2

Configuring Automatic Restart of an Oracle Database

Configure your Oracle database with the Oracle Restart feature to automatically restart the database, the listener, and other Oracle components after a hardware or software failure or whenever your database host computer restarts.

- About Oracle Restart
  Oracle Restart enhances the availability of Oracle databases in a single-instance environment.
- Configuring Oracle Restart
  To configure Oracle Restart, you can add components, remove components, or modify options for components.
- Starting and Stopping Components Managed by Oracle Restart
  When Oracle Restart is in use, Oracle strongly recommends that you use the SRVCTL utility to start and stop components.
- Stopping and Restarting Oracle Restart for Maintenance Operations
  When several components in an Oracle home are managed by Oracle Restart, you can stop Oracle Restart and the components managed by Oracle Restart in the Oracle home.
- SRVCTL Command Reference for Oracle Restart
  You can reference details about the syntax and options for SRVCTL commands specific to Oracle Restart.
- CRSCTL Command Reference
  You can reference details about the syntax for the CRSCTL commands that are relevant for Oracle Restart.

2.1 About Oracle Restart

Oracle Restart enhances the availability of Oracle databases in a single-instance environment.

- Oracle Restart Overview
  When you install Oracle Restart, various Oracle components can be automatically restarted after a hardware or software failure or whenever your database host computer restarts.
- About Startup Dependencies
  Oracle Restart ensures that Oracle components are started in the proper order, in accordance with component dependencies.
- About Starting and Stopping Components with Oracle Restart
  Oracle Restart automatically restarts various Oracle components when required, and automatically stops Oracle components in an orderly fashion when you manually shut down your system.
- About Starting and Stopping Oracle Restart
  The CRSCTL utility starts and stops Oracle Restart.

- Oracle Restart Configuration
  Oracle Restart maintains a list of all the Oracle components that it manages, and maintains configuration information for each component.
- Oracle Restart Integration with Oracle Data Guard
  Oracle Restart is integrated with Oracle Data Guard (Data Guard) and the Oracle Data Guard Broker (the broker).
- Fast Application Notification with Oracle Restart
  Oracle Restart uses Oracle Notification Services (ONS) and Oracle Advanced Queues to publish Fast Application Notification (FAN) high availability events. Integrated Oracle clients use FAN to provide fast notification to clients when the service or instance goes down. The client can automate the failover of database connections between a primary database and a standby database.

2.1.1 Oracle Restart Overview

When you install Oracle Restart, various Oracle components can be automatically restarted after a hardware or software failure or whenever your database host computer restarts.

Table 2-1 lists these components.

Table 2-1 Oracle Components Automatically Restarted by Oracle Restart

Component	Notes
Database instance	Oracle Restart can accommodate multiple databases on a single host computer.
Oracle Net listener	-
Database services	Does not include the default service created upon installation because it is automatically managed by Oracle Database. Also does not include any default services created during database creation or global services. For more information about global services, see the *Oracle Database Global Data Services Concepts and Administration Guide*.
Oracle Automatic Storage Management (Oracle ASM) instance	-
Oracle ASM disk groups	Restarting a disk group means mounting it.
Oracle Notification Services (ONS)	In an Oracle Grid Infrastructure for Standalone Servers (Oracle Restart) environment, ONS can be used in Oracle Data Guard installations for automating failover of connections between primary and standby database through Fast Application Notification (FAN). ONS is a service for sending FAN events to integrated clients upon failover.

Oracle Restart runs periodic check operations to monitor the health of these components. If a check operation fails for a component, the component is shut down and restarted.

Oracle Restart is used in standalone server (non-clustered) environments only. For Oracle Real Application Clusters (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware.

Oracle Restart runs out of the Oracle Grid Infrastructure home, which you install separately from Oracle Database homes. See the *Oracle Grid Infrastructure Installation Guide* for your platform for information about installing the Oracle Grid Infrastructure home.

See Also:

- "Configuring Oracle Restart"
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle Automatic Storage Management

2.1.2 About Startup Dependencies

Oracle Restart ensures that Oracle components are started in the proper order, in accordance with component dependencies.

For example, if database files are stored in Oracle ASM disk groups, then before starting the database instance, Oracle Restart ensures that the Oracle ASM instance is started and the required disk groups are mounted. Likewise, if a component must be shut down, Oracle Restart ensures that dependent components are cleanly shut down first.

Oracle Restart also manages the weak dependency between database instances and the Oracle Net listener (the listener): When a database instance is started, Oracle Restart attempts to start the listener. If the listener startup fails, then the database is still started. If the listener later fails, Oracle Restart does not shut down and restart any database instances.

2.1.3 About Starting and Stopping Components with Oracle Restart

Oracle Restart automatically restarts various Oracle components when required, and automatically stops Oracle components in an orderly fashion when you manually shut down your system.

