid node checking for registration (VNCR) list in the catalog. This command enables you to add a host or a subnet to the VNCR list before starting the first global service manager (by establishing a direct connection to the Global Data Services catalog database).

VNCR enables or denies access from specified IP addresses to Oracle services. See *Oracle Database Net Services Administrator's Guide* for more information about VNCR.

Syntax and Options

Use the `gdsctl add invitednode` command with the following syntax:

```
add {invitednode | invitedsubnet} [-group group_name]
    [-catalog catalog_dbname [-user user_name/password]] vncr_id
```

Table A-11 *gdsctl add invitednode Options*

Option	Description
<code>-catalog catalog_dbname</code>	Specify the Global Data Services catalog database net alias or connect string. If you enter an invalid address or connect string, then GDSCTL uses the pre-established connection created with the <code>connect</code> command.
<code>-group group_name</code>	Specify an alias which defines a group of invited nodes. This alias can be referenced in other commands related to invited nodes.
<code>-user user_name[/password]</code>	Specify the user credentials for the Global Data Services administrator in the catalog database. If you do not specify a user or a password, then GDSCTL prompts you this information.
<code>vncr_id</code>	Specify the list of nodes that can register with the global service manager. The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

Usage Notes

- You must connect to the Global Data Services catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.
- The default for GDSCTL is that `autovncr` is enabled for the catalog. If `autovncr` has been disabled for the catalog, before configuring Global Data Services pools and adding databases to the Global Data Services configuration, then the nodes where those databases run must be part of the valid node checking for registration (VNCR) list for database registration. Use the `add invitednode` (`add invitedsubnet`) command to define the valid nodes.

Examples

Add the netmask `255.255.255.248` to the catalog.

```
GDSCTL> add invitednode 255.255.255.248
```

Add the server east1.example.com to the catalog in the alias group EAST_SRV.

```
GDSCTL> add invitednode east1.example.com
```

Add the server east2.example.com to the catalog in the alias group EAST_SRV.

```
GDSCTL> add invitednode east2.example.com
```

add region

Adds a Global Data Services region to the Global Data Services framework. When the Global Data Services catalog is created, the REGIONORA region is created automatically.

Syntax and Options

Use the `gdsctl add region` command with the following syntax:

```
add region -region region_list [-buddy region_name]
```

Table A–12 *gdsctl add region Options*

Option	Description
<code>-buddy <i>region_name</i></code>	Specify the name of the buddy region.
<code>-region <i>region_list</i></code>	Specify a comma-delimited list of Global Data Services region names. A Global Data Services region should have a name that is unique within the corresponding Global Data Services configuration. If no name is specified at the first region creation time, the default name, oraregion, is given to the region. The region name can be up to 30 characters long and can be any valid identifier - an alphabetical character followed by zero or more alphanumeric ASCII characters or '_'.

Usage Notes

- You must connect to the Global Data Services catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command

Example

Add two Global Data Services regions, EAST and WEST to the current configuration:

```
GDSCTL> add region -region east,west
```

add service

Add a global service to a Global Data Services pool.

Syntax and Options

Use the `gdsctl add service` command with the following syntax:

```
add service [-gdspool gdspool_name] -service service_name
  {-preferred_all | -preferred dbname_list [-available dbname_list] }
  [-locality {ANYWHERE | LOCAL_ONLY [-region_failover]}]
  [-role {PRIMARY | PHYSICAL_STANDBY [-failover_primary] | LOGICAL_STANDBY
  | SNAPSHOT_STANDBY}] [-lag {lag_value | ANY}]
  [-notification {TRUE | FALSE}] [-rlbgoal {SERVICE_TIME | THROUGHPUT}]
```

```

[-dtp {TRUE | FALSE}] [-sql_translation_profile stp_name]
[-clbgoal {SHORT | LONG}] [-tafpolicy {BASIC | NONE | PRECONNECT}]
[-policy policy] [-failovertype {NONE | SESSION | SELECT | TRANSACTION}]
[-failovermethod {NONE | BASIC}] [-failoverretry failover_retries]
[-failoverdelay failover_delay] [-edition edition_name]
[-commit_outcome {TRUE | FALSE}] [-retention retention_seconds]
[-session_state {DYNAMIC | STATIC}]
[-replay_init_time replay_init_time] [-pdbname pdbname]

```

Table A-13 *gdsctl add service Options*

Option	Description
-available <i>dbname_list</i>	<p>Specify a comma-delimited list of available databases on which the service runs if the preferred databases are not available. You <i>cannot</i> specify a list of available instances, only databases. You can use the modify service command with the <code>-server_pool</code> parameter to specify instance-level preferences.</p> <p>The list of available databases must be mutually exclusive with the list of preferred databases.</p> <p>You <i>cannot</i> use this option with the <code>-preferred_all</code> option.</p>
-clbgoal {SHORT LONG}	<p>Connection Load Balancing Goal. Use a value of <code>SHORT</code> for this parameter for run-time load balancing, or if using an integrated connection pool. Use a value of <code>LONG</code> for this parameter for long running connections, such as batch jobs, that you want balanced by the number of sessions for each node for the service.</p> <p>The default value for this option is <code>SHORT</code>.</p>
-commit_outcome {TRUE FALSE}	<p>Enable Transaction Guard; when set to <code>TRUE</code>, the commit outcome for a transaction is accessible after the transaction's session fails due to a recoverable outage.</p>
-dtp {TRUE FALSE}	<p>Indicates whether Distributed Transaction Processing should be enabled for this service. This service can either be a service in a policy-managed database or a preferred service on a single node in an administrator-managed database.</p>
-edition <i>edition_name</i>	<p>Specify the initial session edition of the service.</p> <p>When an edition is specified for a service, all subsequent connections that specify the service use this edition as the initial session edition. However, if a session connection specifies a different edition, then the edition specified in the session connection is used for the initial session edition.</p> <p>GDSCTL does not validate the specified edition name. During connection, the connect user must have <code>USE</code> privilege on the specified edition. If the edition does not exist or if the connect user does not have <code>USE</code> privilege on the specified edition, then an error is raised.</p>
-failover_primary	<p>If you set the <code>-role</code> option to <code>PHYSICAL_STANDBY</code>, then you can use this option to enable the service for failover to the primary database.</p>

Table A-13 (Cont.) gdsctl add service Options

Option	Description
<code>-failoverdelay</code> <i>failover_delay</i>	For Application Continuity and TAF, this parameter specifies the time delay (in seconds) between reconnect attempts for each incident at failover.
<code>-failovermethod</code> {NONE BASIC}	TAF failover method (for backward compatibility only). If the failover type (<code>-failovertype</code>) is set to a value other than NONE, then you should choose BASIC for this parameter.
<code>-failoverretry</code> <i>failover_retries</i>	For Application Continuity and TAF, this parameter determines the number of attempts to connect after an incident.
<code>-failovertype</code> {NONE SESSION SELECT TRANSACTION}	Specify the failover type. To enable Application Continuity for Java, set this parameter to TRANSACTION. To enable Transparent Application Failover (TAF) for OCI, set this parameter to SELECT or SESSION.
<code>-gdspool</code> <i>gdspool_name</i>	Specify the name of the Global Data Services pool to which you want to add a service. If the pool name is not specified and there is only one <code>gdspool</code> with access granted to the user, then this the <code>gdspool</code> with access granted is used as the default <code>gdspool</code> .
<code>-lag</code> { <i>lag_value</i> ANY}	Specify the lag for the service in seconds. You can use the keyword ANY to indicate that there is no upper threshold on the lag time. This parameter specifies the maximum lag that a provider of this service may have. The service cannot be provided by a database whose lag exceeds this value. The default value for <code>lag</code> is ANY.
<code>-locality</code> {ANYWHERE LOCAL_ONLY}	Specify the service region locality. If you do not specify this option, then GDSCTL uses the default value of ANYWHERE for the service.
<code>-notification</code> {TRUE FALSE}	Enable Fast Application Notification (FAN) for OCI connections.
<code>-pdbname</code> <i>pdb_name</i>	Specify the pluggable database name.
<code>-policy</code> {AUTOMATIC MANUAL}	Specify the management policy for the service. If you specify AUTOMATIC (the default), then the service automatically starts when the database restarts, either by a planned restart or after a failure. Automatic restart is also subject to the service role. If you specify MANUAL, then the service is never automatically restarted upon planned restart of the database. A MANUAL setting does not prevent the global service manager from monitoring the service when it is running and restarting it if a failure occurs.

Table A-13 (Cont.) gdsctl add service Options

Option	Description
<code>-preferred dbname_list</code>	<p>Specify a comma-delimited list of preferred databases on which the service runs. You <i>cannot</i> specify preferred instances, only databases. You can use the <code>modify service</code> command to specify instance-level preferences.</p> <p>The list of preferred databases must be mutually exclusive with the list of available databases.</p> <p>You <i>cannot</i> use this option with the <code>-preferred_all</code> option.</p>
<code>-preferred_all</code>	<p>Specifies that all the databases in the Global Data Services pool are preferred databases. Any databases you later add to the pool are configured as preferred databases for this service.</p> <p>You <i>cannot</i> use this option with the <code>-preferred</code> and <code>-available</code> options.</p>
<code>-region_failover</code>	<p>Indicates that the service is enabled for region failover. You can only use this option when you specify <code>LOCAL_ONLY</code> for the <code>-locality</code> option.</p>
<code>-replay_init_time replay_init_time</code>	<p>For Application Continuity, this parameter specifies the time (in seconds) after which replay cannot be initiated. The default value is 300 seconds.</p>
<code>-retention retention_seconds</code>	<p>If <code>commit_outcome</code> is set to <code>TRUE</code>, then this parameter determines the amount of time (in seconds) that the commit outcome is retained in the database.</p>
<code>-rlbgoal {SERVICE_TIME THROUGHPUT}</code>	<p>Run-time Load Balancing Goal (for the Load Balancing Advisory). Set this parameter to <code>SERVICE_TIME</code> to balance connections by response time. Set this parameter to <code>THROUGHPUT</code> to balance connections by throughput.</p> <p>If you do not use this option, then the value defaults to <code>SERVICE_TIME</code> for the run-time load balancing goal.</p>
<code>-role {[PRIMARY] [PHYSICAL_STANDBY] [-failover_primary] [LOGICAL_STANDBY] [SNAPSHOT_STANDBY]}</code>	<p>Specify the database role that the database must be for this service to start on that database. This applies only to Global Data Services pools that contain an Oracle Data Guard broker configuration.</p> <p>See Also: <i>Oracle Data Guard Concepts and Administration</i> for more information about database roles</p>