There may be times, however, when you want to manually start or stop individual Oracle components. Oracle Restart includes the Server Control (SRVCTL) utility that you use to manually start and stop Oracle Restart–managed components. When Oracle Restart is in use, Oracle strongly recommends that you use SRVCTL to manually start and stop components.

After you stop a component with SRVCTL, Oracle Restart does not automatically restart that component if a failure occurs. If you then start the component with SRVCTL, that component is again available for automatic restart.

Oracle utilities such as SQL*Plus, the Listener Control utility (`LSNRCTL`), and `ASMCMD` are integrated with Oracle Restart. If you shut down the database with SQL*Plus, Oracle Restart does not interpret this as a database failure and does not attempt to restart the database. Similarly, if you shut down the Oracle ASM instance with SQL*Plus or `ASMCMD`, Oracle Restart does not attempt to restart it.

An important difference between starting a component with SRVCTL and starting it with SQL*Plus (or another utility) is the following:

- When you start a component with SRVCTL, any components on which this component depends are automatically started first, and in the proper order.
- When you start a component with SQL*Plus (or another utility), other components in the dependency chain are not automatically started; you must ensure that any components on which this component depends are started.

In addition, Oracle Restart enables you to start and stop all of the components managed by Oracle Restart in a specified Oracle home using a single command. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home. This capability is useful when you are installing a patch.

See Also:

"Starting and Stopping Components Managed by Oracle Restart"

2.1.4 About Starting and Stopping Oracle Restart

The CRSCTL utility starts and stops Oracle Restart.

You can also use the CRSCTL utility to enable or disable Oracle high availability services. Oracle Restart uses Oracle high availability services to start and stop automatically the components managed by Oracle Restart. For example, Oracle high availability services daemons automatically start databases, listeners, and Oracle ASM instances. When Oracle high availability services are disabled, none of the components managed by Oracle Restart are started when a node is rebooted.

Typically, you use the CRSCTL utility when you must stop all of the running Oracle software in an Oracle installation. For example, you might need to stop Oracle Restart when you are installing a patch or performing operating system maintenance. When the maintenance is complete, you use the CRSCTL utility to start Oracle Restart.

See Also:

"Stopping and Restarting Oracle Restart for Maintenance Operations" for information about using the CRSCTL utility

2.1.5 Oracle Restart Configuration

Oracle Restart maintains a list of all the Oracle components that it manages, and maintains configuration information for each component.

All of this information is collectively known as the **Oracle Restart configuration**. When Oracle Restart starts a component, it starts the component according to the configuration information for that component. For example, the Oracle Restart configuration includes the location of the server parameter file (SPFILE) for databases, and the TCP port to listen on for listeners.

If you install Oracle Restart and then create your database with Database Configuration Assistant (DBCA), DBCA automatically adds the database to the Oracle Restart configuration. When DBCA then starts the database, the required dependencies between the database and other components (for example disk groups in which the database stores data) are established, and Oracle Restart begins to manage the database.

You can manually add and remove components from the Oracle Restart configuration with SRVCTL commands. For example, if you install Oracle Restart onto a host on which a database is already running, you can use SRVCTL to add that database to the Oracle Restart configuration. When you manually add a component to the Oracle Restart configuration and then start it with SRVCTL, Oracle Restart begins to manage the component, restarting it when required.

Note:

Adding a component to the Oracle Restart configuration is also referred to as "registering a component with Oracle Restart."

Other SRVCTL commands enable you to view the status and configuration of Oracle Restart–managed components, temporarily disable and then reenable management for components, and more.

Note:

Starting with Oracle Database 19c, customer use of the `SERVICE_NAMES` parameter is deprecated. It can be desupported in a future release. To manage your services, Oracle recommends that you use the `SRVCTL` or `GDSCTL` command line utilities, or the `DBMS_SERVICE` package.

When Oracle Restart is installed, many operations that create Oracle components automatically add the components to the Oracle Restart configuration. Table 2-2 lists some create operations and whether the created component is automatically added.

Table 2-2 Create Operations and the Oracle Restart Configuration

Create Operation	Created Component Automatically Added to Oracle Restart Configuration?
Create a database with OUI or DBCA	Yes
Create a database with the `CREATE DATABASE` SQL statement	No
Create an Oracle ASM instance with OUI, DBCA, or ASMCA	Yes
Create a disk group (any method)	Yes
Add a listener with NETCA	Yes
Create a database service with SRVCTL	Yes
Create a database service by modifying the `SERVICE_NAMES` initialization parameter[1]	No
Create a database service with `DBMS_SERVICE.CREATE_SERVICE`	No
Create a standby database	No

[1] Not recommended when Oracle Restart is in use

Table 2-3 lists some delete/drop/remove operations and whether the deleted component is also automatically removed from the Oracle Restart configuration.