Table A-13 (Cont.) gdsctl add service Options

Option	Description
-service <i>service_name</i>	<p>Specify the name of the global service.</p> <p>The service name specified in the <code>add service</code> command can be domain qualified (for example, <code>sales.example.com</code>) or not (for example, <code>sales</code>). If the specified name is not domain qualified, the service is created with the default domain name <code><GDS_pool_name>.<GDS_configuration_name></code>, however the shorter non-domain qualified name can be used with subsequent <code>gdsctl</code> commands to manage the service. If the specified name is domain qualified, the full domain qualified service name must be used in all subsequent <code>gdsctl</code> commands used to manage the service.</p> <p>A global service name must be unique within a GDS pool and when qualified by domain, must also be unique within a GDS configuration. A global service cannot be created at a database if a local or global service with the same name already exists at that database.</p> <p>A global service name can contain alphanumeric characters, underscore (<code>_</code>), and period (<code>.</code>). The first character must be alphanumeric. The maximum length of a global service name is 64 characters. The maximum length of a domain qualified global service name is 250 characters.</p> <p>An Oracle Net connect descriptor used to connect to a global service must contain a domain qualified service name.</p>
-session_state {DYNAMIC STATIC}	<p>For Application Continuity, this parameter specifies whether the session state that is not transactional is changed by the application. A setting of <code>DYNAMIC</code> is recommended for most applications.</p>
-sql_translation_profile <i>stp_name</i>	<p>Use this option to specify a SQL translation profile for a service that you are adding after you have migrated applications from a non-Oracle database to an Oracle database.</p> <p>This option corresponds to the SQL translation profile parameter in the <code>DBMS_SERVICE</code> service attribute.</p> <p>Notes:</p> <ul style="list-style-type: none"> ■ Before using the SQL translation feature, you must migrate all server-side application objects and data to the Oracle database. ■ Use the <code>config service</code> command to display the SQL translation profile.
-tafpolicy {BASIC NONE }	<p>TAF policy specification (for administrator-managed databases only).</p>

Examples

Add a service named `sales_report` to the Global Data Services pool `MYREADERFARM` with a value of `ANYWHERE` for the locality.

```
GDSCTL> add service -gdspool myreaderfarm -service sales_report -locality ANYWHERE
```

Add a service named `daily_sales_rept` to the Global Data Services pool `MYDGPOOL` with preferred instance set to `DB1` and the available instances set to `DB3` and `DB4`. The service should use the basic transaction failover policy.

```
GDSCTL> add service -gdspool mydgpool -s daily_sales_rept -preferred db1  
-available db3,db4 -tafpolicy BASIC
```

See Also: ["Creating a Global Service"](#) on page 3-4

config

The `gdsctl config` command displays the static configuration of the specified component or set of components. This configuration data is retrieved from the catalog database. You can also use this command with no options to display the configuration data for all components defined for the configuration. When using the `config` command, it does not matter if the components (except for the catalog database) are started.

Table A-14 *gdsctl config Summary*

Command	Description
<code>config database</code> on page A-22	Displays the static configuration data for the specified database.
<code>config gdspool</code> on page A-23	Displays the static configuration data for the specified database pool.
<code>config gsm</code> on page A-24	Displays the static configuration data for the specified global service manager.
<code>config region</code> on page A-24	Displays the static configuration data for the specified region.
<code>config service</code> on page A-25	Displays the static configuration data for the specified region.
<code>config vncr</code> on page A-26	Displays the VNCR configuration data for the specified global service manager.

config database

Displays the static configuration data stored in the catalog for the specified database.

Syntax and Options

Use the `gdsctl config database` command with the following syntax:

```
config database [-database db_name]
```

Table A-15 *gdsctl config database Options*

Syntax	Description
<code>-database db_name</code>	Specify the name of a database. If you do not specify a database name, then GDSCTL displays the configuration data for all databases in the Global Data Services configuration.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Examples

Display the static configuration data stored in the catalog for all the databases in the Global Data Services configuration.

```
GDSCTL>config database
```

The `gdsctl config database` command returns information similar to the following:

```
Name      Pool      Status    Region
```

```

-----
dbcat    sales    Ok        east
dbcat1   sales    Ok        west
dbcat3   sales    Ok        west

```

config gdspool

Displays the static configuration data that is stored in the catalog for the specified database pool.

Syntax and Options

Use the `gdsctl config gdspool` command with the following syntax:

```
config gdspool [-gdspool gdspool_name]
```

Table A-16 *gdsctl config gdspool Options*

Syntax	Description
<code>-gdspool <i>gdspool_name</i></code>	Specify the name of a database pool. If you do not specify a database pool name, then GDSCTL displays the configuration data for all database pools.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command

Example

Display the static configuration data stored in the catalog for all Global Data Services pools.

```
GDSCTL>config gdspool
```

The `gdsctl config gdspool` command returns output similar to the following:

```

Name          Broker
-----
dbpoolora     No
mkt           No
sales         No
marketing     No

```

The following command shows the configuration detail of Global Data Services pool `marketing`.

```
GDSCTL>config gdspool -gdspool marketing
```

The above example returns output similar to the following:

```

GDS Pool administrators
-----

Databases
-----
dbcat2
dbcat1
dbcat3

Services
-----

```

```
sales_report
sales_analysis
sales_estimation
sales_peragent
sales_global
```

config gsm

Displays the static configuration data stored in the catalog for the specified global service manager.

Syntax and Options

Use the `gdctl config gsm` command with the following syntax:

```
config gsm [-gsm gsm_name]
```

Table A-17 *gdctl config gsm Options*

Syntax	Description
<code>-gsm gsm_name</code>	Specify the name of a global service manager. If you do not specify a global service manager name, then GDSCTL displays the static configuration data for all global service managers in the cloud.

Usage Notes

- You must connect to the catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command

Example

Display the static configuration data stored in the catalog for the global service manager `mygsm`:

```
GDSTL>config gsm -gsm mygsm
```

The `gdctl config gsm` command returns information similar to the following:

```
Name: mygsm
Endpoint 1: (ADDRESS=(HOST=stcal.us.hq.com) (PORT=1523) (PROTOCOL=tcp))
Endpoint 2: (ADDRESS=(HOST=stcal.us.hq.com) (PORT=1523) (PROTOCOL=tcp))
Local ONS port: 6123
Remote ONS port: 6234
Region: east
Buddy
-----
```

config region

Displays the static configuration data for the specified region.

Syntax and Options

Use the `gdctl config region` command with the following syntax:

```
config region [-region gsm_name]
```

Table A-18 *gdctl config region Options*

Syntax	Description
<code>-region gsm_name</code>	Specify the name of a global service manager.

Example

Displays the static configuration data for the specified region.

```
GDSCTL>config region -region eastcoast
```

Displays the following output:

```
Databases
-----
debug
msvc4

GSM
-----
mygsm
```

config service

Displays the static configuration data stored in the Global Data Services catalog for the specified services that are located in a database pool.

Syntax and Options

Use the `gdsctl config service` command with the following syntax:

```
config service [-gdspool gdspool_name] [-service service_name]
```

Table A-19 *gdsctl config service Options*

Syntax	Description
<code>-gdspool <i>gdspool_name</i></code>	Specify the name of the database pool that contains the services. If the name is not specified, and there is only one <i>gdspool</i> with access granted to the user, it is used as the default <i>gdspool</i> .
<code>-service <i>service_name</i></code>	Specify a comma-delimited list of service names. If you do not use this option, then GDSCTL displays the configuration data for all services in the specified database pool.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Example

Show all the services in the user's Global Data Services pool:

```
GDSCTL>config service
```

The `gdsctl config service` command returns information similar to the following:

```
Name          Network name          Pool    Started Preferred all
----          -
sales_svc1    sales_svc1.sales.oradbcloud  sales  Yes    Yes
sales_svc2    sales_svc2.sales.oradbcloud  sales  NO     Yes
sales_svc3    sales_svc3.sales.oradbcloud  sales  Yes    Yes
mkt_svc1      mkt_svc1.mkt.oradbcloud     mkt    NO     Yes
```

Display the static configuration data stored in the Global Data Services catalog for `sales_svc1`:

```
GDSCTL>config service -service sales_svc1
```

returns:

```

Name: sales_svc1
Network name: sales_svc1.sales.oradbcloud
Pool: sales
Started: Yes
Preferred all: Yes
Locality: ANYWHERE
Region Failover: No
Role: NONE
Primary Failover: No
Lag: ANY
Runtime Balance: SERVICE_TIME
Connection Balance: SHORT
Notification: Yes
TAF Policy: NONE
Policy: AUTOMATIC
DTP: No
Failover Method: NONE
Failover Type: NONE
Failover Retries:
Failover Delay:
Edition:
PDB:
Commit Outcome:
Retention Timeout:
Replay Initiation Timeout:
Session State Consistency:
SQL Translation Profile:

```

Supported services

```

-----
Database          Preferred Status
-----
dbcat2            Yes      Enabled
dbcat1            Yes      Enabled
dbcat3            Yes      Enabled

```

config vnrcr

Displays the static configuration data stored in the catalog for valid node checking for registration (VNCR).

Syntax and Options

Use the `gdsctl config vnrcr` command with the following syntax:

```
config vnrcr [-group gsm_name]
```

Table A-20 *gdsctl config invitednode Options*

Syntax	Description
<code>-group <i>gsm_name</i></code>	Specify the name of a global service manager.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Example

Displays the list of hosts in the VNCR list:

```
GDCTL>config vncr
```

The `gdctl config vncr` command returns information similar to the following:

Name	Group	ID
----	-----	--
example		
22.231.113.64		
example.com		

configure

The `gdsctl configure` command sets the GDSCTL parameters.

Syntax and Options

Use the `gdsctl configure` command with the following syntax:

```
configure [-gsmport port] [-timeout seconds] [-show] [-driver {THIN | OCI}]
  [-resolve {IP | HOSTNAME | QUAL_HOSTNAME}]
  [-log {ALL|OFF|INFO|FINE|FINER|FINEST|SEVERE|WARNING}] [-log_file log_file]
  [-gsm gsm_name]
```

Table A-21 *gdsctl configure Options*

Syntax	Description
<code>-driver THIN OCI</code>	Oracle JDBC driver.
<code>-gsm <i>gsm_name</i></code>	Set current global service manager.
<code>-gsmport <i>port</i></code>	Default global service manager port.
<code>-log {ALL OFF INFO FINE FINER FINEST SEVERE WARNING}</code>	Set the logging level. The default is OFF.
<code>-log_file <i>log_file</i></code>	Set the location of the log file. The default is <code>\$TNS_ADMIN/GDSTL.log</code> .
<code>-resolve IP HOSTNAME QUAL_HOSTNAME</code>	Default host resolution for global service manager endpoint.
<code>-show</code>	Show the configuration.
<code>-timeout <i>seconds</i></code>	Global service manager requests timeout in seconds.

Example

Set the `mygsm` driver to OCI:

```
configure -driver OCI mygsm
```

connect

The `gdsctl connect` command specifies the credentials to administer a global service management environment. Credentials must be specified to perform certain operations using GDSCTL.

Syntax and Options

Use the `gdsctl connect` command with the following syntax:

```
connect [user_name[/password]]@connect_identifier
```

Table A–22 *gdsctl connect Options*

Syntax	Description
<i>connect_identifier</i>	Specify an Oracle Net connect descriptor or a net service name that maps to a connect descriptor (for example, a list of global service managers).
<i>password</i>	Specify the password for the specified user. If you do not specify a password, then you are prompted to enter a password. The password is obscured when entered.
<i>user_name</i>	Specify the name of the user to connect as. The user that you specify must have either the Global Data Services administrator or the pool administrator role. If you specify no user name, then you are prompted for a user name.

Usage Notes

- You must connect to the catalog database as a user with either the Global Data Services administrator or the pool administrator privileges, depending on which command you want to run after you connect

WARNING: Specifying a password as a connect command option is a security risk. You can avoid this risk by omitting the password, and entering it only when the system prompts for it.

Examples

Connect as the `gsadmin` user to the private cloud:

```
GDSCTL> connect gsadmin@mycloud
Enter password:
```

Connect using a connect descriptor, without specifying a user name and password:

```
GDSCTL> connect (DESCRIPTION= (ADDRESS= (PROTOCOL=tcp) (HOST=myhost) (PORT=1521)))
Enter username:
```

create catalog

The `gdsctl create catalog` command creates a Global Data Services catalog for global service management in a specific database. It also generates a pair of PKI public and private keys and stores them in the catalog, along with a fixed string "GSM" that is encrypted with a private key. It uses the `GSMCATUSER` password.

Syntax and Options

Use the `gdsctl create catalog` command with the following syntax:

```
create catalog -database connect_id [-user user_name[/password]
  [-region region_list] [-gdspool gdspool_list] [-configname config_name]
  [-autovncr {ON | OFF}] [-force]
```

Table A-23 *gdsctl create catalog Options*

Syntax	Description
<code>-autovncr {ON OFF}</code>	This option enables (ON) or disables (OFF) autovncr mode. The default value is ON.
<code>-configname config_name</code>	Specify the name of the GDS configuration. The default configuration name is <code>ORADBCLOUD</code> . The configuration name can be up to 32 bytes long and can contain an alphabetical character followed by zero or more alphanumeric ASCII characters, <code>'_'</code> , or <code>'#'</code> and possibly separated by periods if there are multiple identifiers.
<code>-database connect_id</code>	Specify the connect identifier for the database in which you want to create catalog.
<code>-force</code>	Rewrites existing global service manager configuration on catalog database.
<code>-gdspool gdspool_list</code>	Specify a comma-delimited list of database pool names. When you use this option, the specified database pools are created as part of the catalog creation. If you do not specify this option, then <code>GDSCTL</code> creates a default database pool named <code>DBPOOLORA</code> .
<code>-region region_list</code>	Specify a comma-delimited list of region names. This command creates each region and adds the regions to the catalog. If you do not specify a region, then a default region named <code>REGIONORA</code> is created.
<code>-user user_name[/password]</code>	Specify a user (and optionally, the password) that has the Global Data Services administrator privileges on the catalog database. If you do not use this option, then <code>GDSCTL</code> prompts you for the name and the password of a user with Global Data Services administrator privileges. If you specify a user name but not the password for the user, then <code>GDSCTL</code> prompts you for the password.

Usage Notes

- You must have the Global Data Services administrator privileges on the computer where you want to create the Global Data Services catalog.
- Auto VNCr is best used in environments with simple private networks where ease of configuration is the most important consideration. To have the highest level of control over which hosts may participate in a GDS configuration, disable

Auto VNCR and explicitly add the IP address(es) of each database host to the VNCR configuration.

Example

Create a Global Data Services catalog for global service management in the database named DB1. Also create the regions EAST and WEST, and the database pool READERFARM.

```
GDSCTL> create catalog -database db1 -region west,east -gdspool readerfarm
```

delete catalog

The `gdsctl delete catalog` command deletes the Global Data Services catalog.

Syntax and Options

Use the `gdsctl delete catalog` command with the following syntax:

```
delete catalog [-connect connect_identifier]
```

Table A–24 *gdsctl delete catalog Options*

Syntax	Description
<code>-connect <i>connect_identifier</i></code>	Specify an Oracle Net connect descriptor or a net service name that maps to a connect descriptor. If you do not use this option, then GDSCTL deletes the Global Data Services catalog that is used by the global service manager associated with the current session.

Usage Notes

- You must have the Global Data Services administrator privileges on the computer where the database resides from which you want to delete the Global Data Services catalog

Example

Delete the Global Data Services catalog located in the database named DB1.

```
GDSCTL> delete catalog -connect db1.example.com
```

disable service

The `gdsctl disable service` command disables specified services. The specified services are also stopped.

Syntax and Options

Use the `gdsctl disable service` command with the following syntax:

```
disable service [-gdspool gdspool_name] [ -service service_name_list]
               -database db_name
```

Table A–25 *gdsctl disable service Options*

Syntax	Description
<code>-database <i>db_name</i></code>	Specify the name of the database on which to the service is located. If you do not specify this option, then the service is disabled, globally.
<code>-gdspool <i>gdspool_name</i></code>	Specify the database pool in which the services are located. If not specified and there is only one <code>gdspool</code> with access granted to the user, then this one is used as the default <code>gdspool</code> .
<code>-service <i>service_name_list</i></code>	Specify a comma-delimited list of global service names. If you do not use <code>-service</code> to specify an individual global service or to specify a list of global services, then all the services in the database pool are disabled.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command

Example

Disable and stop the service `G_SALES_REPORT` on all databases in the database pool `READERFARM`.

```
GDSCTL> disable service -gdspool readerfarm -service g_sales_report
```

See Also: ["Disabling a Global Service"](#) on page 3-6

enable service

The `gdsctl enable service` command enables the specified services. If the cardinality for a service has not been reached, then this command also starts that service.

Syntax and Options

Use the `gdsctl enable service` command with the following syntax:

```
enable service [-gdspool gdspool_name] [-service service_name] -database db_name
```

Table A–26 *gdsctl enable service Options*

Syntax	Description
<code>-database <i>db_name</i></code>	Specify the name of the database on which the service is located. If you do not specify this option, then the service is enabled globally.
<code>-gdspool <i>gdspool_name</i></code>	Specify the database pool in which the services are located. If not specified and there is only one <i>gdspool</i> with access granted to the user, it is used as the default <i>gdspool</i> .
<code>-service <i>service_name</i></code>	Specify a comma-delimited list of global service names. If you do not use <code>-service</code> to specify an individual global service or to specify a list of global services, then all the services in the database pool are disabled.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Example

Enable the service `G_SALES_REPORT` on the database `DB1` in the database pool `READERFARM`.

```
GDSCTL> enable service -gdspool readerfarm -service g_sales_report -database db1
```

See Also: ["Enabling a Global Service"](#) on page 3-6

export catalog

The `gdsctl export catalog` command saves the current catalog configuration to a local file.

Syntax and Options

Use `gdsctl export catalog` command with the following syntax:

```
export catalog file_name
```

Table A-27 *gdsctl export catalog Options*

Syntax	Description
<i>file_name</i>	Name of a file on the same computer where the command is being executed. The configuration will be saved to this file. If the file already exists, it will be overwritten without a prompt. If the file is not writable (for example the path does not exist), you will get an error.

Usage Notes

- You must connect to the catalog database as a user with GDS Administrator privileges before running this command.
- It is recommended that you validate the catalog, using the `validate catalog` command before exporting it.

Example

Save the catalog backup to your home directory.

```
GDSCTL> export catalog /home/user/cat-201307.backup
```

import catalog

The `gdsctl import catalog` command is used to restore catalog configuration from the file, created using `export catalog` command. When restoring to a new catalog database, catalog must be created first, using `create catalog` command.

Syntax and Options

Use `gdsctl import catalog` command with the following syntax:

```
import catalog file_name
```

Table A-28 *gdsctl export catalog Options*

Syntax	Description
<i>file_name</i>	Name of a file on the same computer where the command is being executed. The configuration will be restored from this file. If the file is not readable, you will get an error.

Usage Notes

- You must connect to the catalog database as a user with GDS Administrator privileges before running this command.
- The import procedure can be considered finished only when there are no pending requests after import. Use the `config` command to get the list of pending requests.

Example

Load the catalog backup from your home directory.

```
GDSCTL> import catalog /home/user/cat-201307.backup
```

modify

The `gdctl modify` command changes the configuration parameters for components in the global service manager configuration.

Table A–29 *gdctl modify Summary*

Command	Description
<code>modify catalog</code> on page A-37	Modifies the catalog attributes.
<code>modify database</code> on page A-38	Modifies the configuration parameters of the databases
<code>modify gdspool</code> on page A-39	Add or remove a user from the list of database pool administrators
<code>modify gsm</code> on page A-40	Modifies the configuration options for a global service manager
<code>modify region</code> on page A-41	Modifies the regions currently defined for the global service management framework
<code>modify service</code> on page A-41	Modifies the preferred or available databases for a service, or other service attributes

modify catalog

The `gdctl modify catalog` command modifies the properties of the global service manager catalog. To use this command, there must be a least one global service manager running and a connection with the catalog database must have already been established (see the `connect` command).

Syntax and Options

Use the `gdctl modify catalog` command with the following syntax:

```
modify catalog [-autovncr {ON | OFF}] [-oldpwd oldpassword -newpwd newpassword]
[-pwd password -newkeys]
```

To update the `GSMCATUSER` password after changing it on the catalog database use

```
modify catalog -oldpwd ** -newpwd ***
```

If you decide to replace the PKI keys, or just after the patchset upgrade from Oracle Database 12c Release 1 (12.1) on the catalog database, run this command:

```
modify catalog -pwd ** -newkeys
```

Table A–30 *gdctl modify catalog Options*

Syntax	Description
<code>-autovncr {ON OFF}</code>	This option enables (ON) or disables (OFF) <code>autovncr</code> mode. The default value is ON.
<code>-newkeys</code>	Generates a new PKI key pair.
<code>-newpwd <i>newpassword</i></code>	Used along with <code>-oldpwd</code> , sets the <code>GSMCATUSER</code> password after changing it on the catalog database.
<code>-oldpwd <i>oldpassword</i></code>	Used along with <code>-newpwd</code> , sets the <code>GSMCATUSER</code> password after changing it on the catalog database.
<code>-pwd <i>password</i></code>	Provides the <code>GSMCATUSER</code> password to generate the PKI keys when using <code>-newkeys</code> .

Usage Notes

- You must connect to the catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command.
- Auto VNCR is best used in environments with simple private networks where ease of configuration is the most important consideration. To have the highest level of control over which hosts may participate in a GDS configuration, disable Auto VNCR and explicitly add the IP address(es) of each database host to the VNCR configuration.
- The `GSMCATUSER` password should be updated regularly for security reasons. Use `modify catalog -oldpwd oldpassword -newpwd newpassword` to perform this operation. This command fetches the encrypted private key and enc string, decrypts them using the old password, re-encrypts them with the new password and stores them again.
- The PKI keys must be updated regularly, which is done using `modify catalog -oldpwd oldpassword -newkeys`. This command generates a new PKI key pair and replaces the corresponding fields in the database.

Example

Turn off `autovncr` mode for the catalog database.

```
connect gsmadmin@mycloud
GDSCTL> modify catalog -autovncr off
```

modify database

Modifies the configuration parameters of the databases in a GDS pool, such as region, connect identifier, global service manager password, SCAN address, and ONS port. For all parameters except for the GDS region, first the appropriate changes must be done by the database administrator and then the `modify database` command must be run to update the modified parameters in the GDS catalog. Alternatively, you can use the `sync database` command for this purpose.