Table 2-3 Delete/Drop/Remove Operations and the Oracle Restart Configuration

Operation	Deleted Component Automatically Removed from Oracle Restart Configuration?
Delete a database with DBCA	Yes
Delete a database by removing database files with operating system commands[1]	No
Delete a listener with NETCA	Yes
Drop an Oracle ASM disk group (any method)	Yes
Delete a database service with SRVCTL	Yes
Delete a database service by any other means	No

[1] Not recommended

2.1.6 Oracle Restart Integration with Oracle Data Guard

Oracle Restart is integrated with Oracle Data Guard (Data Guard) and the Oracle Data Guard Broker (the broker).

When a database shutdown and restart is required in response to a role change request, Oracle Restart shuts down and restarts the database in an orderly fashion (taking dependencies into account), and according to the settings in the Oracle Restart configuration. Oracle Restart also ensures that, following a Data Guard role transition, all database services configured to run in the new database role are active and all services not configured to run in the new role are stopped.

In addition, the Oracle Restart configuration supports Data Guard–related configuration options for the following components:

- **Databases**—When you add a database to the Oracle Restart configuration, you can specify the current Data Guard role for the database: `PRIMARY`, `PHYSICAL_STANDBY`, `LOGICAL_STANDBY`, or `SNAPSHOT_STANDBY`. If the role is later changed using the broker, Oracle Restart automatically updates the database configuration with the new role. If you change the database role without using the broker, you must manually modify the database's role in the Oracle Restart configuration to reflect the new role.
- **Database Services**—When adding a database service to the Oracle Restart configuration, you can specify one or more Data Guard roles for the service. When this configuration option is present, upon database open Oracle Restart starts the service only if one of the service roles matches the current database role.

> **See Also:**
>
> - *Oracle Data Guard Concepts and Administration* for information about Oracle Data Guard
> - "Fast Application Notification with Oracle Restart"
> - "Automating the Failover of Connections Between Primary and Standby Databases"

2.1.7 Fast Application Notification with Oracle Restart

Oracle Restart uses Oracle Notification Services (ONS) and Oracle Advanced Queues to publish Fast Application Notification (FAN) high availability events. Integrated Oracle clients use FAN to provide fast notification to clients when the service or instance goes down. The client can automate the failover of database connections between a primary database and a standby database.

- Overview of Fast Application Notification
  FAN is a high availability notification mechanism that Oracle Restart can use to notify other processes about configuration changes that include service status changes, such as `UP` or `DOWN` events.
- Application High Availability with Services and FAN
  Oracle Database focuses on maintaining service availability. With Oracle Restart, Oracle services are designed to be continuously available. Oracle Restart monitors the database and its services and, when configured, sends event notifications using FAN.

> **See Also:**
>
> *Oracle Database Advanced Queuing User's Guide*

2.1.7.1 Overview of Fast Application Notification

FAN is a high availability notification mechanism that Oracle Restart can use to notify other processes about configuration changes that include service status changes, such as `UP` or `DOWN` events.

FAN provides the ability to immediately terminate inflight transaction when an instance or server fails. Integrated Oracle clients receive the events and respond. Applications can respond either by propagating the error to the user or by resubmitting the transactions and masking the error from the application user. When a `DOWN` event occurs, integrated clients immediately clean up connections to the terminated database. When an `UP` event occurs, the clients create new connections to the new primary database instance.

Oracle Restart publishes FAN events whenever a managed instance or service goes up or down. After a failover, the Oracle Data Guard Broker (broker) publishes FAN events. These FAN events can be used in the following ways:

- Applications can use FAN with Oracle Restart without programmatic changes if they use one of these Oracle integrated database clients: Oracle Database JDBC, Universal

Connection Pool for Java, Oracle Call Interface, and Oracle Database ODP.NET. These clients can be configured for Fast Connection Failover (FCF) to automatically connect to a new primary database after a failover.

- FAN server-side callouts can be configured on the database tier.

For `DOWN` events, such as a failed primary database, FAN provides immediate notification to the clients so that they can failover as fast as possible to the new primary database. The clients do not wait for a timeout. The clients are notified immediately, and they must be configured to failover when they are notified.

For `UP` events, when services and instances are started, new connections can be created so that the application can immediately take advantage of the extra resources.

Through server-side callouts, you can also use FAN to:

- Log status information
- Page DBAs or open support tickets when resources fail to start
- Automatically start dependent external applications that must be co-located with a service

FAN events are published using ONS and Oracle Database Advanced Queuing queues. The queues are configured automatically when you configure a service. You must configure ONS manually using SRVCTL commands.

The Connection Manager (CMAN) and Oracle Net Services listeners are integrated with FAN events, enabling the CMAN and the listener to immediately de-register services provided by the failed instance and to avoid erroneously sending connection requests to a failed database.

See Also:

Oracle Data Guard Broker for information about FAN events in an Oracle Data Guard environment

2.1.7.2 Application High Availability with Services and FAN

Oracle Database focuses on maintaining service availability. With Oracle Restart, Oracle services are designed to be continuously available. Oracle Restart monitors the database and its services and, when configured, sends event notifications using FAN.