Syntax and Options

Use the `gdsctl modify database` command with the following syntax:

```
modify database -database dbname_list [-region region_name]
               [-gdspool gdspool_name] [-connect connect_identifier] [-pwd password]
               [-scan scan_address [-ons port]] [-savename] [-cpu_threshold cpu]
               [-disk_threshold disk]
```

Table A-31 *gdsctl modify database Options*

Option	Description
<code>-connect connect_identifier</code>	Specify an Oracle Net connect descriptor or a net service name that maps to a connect descriptor for the database that is being modified.
<code>-cpu_threshold cpu</code>	Specifies CPU Utilization percentage threshold.
<code>-database dbname_list</code>	Specify a comma-delimited list of database names.
<code>-disk_threshold disk</code>	Specifies the average latency in milliseconds of a synchronous single-block read.
<code>-gdspool gdspool_name</code>	Specify the database pool to which the databases belong.

Table A-31 (Cont.) gdsctl modify database Options

Option	Description
<code>-ons port</code>	Specify the ONS port.
<code>-pwd password</code>	Specify the password for the GSMUSER.
<code>-region region_name</code>	Specify the region to which the databases belong.
<code>-savename</code>	Specify this option to store a net service name specified with the <code>-connect</code> option in the Global Data Services catalog, rather than the connect descriptor mapped to that net service name.
<code>-scan scan_address</code>	Specify the SCAN address for a cluster.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Example

Change the region of databases DB1 and DB3 to EAST.

```
GDSCCTL> modify database -database db1,db3 -region east
```

modify gdspool

Adds and removes users from the list of pool administrators for one or more database pools.

Syntax and Options

Use the `gdsctl modify gdspool` command with the following syntax:

```
modify gdspool -gdspool database_pool_list
  {-removeuser user_name | -adduser user_name}
```

Table A-32 gdsctl modify gdspool Options

Option	Description
<code>-adduser user_name</code>	Specify the user to add to the list of database pool administrators. This option grants the pool administrator role to the specified user.
<code>-gdspool database_pool_list</code>	Specify a comma-delimited list of database pool names.
<code>-removeuser user_name</code>	Specify the user to remove from the list of database pool administrators. This option revokes the pool administrator role from the specified user.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Example

Add PETER to the list of database pool administrators for the pool MYREADERFARM:

```
GDSCCTL> modify gdspool -gdspool myreaderfarm -adduser peter
```

modify gsm

Modifies the configuration parameters of the global service manager. The changes take effect after the global service manager is restarted.

Syntax and Options

Use the `gdsctl modify gsm` command with the following syntax:

```
modify gsm -gsm gsm_name [-catalog connect_id [-pwd password]]
    [-region region_name] [-localons ons_port] [-remoteons ons_port]
    [-endpoint gmsendpoint [-remote_endpoint remote_endpoint]]
    [-listener listener_port] [-wpwd wallet_password]
```

Table A–33 *gdsctl modify gsm Options*

Option	Description
<code>-catalog <i>connect_id</i></code>	Specify the connect identifier for the Global Data Services catalog database. If a network service name is specified, it must be resolvable by the local naming method to a connect descriptor that allows the global service manager being modified to connect to the catalog database.
<code>-endpoint <i>gmsendpoint</i></code>	Specify the protocol address that the global services manager listens on for client connection requests. If you use this option, the value you specify overrides the default endpoint.
<code>-gsm <i>gsm_name</i></code>	Enter the name of the global service manager that you want to modify. If you do not specify a name, then the current global service manager name for the session (specified with the <code>set gsm</code> command) is used.
<code>-listener <i>listener_port</i></code>	Specify the new listener port.
<code>-localons <i>ons_port</i></code>	Specify the new local ONS port.
<code>-pwd <i>password</i></code>	Specify the password for the <code>GSMCATUSER</code> account. If you do not specify a password, then you are prompted to enter one.
<code>-region <i>region_name</i></code>	Specify the region to which the global service manager belongs.
<code>-remote_endpoint <i>remote_endpoint</i></code>	Specify the protocol address that is used by the global service manager to receive database registration requests and communicate with other global service managers in the configuration. If you use this option, the value you specify overrides the default endpoint.
<code>-remoteons <i>ons_port</i></code>	Specify the new remote ONS port.
<code>-wpwd</code>	Specify the password for the global service manager wallet.

Usage Notes

- You must run this command locally on the computer where you want to modify the global service manager.
- This command can be run only by the operating system user who started the global service manager.

- When you run this command, GDSTL connects to the Global Data Services catalog as the GSMCATUSER user and prompts you for the GSMCATUSER password.

Example

Modify the global service manager named `gsm1` so that it is in the EAST region.

```
GDSTL> modify gsm -gsm gsm1 -region east
```

modify region

This command modifies the configuration for a region.

Syntax and Options

Use the `gdctl modify region` command with the following syntax:

```
modify region -region region_list -buddy region_name [-weights weight]
```

Table A-34 *gdctl modify region Options*

Option	Description
-buddy <i>region_name</i>	Specify the name of the buddy region
-region <i>region_list</i>	Specify a comma-delimited list of region names
-weights <i>weight</i>	Used for static RLB distribution. format: name = value,..,name = value

Usage Notes

- You must connect to the catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command.

Example

Modify two regions, EAST and WEST, as follows:

```
GDSTL> modify region -region west -buddy east
```

modify service

Modifies the service attributes. Use this command to:

- Add databases to the preferred or available lists for the service
- Move a service from one database to another database
- Change an available database to a preferred database or a preferred database to an available database
- Modify the high availability attributes of the service

If you want to temporarily move a service from one database to a different database, then use the `relocate service` command.

Syntax and Options

Use the `gdctl modify service` command to manage services.

To add more preferred or available databases to a service:

```
modify service [-gdspool gdspool_name] -service service_name
  {-preferred dbname_list | -available dbname_list}
```

To modify the high availability attributes of a service:

```
modify service [-gdspool gdspool_name] -service service_name
  [-database db_name]
  [-server_pool server_pool_name]
  [-cardinality {UNIFORM | }]
  [-locality {ANYWHERE | LOCAL_ONLY [-region_failover]}]
  [-role {PRIMARY | PHYSICAL_STANDBY [-failover_primary] | LOGICAL_STANDBY
  | SNAPSHOT_STANDBY}] [-lag {lag_value | ANY}]
  [-notification {TRUE | FALSE}]
  [-rlbgoal {SERVICE_TIME | THROUGHPUT}]
  [-dtp {TRUE | FALSE}]
  [-sql_translation_profile stp_name]
  [-clbgoal {SHORT | LONG}]
  [-tafpolicy {BASIC | NONE | PRECONNECT}]
  [-policy policy]
  [-failovertime {NONE | SESSION | SELECT}]
  [-failovermethod {NONE | BASIC}]
  [-failoverretry failover_retries]
  [-failoverdelay failover_delay]
  [-edition edition_name]
  [-commit_outcome {TRUE | FALSE}]
  [-retention retention_seconds]
  [-session_state {DYNAMIC | STATIC}]
  [-replay_init_time replay_init_time]
```

To change an available database to a preferred databases for a service:

```
modify service [-gdspool gdspool_name] -service service_name
  -available dbname_list -preferred
```

To change preferred and available status of a database for a service:

```
modify service [-gdspool gdspool_name] -service service_name
  {-preferred_all | -modifyconfig -preferred dbname_list
  [-available dbname_list]}
  [-force]
```

To modify properties for a global service that are specific to an Oracle RAC database:

```
modify service [-gdspool gdspool_name] -service service_name -database db_name
  {[-server_pool server_pool_name] [-cardinality {UNIFORM | }] |
  { -add_instances [-preferred comma-delimited-list]
  [-available comma-delimited-list] |
  -drop_instances instlist |
  -modify_instances [-preferred comma-delimited-list]
  [-available comma-delimited-list] } }
```

Table A-35 *gdsctl modify service Options*

Option	Description
<code>-add_instances [-preferred <i>comma-delimited-list</i>] [-available <i>comma-delimited-list</i>]</code>	Provides a list of preferred and available instances for the given service on the given database. The provided list over-rides existing assigned instances, if any. Using the <code>-preferred</code> and <code>-available</code> options is optional, but at least one of these must be provided.

Table A-35 (Cont.) gdsctl modify service Options

Option	Description
<code>-available <i>dbname_list</i></code>	<p>Specify a comma-delimited list of available databases on which the service runs, if the preferred databases are not available.</p> <p>The list of available instances must be mutually exclusive with the list of preferred instances.</p> <p>If you attempt to add a preferred or available database to a service that was configured with <code>-preferred_all</code>, then GDSCTL returns an error.</p>
<code>-cardinality {UNIFORM }</code>	<p>Specify the cardinality option for a service running on a policy-managed Oracle RAC database. Services with cardinality set to <code>UNIFORM</code> are offered on all database instances. Services with cardinality set to are offered on only one database instance.</p>
<code>-clbgoal {SHORT LONG}</code>	<p>For connection load balancing goal: set to <code>SHORT</code> if using run-time load balancing, set to <code>LONG</code> for long running connections such as batch jobs or older SQL*Forms style.</p> <p>The default value for this option is <code>SHORT</code>.</p>
<code>-commit_outcome {TRUE FALSE}</code>	<p>Enable Transaction Guard; when set to <code>TRUE</code>, the commit outcome for a transaction is accessible after the transaction's session fails due to a recoverable outage.</p>
<code>-database <i>db_name</i></code>	<p>Specify the name of the database on which you want to modify the service.</p> <p>When <code>-database</code> is specified, you must specify exactly one of:</p> <ul style="list-style-type: none"> ■ <code>-server_pool</code> and/or <code>-cardinality</code>. Either is optional, but at least one must appear, both can be used at once. ■ <code>-add_instances</code>, <code>-modify_instances</code>, or <code>-drop_instances</code>. Exactly one of these three options must be used.
<code>-dtp {TRUE FALSE}</code>	<p>Indicates whether Distributed Transaction Processing should be enabled for this service. This ensures that the service is offered at exactly one instance at a time for XA affinity.</p>
<code>-drop_instances <i>inst_list</i></code>	<p>Provide a list of instances to be removed from the existing assigned instances for a given service on a given database. The provided list of instances will be removed from the existing assigned list.</p>

Table A–35 (Cont.) gdsctl modify service Options

Option	Description
<code>-edition <i>edition_name</i></code>	Specify the initial session edition of the service. When an edition is specified for a service, all subsequent connections that specify the service use this edition as the initial session edition. However, if a session connection specifies a different edition, then the edition specified in the session connection is used for the initial session edition. GDSCTL does not validate the specified edition name. During connection, the connect user must have <code>USE</code> privilege on the specified edition. If the edition does not exist or if the connect user does not have <code>USE</code> privilege on the specified edition, then an error is raised.
<code>-failover_primary</code>	If you set the <code>-role</code> option to <code>PHYSICAL_STANDBY</code> , then you can use this option to enable the service for failover to the primary database.
<code>-failoverdelay <i>failover_delay</i></code>	For Application Continuity and TAF, the time delay (in seconds) between reconnect attempts for each incident at failover.
<code>-failovermethod {NONE BASIC}</code>	TAF failover method (for backward compatibility only). If <code>failovertime</code> is set to either <code>SESSION</code> or <code>SELECT</code> , then choose <code>BASIC</code> for this option.
<code>-failoverretry <i>failover_retries</i></code>	For Application Continuity and TAF, the number of attempts to connect after an incident.
<code>-failovertime {NONE SESSION SELECT TRANSACTION}</code>	Specify the failover type. To enable Application Continuity for Java, set this parameter to <code>TRANSACTION</code> . To enable Transparent Application Failover (TAF) for OCI, set this parameter to <code>SELECT</code> or <code>SESSION</code> .
<code>-force</code>	If you use this option, then all sessions are disconnected when the service is moved, requiring the sessions using the service to reconnect (potentially to a different instance). If you do not use this option, then the sessions that are connected to a database using this service stay connected, but all new sessions cannot be established to the service.
<code>-gdspool <i>gdspool_name</i></code>	Specify the name of the database pool to which the service belongs. If not specified and there is only one <code>gdspool</code> with access granted to user, it is used as the default <code>gdspool</code> .
<code>-lag {<i>lag_value</i> ANY}</code>	Specify the lag for the service in seconds. You can use the keyword <code>ANY</code> to indicate that there is no upper threshold on the lag time. The default value for the <code>-lag</code> option is <code>ANY</code> .
<code>-locality {ANYWHERE LOCAL_ONLY}</code>	The service region locality. If you do not use this option, then the default value of <code>ANYWHERE</code> is used for the service.

Table A-35 (Cont.) gdsctl modify service Options

Option	Description
-modifyconfig	Use this option to indicate that you are changing the current list of preferred and available databases for the service. If you use this option, then any databases that are not specified in either the preferred or available list, but were previously assigned, are removed from the list of databases on which the service can run.
-modify_instances [-preferred <i>comma-delimited-list</i>] [-available <i>comma-delimited-list</i>]	<p>The provided <i>comma-delimited-list</i> of preferred and available instances is merged with the existing list currently stored in the catalog.</p> <p>If you specify an instance in the <i>comma-delimited-list</i> that is not already in the stored list, it is added to the stored list in its correct mode (preferred or available.)</p> <p>If you specify in <i>comma-delimited-list</i> an instance that is already in the stored list, then the mode of the instance in the stored list is modified to the provided mode (preferred or available). If the user provided mode is the same as the stored mode, then the mode of the instance will not be changed.</p> <p>Any instances already in the stored list that are not in the provided list remain unchanged in the stored list.</p> <p>Note that an instance cannot be both preferred and available, it can be in one mode only.</p> <p>-preferred and -available are optional but at least one list must be provided.</p>
-new_db <i>database_name</i>	<p>Specify the name of the new database on which the service runs.</p> <p>If you attempt to move a service that was configured with -preferred_all, then GDSTL returns an error.</p>
-notification {TRUE FALSE}	Enable Fast Application Notification (FAN) for OCI connections.
-old_db <i>database_name</i>	<p>Specify the name of the old database on which the service runs.</p> <p>If you attempt to move a service that was configured with -preferred_all, then GDSTL returns an error.</p>
-policy {AUTOMATIC MANUAL}	<p>Specify the management policy for the service.</p> <p>If you specify AUTOMATIC (the default), then the service automatically starts when the database restarts, either by a planned restart or after a failure. Automatic restart is also subject to the service role.</p> <p>If you specify MANUAL, then the service is never automatically restarted upon planned restart of the database. A MANUAL setting does not prevent the global service manager from monitoring the service when it is running and restarting it if a failure occurs.</p>
-pdbname <i>pdb_name</i>	Specify the pluggable database name.

Table A-35 (Cont.) gdsctl modify service Options

Option	Description
<code>-preferred dbname_list</code>	Specify a comma-delimited list of preferred databases on which the service runs. When changing a database from available to preferred, you do not specify a value for the <code>-preferred</code> option. The list of preferred instances must be mutually exclusive with the list of available instances. If you attempt to add a preferred or available database to a service that was configured with <code>-preferred_all</code> , then GDSCTL returns an error.
<code>-preferred_all</code>	Specifies that all the databases in the database pool are preferred databases. Any new databases added to the pool are configured as preferred databases for this service. This option cannot be used with the <code>-preferred</code> and <code>-available</code> options.
<code>-region_failover</code>	Indicates that the service is enabled for region failover. You can only use this option when you specify <code>LOCAL_ONLY</code> for the <code>-locality</code> option.
<code>-replay_init_time replay_init_time</code>	For Application Continuity, this parameter specifies the time (in seconds) after which replay is not initiated. Default value is 300 seconds.
<code>-retention retention_seconds</code>	For Transaction Guard (<code>commit_outcome</code> set to <code>TRUE</code>), this parameter determines the amount of time (in seconds) that the commit outcome is retained in the database.
<code>-rlbgoal {SERVICE_TIME THROUGHPUT}</code>	Run-time Load Balancing Goal. Set this parameter to <code>SERVICE_TIME</code> to balance connections by response time. Set this parameter to <code>THROUGHPUT</code> to balance connections by throughput. If you do not use this option, then the value defaults to <code>SERVICE_TIME</code> for the run-time load balancing goal.
<code>-role {[PRIMARY] [PHYSICAL_STANDBY] [-failover_primary] [LOGICAL_STANDBY] [SNAPSHOT_STANDBY]}</code>	Specify the database role that the database must be for this service to start on that database. This applies only to database pools that contain an Oracle Data Guard broker configuration. See Also: <i>Oracle Data Guard Concepts and Administration</i> for more information about database roles
<code>-server_pool server_pool_name</code>	Specify the name of the Oracle RAC server pool in the GDS pool database to which the service belongs (for a policy-managed Oracle RAC database).
<code>-service service_name</code>	Specify the name of the global service.
<code>-session_state {DYNAMIC STATIC}</code>	For Application Continuity, this parameter specifies whether the session state that is not transactional is changed by the application. A value of <code>DYNAMIC</code> is recommended for most applications.

Table A-35 (Cont.) gdsctl modify service Options

Option	Description
<code>-sql_translation_profile stp_name</code>	<p>Use this option to specify a SQL translation profile for a service that you are adding after you have migrated applications from a non-Oracle database to an Oracle database.</p> <p>This option corresponds to the SQL translation profile parameter in the <code>DBMS_SERVICE</code> service attribute.</p> <p>Notes:</p> <ul style="list-style-type: none"> Before using the SQL translation feature, you must migrate all server-side application objects and data to the Oracle database. Use the <code>config service</code> command to display the SQL translation profile. <p>See Also: <i>Oracle Database Migration Guide</i> for more information about SQL translation</p>
<code>-tafpolicy {BASIC NONE }</code>	TAF policy specification (for administrator-managed databases only).

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.

Examples

Add the database DB3 as a preferred database for the service G_SALES_REPORT in the database pool MYREADERFARM.

```
GDSCTL> modify service -gdspool myreaderfarm -service g_sales_report -preferred db3
```

Modify the service G_DAILY_SALES_REPT in the database pool MYREADERFARM to change the run-time load balancing goal to THROUGHPUT.

```
GDSCTL> modify service -gdspool myreaderfarm -service g_daily_sales_rept -rlbgoal THROUGHPUT
```

Move the service G_SALES_REPORT in the database pool MYREADERFARM from the database DB1 to DB4.

```
GDSCTL> modify service -gdspool myreaderfarm -service g_sales_report -old_db db1 -new_db db4
```

Upgrade the DB3 database from an available database to a preferred database for the service G_SALES_REPORT in the database pool READFARM.

```
GDSCTL> modify service -gdspool readfarm -service g_sales_report -available db3 -preferred
```

Assume the service G_SALES_REPORT currently has the databases DB1 and DB2 assigned as preferred databases, and the database DB3 assigned as an available database. Exchange the preferred and available databases DB1 and DB3, and remove the DB2 database for the service SALES_REPORT in the database pool READFARM.

```
GDSCTL> modify service -gdspool readfarm -service g_sales_report -modifyconfig -available db3 -preferred db1
```

Modify the properties of the service G_SALES_REPORT in the database pool READFARM to specify that it should run only in the server pool named SALESPOOL for the policy-managed Oracle RAC database DB1.

```
GDSCTL> modify service -gdspool readfarm -service g_sales_report -database db1  
-server_pool salespool
```

Supply the preferred and available instances for the given service on the given database.

```
GDSCTL> modify service -gdspool mypool -service mysvc -database mydb -add_  
instances -preferred inst1,inst2 -available inst3,inst4
```

See Also: ["Modifying Global Service Attributes"](#) on page 3-6

relocate service

The `gdctl relocate service` command stops a service on one database and starts the service on a different database. Unlike using the `modify service` command to change the location of a service, this command does not change the underlying configuration. This command temporarily relocates a service to run on another database.

Note: If you attempt to use this command on a service that was previously configured with the `-preferred_all` option, then GDSTL returns an error.

Syntax and Options

Use the `gdctl relocate service` command with the following syntax:

```
relocate service [-gdspool gdspool_name] -service service_name -old_db db_name
  -new_db db_name [-force] [-override [-sourcepwd password]
  [-targetpwd password]]
```

Table A-36 *gdctl relocate service Options*

Option	Description
<code>-force</code>	If you use this option, then all sessions are disconnected when the service is moved, requiring the sessions using the service to reconnect (potentially to a different instance). If you do not use this option, then the sessions that are connected to a database using this service stay connected, but new sessions cannot be established to the service.
<code>-gdspool <i>gdspool_name</i></code>	Specify the name of the database pool where the service is located. If not specified and there is only one <i>gdspool</i> with access granted to user, it is used as the default <i>gdspool</i> .
<code>-new_db <i>db_name</i></code>	Specify the name of the database to which you want to move the service.
<code>-old_db <i>db_name</i></code>	Specify the name of the database where the service is currently located.
<code>-override</code>	This option causes the command to execute without updating the global service manager catalog. You can use this option when the catalog database is unavailable. During normal operation, you should not use this option.
<code>-service <i>service_name</i></code>	Specify the name of the service you are relocating.
<code>-sourcepwd <i>password</i></code>	Specify the password for the GSMUSER in the source database, or the database where the service is currently located.
<code>-targetpwd <i>password</i></code>	Specify the password for the GSMUSER in the database to which the service is being relocated (the target database).

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command

Example

Relocate the service SALES_REPORT in the READFARM database pool from the DB2 database to the DB3 database.

```
GDSCTL> relocate service -gds pool readfarm -service sales_report -old_db db1 -new_db db3
```

remove

The `gdsctl remove` command removes components from the global service management configuration.

Table A-37 *gdsctl remove Summary*

Command	Description
<code>remove brokerconfig</code> on page A-51	Removes the Oracle Data Guard broker configuration from a database pool
<code>remove database</code> on page A-51	Removes a database from a Global Data Services pool
<code>remove gdspool</code> on page A-52	Removes a database pool
<code>remove gsm</code> on page A-53	Removes a global service manager from the configuration
<code>remove invitednode</code> (<code>remove invitedsubnet</code>) on page A-53	Removes VNCR information from the catalog database
<code>remove region</code> on page A-54	Removes a region from the current configuration
<code>remove service</code> on page A-54	Removes a service from a database pool

remove brokerconfig

Removes an Oracle Data Guard broker configuration from a GDS pool.