- Managing Unplanned Outages
If Oracle Restart detects an outage, then it isolates the failed component and recovers the dependent components. If the failed component is the database instance, then after Oracle Data Guard fails over to the standby database, Oracle Restart on the new primary database starts any services defined with the current role.
- Managing Planned Outages
For repairs, upgrades, and changes that require you to shut down the primary database, Oracle Restart provides interfaces that disable and enable services to minimize service disruption to application users.
- Fast Application Notification High Availability Events
Understand FAN event record parameters and the event types.
- Using Fast Application Notification Callouts
FAN callouts are server-side executables that Oracle Restart executes immediately when high availability events occur.

- Oracle Clients That Are Integrated with Fast Application Notification
  Oracle has integrated FAN with many of the common Oracle client drivers that are used to connect to Oracle Restart databases. Therefore, the easiest way to use FAN is to use an integrated Oracle Client.

2.1.7.2.1 Managing Unplanned Outages

If Oracle Restart detects an outage, then it isolates the failed component and recovers the dependent components. If the failed component is the database instance, then after Oracle Data Guard fails over to the standby database, Oracle Restart on the new primary database starts any services defined with the current role.

FAN events are published by Oracle Restart and the Oracle Data Guard Broker through ONS and Advanced Queuing. You can also perform notifications using FAN callouts.

> **Note:**
>
> Oracle Restart does not run callouts with guaranteed ordering. Callouts are run asynchronously, and they are subject to scheduling variability.

With Oracle Restart, restart and recovery are automatic, including the restarting of the subsystems, such as the listener and the Oracle Automatic Storage Management (Oracle ASM) processes, not just the database. You can use FAN callouts to report faults to your fault management system and to initiate repair jobs.

2.1.7.2.2 Managing Planned Outages

For repairs, upgrades, and changes that require you to shut down the primary database, Oracle Restart provides interfaces that disable and enable services to minimize service disruption to application users.

Using Oracle Data Guard Broker with Oracle Restart allows a coordinated failover of the database service from the primary to the standby for the duration of the planned outage. Once you complete the operation, you can return the service to normal operation.

The management policy for a service controls whether the service starts automatically when the database is restarted. If the management policy for a service is set to `AUTOMATIC`, then it restarts automatically. If the management policy for a service is set to `MANUAL`, then it must be started manually.

> **See Also:**
>
> "Modifying the Oracle Restart Configuration for a Component"

2.1.7.2.3 Fast Application Notification High Availability Events

Understand FAN event record parameters and the event types.

Table 2-4 describes the FAN event record parameters and the event types, followed by name-value pairs for the event properties. The event type is always the first entry and the timestamp is always the last entry. In the following example, the name in the name-value pair is shown in

Fan event type (service_member), and the value in the name-value pair is shown in Properties:

```
FAN event type: service_member
Properties: version=1.0 service=ERP database=FINPROD instance=FINPROD host=node1
status=up
```

Table 2-4 Event Record Parameters and Descriptions

Parameter	Description
VERSION	Version of the event record. Used to identify release changes.
EVENT TYPE	SERVICE, SERVICE_MEMBER, DATABASE, INSTANCE, NODE, ASM, SRV_PRECONNECT. Note that database and Instance types provide the database service, such as DB_UNIQUE_NAME.DB_DOMAIN.
DATABASE UNIQUE NAME	The unique database supporting the service; matches the initialization parameter value for DB_UNIQUE_NAME, which defaults to the value of the initialization parameter DB_NAME.
INSTANCE	The name of the instance that supports the service; matches the ORACLE_SID value.
NODE NAME	The name of the node that supports the service or the node that has stopped; matches the node name known to Cluster Synchronization Services (CSS).
SERVICE	The service name; matches the service in DBA_SERVICES.
STATUS	Values are UP, DOWN, NOT_RESTARTING, PRECONN_UP, PRECONN_DOWN, and UNKNOWN.
REASON	Data_Guard_Failover, Failure, Dependency, User, Autostart, Restart.
CARDINALITY	The number of service members that are currently active; included in all UP events.
TIMESTAMP	The local time zone to use when ordering notification events.

A FAN record matches the database signature of each session as shown in Table 2-5.

Table 2-5 FAN Parameters and Matching Database Signatures

FAN Parameter	Matching Oracle Database Signature
SERVICE	sys_context('userenv', 'service_name')
DATABASE UNIQUE NAME	sys_context('userenv', 'db_unique_name')
INSTANCE	sys_context('userenv', 'instance_name')
NODE NAME	sys_context('userenv', 'server_host')

2.1.7.2.4 Using Fast Application Notification Callouts

FAN callouts are server-side executables that Oracle Restart executes immediately when high availability events occur.

You can use FAN callouts to automate the following activities when events occur, such as:

- Opening fault tracking tickets
- Sending messages to pagers

- Sending e-mail
- Starting and stopping server-side applications
- Maintaining an uptime log by logging each event as it occurs

To use FAN callouts:

- Place an executable in the directory grid_home/racg/usrco on both the primary and the standby database servers. If you are using scripts, then set the shell as the first line of the executable.

The following is an example file for the grid_home/racg/usrco/callout.sh callout:

```
#! /bin/ksh
FAN_LOGFILE= [your path name]/admin/log/`hostname`_uptime.log
echo $* "reported="`date` >> $FAN_LOGFILE &
```

The following output is from the previous example:

```
NODE VERSION=1.0 host=sun880-2 status=nodedown reason=
timestamp=08-Oct-2004 04:02:14 reported=Fri Oct 8 04:02:14 PDT 2004
```

A FAN record matches the database signature of each session, as shown in Table 2-5. Use this information to take actions on sessions that match the FAN event data.

See Also:

Table 2-4 for information about the callout and event details

2.1.7.2.5 Oracle Clients That Are Integrated with Fast Application Notification

Oracle has integrated FAN with many of the common Oracle client drivers that are used to connect to Oracle Restart databases. Therefore, the easiest way to use FAN is to use an integrated Oracle Client.

You can use the CMAN session pools, Oracle Call Interface, Universal Connection Pool for Java, JDBC simplefan API, and ODP.NET connection pools. The overall goal is to enable applications to consistently obtain connections to the available primary database at anytime.

See Also:

"Automating the Failover of Connections Between Primary and Standby Databases"

2.2 Configuring Oracle Restart

To configure Oracle Restart, you can add components, remove components, or modify options for components.

- About Configuring Oracle Restart
  If you install Oracle Restart by installing the Oracle Grid Infrastructure for Standalone Servers and then create your database, the database is automatically added to the Oracle

Restart configuration, and is then automatically restarted when required. However, if you install Oracle Restart on a host computer on which a database already exists, you must manually add the database, the listener, the Oracle Automatic Storage Management (Oracle ASM) instance, and possibly other components to the Oracle Restart configuration.

- Preparing to Run SRVCTL
  Many Oracle Restart tasks require that you run the SRVCTL utility. You must ensure that you run SRVCTL from the correct Oracle home, and that you log in to the host computer with the correct user account.
- Obtaining Help for SRVCTL
  Online help is available for the SRVCTL utility.
- Adding Components to the Oracle Restart Configuration
  In most cases, creating an Oracle component on a host that is running Oracle Restart automatically adds the component to the Oracle Restart configuration. However, in some cases, you must add components manually.
- Removing Components from the Oracle Restart Configuration
  When you use an Oracle-recommended method to delete an Oracle component, the component is also automatically removed from the Oracle Restart configuration.
- Disabling and Enabling Oracle Restart Management for a Component
  You can temporarily disable Oracle Restart management for a component. One reason to do this is when you are performing maintenance on the component. For example, if a component must be repaired, then you might not want it to be automatically restarted if it fails or if the host computer is restarted. When maintenance is complete, you can reenable management for the component.
- Viewing Component Status
  You can use SRVCTL to view the running status (running or not running) for any component managed by Oracle Restart. For some components, additional information is also displayed.
- Viewing the Oracle Restart Configuration for a Component
  You can use SRVCTL to view the Oracle Restart configuration for any component. Oracle Restart maintains different configuration information for each component type. In one form of the SRVCTL command, you can obtain a list of components managed by Oracle Restart.
- Modifying the Oracle Restart Configuration for a Component
  You can use SRVCTL to modify the Oracle Restart configuration of a component. For example, you can modify the port number that a listener listens on when Oracle Restart starts it, or the server parameter file (SPFILE) that Oracle Restart points to when it starts a database.
- Managing Environment Variables in the Oracle Restart Configuration
  The Oracle Restart configuration can store name/value pairs for environment variables.
- Creating and Deleting Database Services with SRVCTL
  When managing a database with Oracle Restart, Oracle recommends that you use SRVCTL to create and delete database services. When you use SRVCTL to add a database service, the service is automatically added to the Oracle Restart configuration and a dependency between the service and the database is established. Thus, if you start the service, Oracle Restart first starts the database if it is not started.
- Enabling FAN Events in an Oracle Restart Environment
  To enable Oracle Restart to publish Fast Application Notification (FAN) events, you must create an Oracle Notification Services (ONS) network that includes the Oracle Restart servers and the integrated clients.

- Automating the Failover of Connections Between Primary and Standby Databases
  In a configuration that uses Oracle Restart and Oracle Data Guard primary and standby databases, the database services fail over automatically from the primary to the standby during either a switchover or failover.
- Enabling Clients for Fast Connection Failover
  Fast Connection Failover provides high availability to Fast Application Notification (FAN) integrated clients, such as clients that use JDBC, OCI, or ODP.NET. If you configure the client to use fast connection failover, then the client automatically subscribes to FAN events and can react to database UP and DOWN events. In response, Oracle Database gives the client a connection to an active instance that provides the requested database service.