Syntax and Options

Use the `gdsctl remove brokerconfig` command with the following syntax:

```
remove brokerconfig [-gdspool gdspool_name]
```

Table A-38 *gdsctl remove brokerconfig Options*

Syntax	Description
<code>-gdspool <i>gdspool_name</i></code>	Specify the GDS pool from which you want to remove the Oracle Data Guard broker configuration (not required--if not specified and there is only one GDS pool with access granted to the user, it is used by default).

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.
- If a GDS pool does not contain a Data Guard Broker configuration, an error is returned.

Example

Remove the Oracle Data Guard broker configuration from the database pool MYDGPOOL.

```
GDSCTL> remove brokerconfig -gdspool myreaderfarm
```

remove database

Removes databases from a database pool.

Syntax and Options

Use the `gdctl remove database` command with the following syntax:

```
remove database {-gdspool gdspool_name -all | [-gdspool gdspool_name]
  -database db_name_list } [-force]
```

Table A–39 *gdctl remove database Options*

Option	Description
-all	Removes all databases in the database pool.
-database <i>dbname_list</i>	Specify a comma-delimited list of database names that you want to remove from the database pool. You cannot specify a database that was added through an Oracle Data Guard broker configuration; you must use Oracle Data Guard to remove these databases.
-force	Removes the database from the catalog even if the database is not available. Using this option can result in global services not being removed from the database.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.

Example

Remove the database DB1 from the global service management configuration.

```
GDSTCL> remove database -database db1
```

remove gdspool

Removes a database pool from the current configuration.

Syntax and Options

Use the `gdctl remove gdspool` command with the following syntax:

```
remove gdspool -gdspool database_pool_list
```

Table A–40 *gdctl remove gdspool Options*

Option	Description
-gdspool <i>database_pool_list</i>	Specify a comma-delimited list of database pool names.

Usage Notes

- You must connect to the catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command.

Example

Remove the database pools tempreaders and myfarm from the Global Data Services framework.

```
GDSTCL> remove gdspool -gdspool tempreaders,myfarm
```

remove gsm

Removes a global service manager from the configuration.

Note: The removal of a global service manager requires at least one global service manager to be running to perform cleanup of Global Data Services databases. If there is only one global service manager in the Global Data Services configuration, then it has to be running to be removed.

Syntax and Options

Use the `gdctl remove gsm` command with the following syntax:

```
remove gsm [-gsm gsm_name]
```

Table A–41 *gdctl remove gsm Options*

Syntax	Description
<code>-gsm <i>gsm_name</i></code>	Specify the name of the global service manager that you want to remove. If the name is not specified, then the current global service manager is removed.

Usage Notes

- You must connect to the catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command.

Example

Remove the global service manager named `gsm5` from the configuration.

```
GDSTL> remove gsm -gsm gsm5
```

remove invitednode (remove invitedsubnet)

Remove host address information from the valid node checking for registration (VNCR) list in the Global Data Services catalog. This command removes either the specified `invitednode` or all `invitednodes` that correspond to an alias.

Syntax and Options

Use the `gdctl remove invitednode` command with the following syntax:

```
remove invitednode {[-group group_name] | vn-cr_id}
```

Table A–42 *gdctl remove invitednode Options*

Option	Description
<code>-group <i>group_name</i></code>	Specify an alias which defines a group of VNCRs. This alias can be referenced in other commands related to invited nodes.
<code><i>vn-cr_id</i></code>	The host address information, which can be an IPv4 or IPv6 address, a host name, a netmask, or other identifier for a server. The host address information cannot contain any spaces.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command

Examples

Remove the invitednode 194.66.82.32 from the catalog.

```
GDSCTL> remove invitednode 194.66.82.32
```

Remove the VNCR alias group EAST_SRV from the catalog.

```
GDSCTL> remove invitednode -group east_srv
```

remove region

Removes the specified regions from the global service management framework.

Syntax and Options

Use the `gdsctl remove region` command with the following syntax:

```
remove region -region region_list
```

Table A–43 *gdsctl remove region Options*

Option	Description
<code>-region <i>region_list</i></code>	Specify a comma-delimited list of region names

Usage Notes

- You must connect to the catalog database as a user with the Global Data Services administrator privileges, using the `connect` command before running this command.

Example

Remove the region named SOUTH from the configuration.

```
GDSCTL> remove region -region south
```

remove service

Removes a service from a database pool.

Syntax and Options

Use the `gdsctl remove service` command with the following syntax:

```
remove service [-gdspool gdspool_name] -service service_name [-force]
```

Table A–44 *gdsctl remove service Options*

Option	Description
<code>-force</code>	Indicates this action should be performed even if one or more databases in the database pool are not available.
<code>-gdspool <i>gdspool_name</i></code>	Specify the name of the database pool from which you want to remove the service. If not specified and there is only one <i>gdspool</i> with access granted to user, then it is used as the default <i>gdspool</i> .
<code>-service <i>service_name</i></code>	Specify the name of the service that you want to remove.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command

Example

Remove the service `sales_report` from the database pool `MYREADERFARM`.

```
GDCTL> remove service -gdspool myreaderfarm -service sales_report
```

See Also: ["Deleting a Global Service"](#) on page 3-7

services

The `gdsctl services` command retrieves the information about the services that are registered with the specified global service manager.

Syntax and Options

Use the `gdsctl services` command with the following syntax:

```
services [-gsm gsm_name] [-service service_name]
```

Table A–45 *services Options*

Option	Description
<code>-gsm <i>gsm_name</i></code>	Specify the name of a global service manager. If the name is not specified, then GDSCTL uses the current global service manager name for the session (specified with the <code>GDSCTL set gsm</code> command).
<code>-service <i>service_name</i></code>	Specify a fully qualified service name. If the service name is not specified, then the information about all the services registered with the global service manager is retrieved.

Usage Notes

- You must run this command on the host where the global service manager for which you want to retrieve service information resides
- You must have the privileges of the user who started the global service manager to run this command

Example

Display information about the services registered with global service manager `mygsm`:

```
GDSCTL> services -gsm mygsm
```

The `gdsctl services` command returns output similar to the following:

```
GDSCTL>services -gsm mygsm
Service "localsvc.dbpoolora.oradbcloud" has 2 instance(s). Affinity: LOCALPREF
  Instance "dbpoolora%1", name: "gdscat", db: "gdscat", region: "regionora",
  status: ready.
  Instance "dbpoolora%11", name: "gdscat2", db: "gdscat2", region: "regionora",
  status: ready.
Service "sales_report1.dbpoolora.oradbcloud" has 2 instance(s). Affinity:
LOCALONLY
  Instance "dbpoolora%1", name: "gdscat", db: "gdscat", region: "regionora",
  status: ready.
  Instance "dbpoolora%11", name: "gdscat2", db: "gdscat2", region: "regionora",
  status: ready.
Service "sales_report2.dbpoolora.oradbcloud" has 2 instance(s). Affinity: ANYWHERE
  Instance "dbpoolora%1", name: "gdscat", db: "gdscat", region: "regionora",
  status: ready.
  Instance "dbpoolora%11", name: "gdscat2", db: "gdscat2", region: "regionora",
  status: ready.
```

Note: Affinity values can be LOCALONLY when the service locality is defined as local_only, LOCALPREF when the service locality is defined as local_only with the region_failover option enabled, and ANYWHERE when the service locality is defined as anywhere.

Display the status of mthly_report service:

```
GDSCTL>services -service mthly_report.sales.oradbcloud
```

Returns output similar to the following:

```
Service "mthly_report.sales.oradbcloud" has 1 instance(s). Affinity:
ANYWHERE
    Instance "sales%1", name: "debug", db: "debug", region: "eastcoast",
status: ready.
```

set

The `gdsctl set` command is used to configure the context of the current GDSCTL session.

Table A-46 *gdsctl set Summary*

Command	Description
<code>set gsm</code> on page A-58	Sets the name of the global service manager for the current session.
<code>set inbound_connect_level</code> on page A-58	Sets the <code>INBOUND_CONNECT_LEVEL</code> listener parameter.
<code>set log_level</code> on page A-59	Sets the <code>LOG_LEVEL</code> listener parameter.
<code>set outbound_connect_level</code> on page A-59	Sets the <code>OUTBOUND_CONNECT_LEVEL</code> listener parameter.
<code>set trace_level</code> on page A-60	Sets the <code>TRACE_LEVEL</code> listener parameter.

set gsm

Sets the global service manager for the current session. This command establishes to which global service manager the successive commands apply. The specified global service manager name is resolved in the `gsm.ora` configuration file.

Syntax and Options

Use the `gdsctl set gsm` command with the following syntax:

```
set gsm -gsm gsm_name
```

Table A-47 *gdsctl set gsm Options*

Syntax	Description
<code>-gsm gsm_name</code>	Specify the name of the global service manager to work with in the current session. If you do not specify a specific global service manager, then GDSCTL uses the default global service manager name of <code>GSMORA</code> .

Usage Notes

- You must run this command on the host where the global service manager that you want to set for the current session resides.
- You must have the privileges of the user who started the global service manager to run this command.

Example

Set the global service manager for the current session to `gsm1`.

```
GDSCTL> set gsm -gsm gsm1
```

set inbound_connect_level

Sets the `INBOUND_CONNECT_LEVEL` listener parameter.

Syntax and Options

Use the `gdsctl set inbound_connect_level` command with the following syntax:

```
set inbound_connect_level [-gsm gsm_name] timeout_value
```

Table A-48 *gdctl set inbound_connect_level Options*

Option	Description
<i>-gsm gsm_name</i>	Specify the name of the global service manager that you want to start. If you do not specify a specific global service manager, then GDSCTL uses the current global service manager name for the session (specified with the <code>set gsm</code> command).
<i>timeout_value</i>	Specify in seconds the connection timeout value.

Usage Notes

- You must run this command on the host where the global service manager for which you want to set the INBOUND_CONNECT_LEVEL listener parameter resides
- You must have the privileges of the user who started the global service manager to run this command

Example

Set the INBOUND_CONNECT_LEVEL listener parameter for mygsm to time out in 60 seconds:

```
GDSCTL> set inbound_connect_level -gsm mygsm 60
```

set log_level

Sets the log level for the listener associated with a specific global service manager.

Syntax and Options

Use the `gdctl set log_level` command with the following syntax:

```
set log_level [-gsm gsm_name] log_level
```

Table A-49 *gdctl set log_level Options*

Option	Description
<i>-gsm gsm_name</i>	Specify the name of the global service manager.
<i>log_level</i>	Specify the level of detail to write to the log. Valid values are ON or OFF.

Usage Notes

- You must run this command on the host where the global service manager for which you want to set the listener log level resides
- You must have the privileges of the user who started the global service manager to run this command

Example

Set logging on for global service manager mygsm:

```
GDSCTL> set log_level -gsm mygsm ON
```

set outbound_connect_level

Sets the timeout value for the outbound connections for the listener associated with a specific global service manager.