2.2.1 About Configuring Oracle Restart

If you install Oracle Restart by installing the Oracle Grid Infrastructure for Standalone Servers and then create your database, the database is automatically added to the Oracle Restart configuration, and is then automatically restarted when required. However, if you install Oracle Restart on a host computer on which a database already exists, you must manually add the database, the listener, the Oracle Automatic Storage Management (Oracle ASM) instance, and possibly other components to the Oracle Restart configuration.

After configuring Oracle Restart to manage your database, you may want to:

- Add additional components to the Oracle Restart configuration.
- Remove components from the Oracle Restart configuration.
- Temporarily suspend Oracle Restart management for one or more components.
- Modify the Oracle Restart configuration options for an individual component.

> **See Also:**
>
> "About Oracle Restart"

2.2.2 Preparing to Run SRVCTL

Many Oracle Restart tasks require that you run the SRVCTL utility. You must ensure that you run SRVCTL from the correct Oracle home, and that you log in to the host computer with the correct user account.

Table 2-6 lists the components that you can configure with SRVCTL, and for each component, lists the Oracle home from which you must run SRVCTL.

Table 2-6 Determining the Oracle Home from which to Start SRVCTL

Component Being Configured	Oracle Home from which to Start SRVCTL
Database, database service	Database home
Oracle ASM instance, disk group, listener[1], ONS	Oracle Grid Infrastructure home

[1] Assumes the listener was started from the Oracle Grid Infrastructure home. If you installed Oracle Restart for an existing database, the listener may have been started from the database home, in which case you start SRVCTL from the database home.

To prepare to run SRVCTL:

1. Use Table 2-6 to determine the Oracle home from which you must run SRVCTL.
2. If you intend to run a SRVCTL command that modifies the Oracle Restart configuration (`add`, `remove`, `enable`, `disable`, and so on), then do one of the following:
   - On UNIX and Linux, log in to the database host computer as the user who installed the Oracle home that you determined in Step 1.
   - On Windows, log in to the database host computer as an Administrator.

   Otherwise, log in to the host computer as any user.
3. Open the command window that you will use to enter the SRVCTL commands.

   To enter commands, you might need to ensure that the SRVCTL program is in your `PATH` environment variable. Otherwise, you can enter the absolute path to the program.

2.2.3 Obtaining Help for SRVCTL

Online help is available for the SRVCTL utility.

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl
   ```

For more detailed help, enter the following command:

```
srvctl -help
```

For detailed help on a particular command, enter:

```
srvctl command -help
```

For example, to obtain help for the `add` command and the different options for each component type, enter:

```
srvctl add -help
```

For detailed help on a particular command for a particular component type, enter:

```
srvctl command object -help
```

For example, to obtain help about adding a database service, enter the following command:

```
srvctl add service -help
```

See "SRVCTL Command Reference for Oracle Restart" for a list of SRVCTL commands and Table 2-7 for a list of components.

Starting with Oracle Database 12*c*, single-letter parameters are deprecated in favor of keyword parameters. To support backward compatibility, you can use a mix of single-letter parameters and new keyword parameters. The help shows the keyword parameters by default, but you can obtain the single-letter equivalents, where applicable, by adding the `-compatible` parameter after the `-help` parameter.

For example, to obtain help about adding a database service that includes the single-letter equivalents, enter the following command:

```
srvctl add service -help -compatible
```

The single-letter equivalents appear in parentheses next to the keyword parameters. Parameters that are new in Oracle Database 12*c* and later do not have single-letter equivalents.

2.2.4 Adding Components to the Oracle Restart Configuration

In most cases, creating an Oracle component on a host that is running Oracle Restart automatically adds the component to the Oracle Restart configuration. However, in some cases, you must add components manually.

(See Table 2-2.) The component is then automatically restarted when required.

The following are occasions when you must manually add components to the Oracle Restart configuration with SRVCTL:

- You install Oracle Restart after creating the database.
- You create an additional Oracle database on the same host computer using the `CREATE DATABASE` SQL statement.
- You create a database service with `DBMS_SERVICE.CREATE_SERVICE` package procedure. (The recommended way is to use SRVCTL.)

> **Note:**
>
> Adding a component to the Oracle Restart configuration is also referred to as "registering a component with Oracle Restart."

Adding a component to the Oracle Restart configuration does not start that component. You must use a `srvctl start` command to start it.

When you add a component to the Oracle Restart configuration with SRVCTL, you can specify optional configuration settings for the component.

To add a component to the Oracle Restart configuration with SRVCTL:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl add object options
   ```

   where *`object`* is one of the components listed in Table 2-7. See the SRVCTL add command for available options for each component.