Syntax and Options

Use the `gdsctl set outbound_connect_level` command with the following syntax:

```
set outbound_connect_level [-gsm gsm_name] timeout_value
```

Table A–50 *gdsctl set outbound_connect_level Options*

Option	Description
<code>-gsm <i>gsm_name</i></code>	Specify the name of the global service manager
<code><i>timeout_value</i></code>	Specify the connection timeout value

Usage Notes

- You must run this command on the host where the global service manager for which you want to set the timeout value of outbound connections for the listener resides.
- You must have the privileges of the user who started the global service manager to run this command.

Example

Set timeout value for all outbound connections:

```
GDSCTL> set outbound_connect_level 60
```

set trace_level

Sets the trace level for the listener associated with the specified global service manager.

Syntax and Options

Use the `gdsctl set trace_level` command with the following syntax:

```
set trace_level [-gsm gsm_name] trace_level
```

Table A–51 *gdsctl set trace_level Options*

Option	Description
<code>-gsm <i>gsm_name</i></code>	Specify the name of the global service manager. If the name is not specified, then GDSCTL uses the current global service manager name for the session (specified with the <code>GDSCTL set gsm</code> command).
<code><i>trace_level</i></code>	Specify the trace level for the global service manager listener. Valid values are USER - provides traces to identify user-induced error conditions ADMIN - provides traces to identify installation-specific problems SUPPORT - provides trace with troubleshooting information for Oracle Support Services OFF - provides no tracing

Usage Notes

- You must run this command on the host where the global service manager for which you want to set the listener trace level resides.
- You must have the privileges of the user who started the global service manager to run this command.

Example

Set the trace level for all listeners associated with `mygsm` to `ADMIN`

```
GDCTL> set trace_level -gsm mygsm ADMIN
```

start

The `gdctl start` command starts components that are part of the global service management framework.

Table A-52 *gdctl start Summary*

Command	Description
<code>start gsm</code> on page A-62	Starts the global service manager
<code>start service</code> on page A-62	Starts the specified services

start gsm

Starts a specific global service manager.

Syntax and Options

Use the `gdctl start gsm` command with the following syntax:

```
start gsm [-gsm gsm_name]
```

Table A-53 *gdctl start gsm Options*

Option	Description
<code>-gsm gsm_name</code>	Specify the name of the global service manager that you want to start. If you do not specify a specific global service manager, then GDCTL uses the current global service manager name for the session (specified with the <code>set gsm</code> command).

Usage Notes

- You must run GDCTL on the same host where the global service manager you want to start is located.
- You must have operating system privileges on the computer where you want to start the global service manager to run this command.

Example

Start the global service manager `gsm1` on the local host.

```
GDCTL> start gsm -gsm gsm1
```

start service

Starts specific services.

Syntax and Options

Use the `gdctl start service` command with the following syntax:

```
start service [-gdspool gdspool_name] -service service_name
  [{-database db_name | -override [-pwd password] -connect connect_identifier}]
```

Table A-54 *gdsctl start service Options*

Option	Description
<code>-database db_name</code>	Specify the name of the database on which you want to start the service. If you do not specify this option, then GDSCTL starts the services on all preferred databases.
<code>-connect connect_identifier</code>	Specify an Oracle Net connect descriptor or net service name that resolves to a connect descriptor.
<code>-gdspool gdspool_name</code>	Specify the name of the database pool in which the services that you want to start are located. If not specified and there is only one <i>gdspool</i> with access granted to the user, it is used as the default <i>gdspool</i> .
<code>-override</code>	This option causes the command to run without updating the global service manager catalog. You can use this option when the catalog database is unavailable. During normal operation, you should not use this option.
<code>-pwd password</code>	Specify the password of the GSMUSER in the specified database.
<code>-service service_name</code>	Specify a comma-delimited list of global service names. If you do not use this option, then GDSCTL starts all the services in the database pool.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command.

Example

Start the service SALES_REPORT, located in the READERFARM database pool.

```
GDSCTL> start service -gdspool readerfarm -service sales_report
```

See Also: ["Starting a Global Service"](#) on page 3-4

status

The `gdctl status` command displays the running status of a specific component or set of components. For example, you can use the `gdctl status` command to check whether a component is started.

Table A-55 *gdctl status Summary*

Command	Description
<code>status database</code> on page A-64	Displays the status of a specific database
<code>status gsm</code> on page A-65	Displays the status of a specific global service manager
<code>status service</code> on page A-66	Displays the status of one or more services in a specific database pool

status database

Displays the status of all databases.

Syntax and Options

Use the `gdctl status database` command with the following syntax:

```
{status database | databases} [-gsm gsm_name][-database db_name]
  [-gdspool gdspool_name]
```

Table A-56 *gdctl status database Options*

Option	Description
<code>-database <i>db_name</i></code>	Specify the name of the database on which you want to start the service. If you do not specify this option, then GDSCCTL starts the services on all preferred databases.
<code>-gdspool <i>gdspool_name</i></code>	Specify the name of the database pool in which the services you want to start are located. If not specified and there is only one <i>gdspool</i> with access granted to the user, it is used as the default <i>gdspool</i> .
<code>-gsm <i>gsm_name</i></code>	Specify the name of a global service manager to check. If the name is not specified, then GDSCCTL uses the current global service manager name for the session (specified with the GDSCCTL <code>set gsm</code> command).

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.

Example

Display the status of all databases:

```
GDSCCTL>status database
```

The `gdctl status database` command returns output similar to the following:

```
Database: "dbcat1" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1
Region: east
```

```
Service: "sales_svc2" Globally started: N Started: N
```

```

Scan: Y Enabled: Y Preferred: Y
Service: "sales_svc1" Globally started: Y Started: Y
Scan: N Enabled: Y Preferred: Y
Registered instances:
sales%11
Database: "dbcat2" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1
Region: east
Service: "sales_svc2" Globally started: N Started: N
Scan: Y Enabled: Y Preferred: Y
Service: "sales_svc1" Globally started: Y Started: Y
Scan: N Enabled: Y Preferred: Y
Registered instances:
sales%1

```

status gsm

Displays the status of a specific global service manager.

Syntax and Options

Use the `gdctl status gsm` command with the following syntax:

```
status gsm [-gsm gsm_name]
```

Table A-57 *gdctl status gsm Options*

Option	Description
<code>-gsm gsm_name</code>	Specify the name of a global service manager to check. If the name is not specified, then GDSCTL uses the current global service manager name for the session (specified with the GDSCTL <code>set gsm</code> command).

Usage Notes

- You must run GDSCTL on the same host where the global service manager for which you want to display the status is located.
- You must have operating system privileges on the computer where you want to display the global service manager status to run this command.

Example

Display status of mygsm:

```
GDSCTL> status gsm -gsm mygsm
```

The `gdctl status gsm` command returns output similar to the following:

```

Alias MYGSM
Version 12.1.0.0.2
Start Date 03-JUL-2012 16:48:54
Trace Level support
Listener Log File /u01/ORACLE/mygsm/alert/log.xml
Listener Trace File /u01/ORACLE/mygsm/trace/ora_14816_47568108067776.trc
Endpoint summary (ADDRESS=(HOST=mymv.us.hq.com) (PORT=1523) (PROTOCOL=tcp))
GSMOCI Version 0.1.8
Mastership Y
Connected to GDS catalog Y
Process Id 14818
Number of reconnections 0
Pending tasks. Total 0
Tasks in process. Total 0

```

```

Regional Mastership TRUE
Total messages published 28599
Time Zone -07:00
Orphaned Buddy Regions:
None
GDS region regionora

```

status service

Displays the status of a specific service. This command is similar to [services](#).

Syntax and Options

Use the `gdsctl status service` command with the following syntax:

```
{status service | services} [-gsm gsm_name] [-service service_name]
[{-raw|-verbose|-support}]
```

Table A-58 *gdsctl status service Options*

Option	Description
<code>-gsm <i>gsm_name</i></code>	Specify the name of a global service manager. If the name is not specified, then GDSCTL uses the current global service manager name for the session (specified with the GDSCTL <code>set gsm</code> command).
<code>-raw</code>	Used by oracle internal components.
<code>-service <i>service_name</i></code>	Specify a comma-delimited list of global service names. If you do not specify any services, then GDSCTL displays the status of all services in the database pool.
<code>-support</code>	Display more detailed information concerning load balancing.
<code>-verbose</code>	Display extra information related to load balancing.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command.

Example

Display the status of service `sales_report1.sales.oradbcloud`:

```
GDSCTL>status service -service sales_report1.sales.oradbcloud
```

The `gdsctl status service` command returns output similar to the following:

```

Service "sales_report1.sales.oradbcloud" has 3 instance(s). Affinity: ANYWHERE
  Instance "sales%1", name: "dbcat2", db: "dbcat2", region: "east",
  status: ready.
  Instance "sales%11", name: "dbcat1", db: "dbcat1", region: "west",
  status: ready.
  Instance "sales%31", name: "dbcat3", db: "dbcat3", region: "east",
  status: ready.

```

stop

The `gdctl stop` command stops components that are part of the Global Service Management framework.

Table A-59 *gdctl stop Summary*

Command	Description
<code>stop gsm</code> on page A-67	Stops a specific global service manager on the local host
<code>stop service</code> on page A-67	Stops a specific service

stop gsm

Stops a specific global service manager. You must run GDSCTL on the same host where the global service manager that you want to stop is located.

Syntax and Options

Use the `gdctl stop gsm` command with the following syntax:

```
stop gsm [-gsm gsm_name]
```

Table A-60 *gdctl stop gsm Options*

Option	Description
<code>-gsm gsm_name</code>	Specify the name of a global service manager you want to stop. If you do not specify a specific global service manager, then GDSCTL uses the current global service manager name for the session (specified with the <code>set gsm</code> command).

Usage Notes

- You must run GDSCTL on the same host where the global service manager that you want to stop is located.
- You must have operating system privileges on the computer where you want to start the global service manager to run this command.

Example

Stop the global service manager `gsm1` on the local host.

```
GDSCTL> stop gsm -gsm gsm1
```

stop service

Stops a specific service.

Syntax and Options

Use the `gdctl stop service` command with the following syntax:

```
stop service [-gdspool gdspool_name] [-service service_name]
  [{-database db_name | -override [-pwd password] -connect connect_identifier}]
  [-force]
```

Table A–61 *gdsctl stop service Options*

Option	Description
-connect <i>connect_identifier</i>	Specify an Oracle Net connect descriptor or net service name that resolves to a connect descriptor.
-database <i>db_name</i>	Specify the name of the database on which you want to stop the service. If you do not specify this option, then GDSCTL stops the services on all databases on which the service is currently running.
-force	If you use this option, then GDSCTL disconnects all sessions when the service is stopped, requiring the sessions using the service to reconnect (potentially to a different instance). If you do not use this option, then the sessions that are connected to a database using this service remain connected, but new sessions cannot be established to the service.
-gdspool <i>gdspool_name</i>	Specify the name of the database pool in which the service that you want to stop is located. If not specified and there is only one <i>gdspool</i> with access granted to user, that <i>gdspool</i> is used as the default <i>gdspool</i> .
-override	This option causes the command to execute without updating the global service manager catalog. You can use this option when the catalog database is unavailable. During normal operation, you should not use this option.
-pwd <i>password</i>	Specify the password of the GSMUSER in the specified database.
-service <i>service_name</i>	Specify a comma-delimited list of global service names you want to stop. If you do not use this option, then GDSCTL stops all the services in the database pool.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the [connect](#) command before running this command.

Example

Stop the service SALES_REPORT, on all databases in the database pool READERFARM.

```
GDSCTL> stop service -gdspool readerfarm -service sales_report
```

See Also: ["Stopping a Global Service"](#) on page 3-5

sync brokerconfig

The `gdsctl sync brokerconfig` command synchronizes the Oracle Data Guard broker configuration in the global service manager with the configuration in the database pool.

Syntax and Options

Use the `gdsctl sync brokerconfig` command with the following syntax:

```
sync brokerconfig [-gdspool gdspool_name] [-database db_name]
```

Table A-62 *gdsctl sync brokerconfig Options*

Option	Description
<code>-database <i>db_name</i></code>	Specify the name of a database in the database pool to use as a referential database , from which the configuration is queried. If you do not use this option, then GDSCTL uses the primary database as the referential database. If a primary database does not exist in the Oracle Data Guard broker configuration, then GDSCTL uses a random database from the pool as the referential database.
<code>-gdspool <i>gdspool_name</i></code>	Specify the database pool to which the Oracle Data Guard broker configuration belongs. If not specified and there is only one <i>gdspool</i> with access granted to user, that <i>gdspool</i> is used as the default <i>gdspool</i> . If the specified database pool does not contain an Oracle Data Guard broker configuration, then GDSCTL returns an error.

Usage Notes

- You must connect to the catalog database as a user with the pool administrator privileges, using the `connect` command before running this command.

Example

Synchronize the Oracle Data Guard broker configuration in the database pool MYREADERFARM with the configuration stored in the Global Data Services catalog.

```
GDSCTL> sync brokerconfig -gdspool myreaderfarm
```

sync database

The `gdsctl sync database` command brings attributes of global services and GDS related parameters of a pool database in synchronization with the contents of the GDS catalog.

Syntax and Options

Use the `gdsctl sync database` command with the following syntax:

```
sync database [-gdspool gdspool_name] [-database database_name]
```

Table A-63 *gdsctl sync database Options*

Option	Description
<code>-database <i>database_name</i></code>	Specify the name of a database in the database pool to use as a <i>referential database</i> , from which the configuration is queried.
<code>-gdspool <i>gdspool_name</i></code>	Specify the GDS pool to which the database belongs. If not specified and there is only one GDS pool with access granted to user, it is used as the default GDS pool.

Usage Notes

- If database has no GDS region assigned, an error is returned.
- If a GDS pool is specified and the database option is not specified, then each database in the pool is synchronized.

Example

Synchronize a database in the default database pool with the database `mydb`.

```
GDSCTL> sync database -database mydb
```

validate catalog

The `gdsctl validate catalog` command cross checks the Global Data Services catalog, global service manager run-time status, and pool databases, and reports inconsistencies and errors.

Because the execution of this command involves accessing all databases in a Global Data Services configuration, the GSMCATUSER password is required run it. The password is stored in the wallet of any global service manager that is part of the Global Data Services configuration. Therefore, if you run the command from the ORACLE_HOME of any of the global service managers, the password is automatically extracted from the wallet and does not have to be provided. Otherwise, you must provide the GSMCATUSER password using the `-catpwd` command option. Alternatively, if all databases in the Global Data Services configuration have the same GSMUSER password, you can specify the password instead of the GSMCATUSER password by using the `-dbpwd` option.

Syntax and Options

Use the `gdsctl validate catalog` command with the following syntax:

```
validate [catalog] [-gsm gsm_name] [ {-config | -database db_name} ]
[-catpwd cpwd] [-dbpwd dpwd]
```

Table A-64 *gdsctl validate catalog Options*

Option	Description
<code>-catpwd cpwd</code>	Provides the GDSCATUSER password, otherwise it is read from the local wallet file by default.
<code>-config</code>	Indicates that the validation should be performed on the Global Data Services catalog configuration only.
<code>-database db_name</code>	Indicates the name of the database for which the cross-check validation should be performed.
<code>-dbpwd dpwd</code>	Provides the pool database password directly if there is only one database in the pool, or if multiple databases in the pool share the same password.
<code>-gsm gsm_name</code>	Specify the global service manager name. If the name is not specified, then GDSCTL uses the current global service manager name for the session (specified with the <code>set gsm</code> command).

Example

Validate the catalog:

```
GDSCTL> validate
```

The output should be similar to the following:

```
Validation results:
VLD2: Region "regionora" does not have buddy region
VLD11: GDS pool "marketing" does not contain any databases
VLD12: GDS pool "marketing" does not contain any global services
VLD11: GDS pool "sales" does not contain any databases
VLD12: GDS pool "sales" does not contain any global services
VLD11: GDS pool "mkt" does not contain any databases
VLD12: GDS pool "mkt" does not contain any global services
```

Glossary

catalog database

The Oracle Database 12c database in which the Global Data Services catalog resides.

endpoint

The address or connection point to a global service manager or listener.

GDSCTL

Global Data Services command-line interface.

Global Data Services catalog

A repository that holds configuration and run-time status of a Global Data Services configuration, including data on global services, their attributes, and all logical and physical components of the configuration, such as Global Data Services pools, Global Data Services regions, global service managers, and database instances. The catalog may also contain data on replication and network topologies related to the configuration.

Global Data Services configuration

A set of databases that are integrated by the Global Data Services framework into a single virtual server that offers one or more global services, while ensuring high performance, availability, and optimal utilization of resources.

Global Data Services pool

A set of databases within a GDS configuration that provides a unique set of global services and belongs to a certain administrative domain.

Global Data Services region

A logical boundary that contains database clients and servers that are considered to be in proximity to each other.

global service

A database service that can be provided by multiple databases synchronized through data replication.

global service manager

A software component that provides service-level load balancing and centralized management of services within the Global Data Services configuration.

global service

A service that is offered on only one database of a Global Data Services pool at a time.

Oracle Notification Service (ONS)

A publish and subscribe service for communicating information about all FAN events.

valid node checking for registration list

See "[VNCR](#)" on page Glossary-2.

VNCR

Valid node checking for registration. Allows or denies access from specified IP addresses to Oracle Global Data Services pool.

A

affinity

- any-region, 1-12
- local region, 1-12
- local region with failover, 1-13
- any-region affinity, 1-12

B

broker configuration, 1-7

brokerconfig

- GDSCTL object name, A-8

C

catalog

- GDSCTL object name, A-8

D

database

- GDSCTL object name, A-8
- database cardinality, 1-8

G

GDSCTL, A-1

- command syntax, A-7
- commands
 - add, A-7
 - add brokerconfig, A-10
 - add database, A-10, A-11
 - add gdspool, A-10, A-12
 - add gsm, A-4, A-6, A-10, A-13
 - add invitednode (add invitedsubnet), A-10, A-12, A-13, A-15
 - add region, A-10, A-16
 - add service, A-10, A-16
 - config, A-7
 - config database, A-22
 - config gdspool, A-23
 - config gsm, A-24
 - config region, A-24
 - config service, A-20, A-25
 - config vncr, A-26
 - configure, A-7

- connect, A-7, A-12, A-13, A-15, A-16, A-29
 - create catalog, A-7, A-30
 - delete catalog, A-7, A-32
 - disable service, A-7, A-33
 - enable service, A-7, A-34
 - modify, A-7
 - modify catalog, A-37
 - modify database, A-38
 - modify gdspool, A-39
 - modify gsm, A-6, A-40
 - modify region, A-41
 - modify service, A-17, A-19, A-41
 - relocate service, A-7, A-49
 - remove, A-8
 - remove brokerconfig, A-51
 - remove database, A-52
 - remove gdspool, A-52
 - remove gsm, A-53
 - remove region, A-54
 - remove service, A-54
 - remove vncr, A-53
 - services, A-8, A-56
 - set, A-8
 - set gsm, A-14, A-58
 - set inbound_connect_level, A-6, A-58
 - set log_level, A-6, A-59
 - set outbound_connect_level, A-6, A-60
 - set trace_level, A-6, A-60
 - start, A-8
 - start gsm, A-6, A-62
 - start service, A-62
 - status, A-8
 - status database, A-64
 - status gsm, A-6, A-65
 - status service, A-66
 - stop, A-8
 - stop gsm, A-6, A-67
 - stop service, A-67
 - sync brokerconfig, A-8, A-69
 - validate catalog, A-8, A-71
- ### commnds
- add, A-10
- ### config
- usage description, A-22
- ### connect
- usage description, A-29

- help, A-4
- keywords, A-8
- modify, usage description, A-37
- object name
 - brokerconfig, A-8
 - catalog, A-8
 - database, A-8
 - gds pool, A-9
 - gsm, A-9
 - region, A-9
 - service, A-9
 - VNCR, A-8
- object names, A-8
- remove, usage description, A-51
- running commands in a batch file, A-4
- set, usage description, A-58
- start, usage description, A-62
- status, usage description, A-64
- stop, usage description, A-67
- usage information, A-4
- gdsctl
 - scripting commands, A-4
- gdsctl sync database, A-70
- gds pool
 - GDSCTL object name, A-9
- Global Data Services
 - administrator, 2-2
 - catalog
 - creating, 2-5, 2-6
 - requirements, 2-5
 - catalog database
 - connecting to, 2-7
 - logical components
 - database pool, 1-3
 - region, 1-4
 - physical components
 - catalog, 1-5
 - global service manager, 1-4
 - Oracle Notification Service server, 1-5
 - pool
 - adding, 2-8
 - region
 - adding, 2-9
- global run-time connection load balancing, 1-13
- global service manager
 - installing, 2-3
- global services
 - attributes, 1-6
 - in an Oracle Data Guard broker
 - configuration, 1-7
 - in an Oracle RAC database, 1-7
 - overview, 1-5
- gsm
 - GDSCTL object name, A-9

L

- load balancing
 - server-side, 1-12
- local region affinity, 1-12

- local region affinity with failover, 1-13
- local services, 1-5

O

- objects
 - GDSCTL object names and abbreviations, A-8
- Oracle Data Guard, 1-7

P

- preferred databases, 1-8

R

- region
 - GDSCTL object name, A-9
- region affinity, 1-12
 - any, 1-12
 - local, 1-12
 - local with failover, 1-13
- replication lag time, 1-9
- run-time connection load balancing, 1-13
 - service-level goals, 1-13
 - SERVICE_TIME, 1-13
 - THROUGHPUT, 1-13

S

- server-side load balancing, 1-12
- service
 - GDSCTL object name, A-9
- services
 - managing, A-41
- SYS_CONTEXT, 4-5

V

- valid node checking for registration
 - See* VNCR
- VNCR
 - GDSCTL object name, A-8