Example 2-1   Adding a Database

This example adds a database with a `DB_UNIQUE_NAME` of `dbcrm`. The mandatory `-oraclehome` option specifies the Oracle home location.

```
srvctl add database -db dbcrm -oraclehome /u01/app/oracle/product/
database_release_number/dbhome_1
```

Example 2-2   Adding a Database Service

For the database with the `DB_UNIQUE_NAME` of `dbcrm`, this example both creates a new database service named `crmbatch` and adds it to the Oracle Restart configuration.

```
srvctl add service -db dbcrm -service crmbatch
```

See "Creating and Deleting Database Services with SRVCTL" for more examples.

Example 2-3   Adding the Default Listener

This example adds the default listener to the Oracle Restart configuration.

```
srvctl add listener
```

> **Note:**
>
> When you install a database or manually add a database to the Oracle Restart configuration, and you have a separate Oracle Grid Infrastructure installation owner user, then you must also add the grid user as a member of the `OSRACDBA` group of that database to enable Oracle Grid Infrastructure components to connect to the database. This is because the Oracle Grid Infrastructure components must be able to connect to the database as `SYSRAC` to start and stop the database.
>
> For example, if the host user who installed the Oracle Grid Infrastructure home is named `grid` and the `OSRACDBA` group of the Oracle home is named `racdba`, then user `grid` must be a member of the `racdba` group.

> **See Also:**
>
> - "Starting and Stopping Components Managed by Oracle Restart"
> - "Operating System Groups"
> - "SRVCTL Command Reference for Oracle Restart"

2.2.5 Removing Components from the Oracle Restart Configuration

When you use an Oracle-recommended method to delete an Oracle component, the component is also automatically removed from the Oracle Restart configuration.

For example, if you use Database Configuration Assistant (DBCA) to delete a database, DBCA removes the database from the Oracle Restart configuration. Likewise, if you use Oracle Net Configuration Assistant (NETCA) to delete a listener, NETCA removes the listener from the Oracle Restart configuration. See Table 2-3 for more examples. If you use a non-recommended or manual method to delete an Oracle component, you must first use SRVCTL to remove the component from the Oracle Restart configuration. Failing to do so could result in an error.

To remove a component from the Oracle Restart configuration:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl remove object [options]
   ```

   where *`object`* is one of the components listed in Table 2-7. See the SRVCTL remove command for available options for each component.

Example 2-4 Removing a Database

This example removes a database with a `DB_UNIQUE_NAME` of `dbcrm`.

```
srvctl remove database -db dbcrm
```

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.6 Disabling and Enabling Oracle Restart Management for a Component

You can temporarily disable Oracle Restart management for a component. One reason to do this is when you are performing maintenance on the component. For example, if a component must be repaired, then you might not want it to be automatically restarted if it fails or if the host computer is restarted. When maintenance is complete, you can reenable management for the component.

When you disable a component:

- It is no longer automatically restarted.
- It is no longer automatically started through a dependency.
- It cannot be started with SRVCTL.
- Any component dependent on this resource is no longer automatically started or restarted.

To disable or enable automatic restart for a component:

1. Prepare to run SRVCTL, as described in "Preparing to Run SRVCTL".
2. Do one of the following:
   - To disable a component, enter the following command:

     ```
     srvctl disable object [options]
     ```

   - To enable a component, enter the following command:

     ```
     srvctl enable object [options]
     ```

   Replace *`object`* with one of the components listed in Table 2-7. See the SRVCTL disable command and the enable command for available options for each component.

Example 2-5 Disabling Automatic Restart for a Database

This example disables automatic restart for a database with a `DB_UNIQUE_NAME` of `dbcrm`.

```
srvctl disable database -db dbcrm
```

Example 2-6 Disabling Automatic Restart for an Oracle ASM Disk Group

This example disables automatic restart for the Oracle ASM disk group named `recovery`.

```
srvctl disable diskgroup -diskgroup recovery
```

Example 2-7 Enabling Automatic Restart for an Oracle ASM Disk Group

This example reenables automatic restart for the disk group `recovery`.

```
srvctl enable diskgroup -diskgroup recovery
```

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.7 Viewing Component Status

You can use SRVCTL to view the running status (running or not running) for any component managed by Oracle Restart. For some components, additional information is also displayed.

To view component status:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl status object [options]
   ```

   where *`object`* is one of the components listed in Table 2-7. See the SRVCTL status command for available options for each component.

Example 2-8 Viewing Status of a Database

This example displays the status of the database with a `DB_UNIQUE_NAME` of `dbcrm`.

```
srvctl status database -db dbcrm

Database is running.
```

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.8 Viewing the Oracle Restart Configuration for a Component

You can use SRVCTL to view the Oracle Restart configuration for any component. Oracle Restart maintains different configuration information for each component type. In one form of the SRVCTL command, you can obtain a list of components managed by Oracle Restart.

To view component configuration:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl config object options
   ```

   where *`object`* is one of the components listed in Table 2-7. See the SRVCTL config command for available options for each component.

Example 2-9 Viewing a List of All Databases Managed by Oracle Restart

```
srvctl config database
```

```
dbcrm
orcl
```

Example 2-10 Viewing the Configuration of a Particular Database

This example displays the configuration of the database with a `DB_UNIQUE_NAME` of `orcl`.

```
srvctl config database -db orcl

Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/database_release_number/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain: us.example.com
Start options: open
Stop options: immediate
Database role:
Management policy: automatic
Disk Groups: DATA
Services: mfg,sales
```

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.9 Modifying the Oracle Restart Configuration for a Component

You can use SRVCTL to modify the Oracle Restart configuration of a component. For example, you can modify the port number that a listener listens on when Oracle Restart starts it, or the server parameter file (SPFILE) that Oracle Restart points to when it starts a database.

To modify the Oracle Restart configuration for a component:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl modify object options
   ```

   where *`object`* is one of the components listed in Table 2-7. See the SRVCTL modify command for available options for each component.

Example 2-11 Modifying the Oracle Restart Configuration for a Database

For the database with a `DB_UNIQUE_NAME` of `dbcrm`, the following command changes the management policy to `MANUAL` and the start option to `NOMOUNT`.

```
srvctl modify database -db dbcrm -policy MANUAL -startoption NOMOUNT
```

With a `MANUAL` management policy, the database is never automatically started when the database host computer is restarted. However, Oracle Restart continues to monitor the database and restarts it if a failure occurs.

See Also:

- "Viewing the Oracle Restart Configuration for a Component"
- "SRVCTL Command Reference for Oracle Restart"

2.2.10 Managing Environment Variables in the Oracle Restart Configuration

The Oracle Restart configuration can store name/value pairs for environment variables.

- About Environment Variables in the Oracle Restart Configuration
  You can set environment variable values in the Oracle Restart configuration.
- Setting and Unsetting Environment Variables
  You use SRVCTL to set and unset environment variable values in the Oracle Restart configuration for a component.
- Viewing Environment Variables
  You use SRVCTL to view the values of environment variables in the Oracle Restart configuration for a component.

2.2.10.1 About Environment Variables in the Oracle Restart Configuration

You can set environment variable values in the Oracle Restart configuration.

If you typically set environment variables (other than `ORACLE_HOME` and `ORACLE_SID`) before starting your Oracle database, then you can set these environment variable values in the Oracle Restart configuration. You can store any number environment variables in the individual configurations of the following components:

- Database instance
- Listener
- Oracle ASM instance

When Oracle Restart starts one of these components, it first sets environment variables for that component to the values stored in the component configuration. Although you can set environment variables that are used by Oracle components in this manner, this capability is primarily intended for operating system environment variables.

The following sections provide instructions for setting, unsetting, and viewing environment variables:

- Setting and Unsetting Environment Variables
- Viewing Environment Variables

Note:

Do not use this facility to set standard environment variables like `ORACLE_HOME` and `ORACLE_SID`; these are set automatically by Oracle Restart.

2.2.10.2 Setting and Unsetting Environment Variables

You use SRVCTL to set and unset environment variable values in the Oracle Restart configuration for a component.

To set or unset environment variables in the configuration:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Do one of the following:
   - To set an environment variable in the configuration, enter the following command:

     ```
     srvctl setenv {asm|database|listener} options
     ```

   - To remove an environment variable from the configuration, enter the following command:

     ```
     srvctl unsetenv {asm|database|listener} options
     ```

   See the SRVCTL setenv command and the unsetenv command for available options for each component.

Example 2-12 Setting Database Environment Variables

This example sets the `NLS_LANG` and the AIX `AIXTHREAD_SCOPE` environment variables in the Oracle Restart configuration for the database with a `DB_UNIQUE_NAME` of `dbcrm`:

```
srvctl setenv database -db dbcrm -envs "NLS_LANG=AMERICAN_AMERICA.AL32UTF8,
   AIXTHREAD_SCOPE=S"
```

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.10.3 Viewing Environment Variables

You use SRVCTL to view the values of environment variables in the Oracle Restart configuration for a component.

To view environment variable values in the configuration:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl getenv {database|listener|asm} options
   ```

   See the SRVCTL getenv command for available options for each component.

Example 2-13 Viewing All Environment Variables for a Database

This example gets and displays the environment variables in the Oracle Restart configuration for the database with a `DB_UNIQUE_NAME` of `dbcrm`:

```
srvctl getenv database -db dbcrm

dbcrm:
NLS_LANG=AMERICAN_AMERICA
```

```
AIXTHREAD_SCOPE=S
GCONF_LOCAL_LOCKS=1
```

Example 2-14 Viewing Specific Environment Variables for a Database

This example gets and displays the `NLS_LANG` and `AIXTHREAD_SCOPE` environment variables from the Oracle Restart configuration for the same database:

```
srvctl getenv database -db dbcrm -envs "NLS_LANG,AIXTHREAD_SCOPE"

dbcrm:
NLS_LANG=AMERICAN_AMERICA
AIXTHREAD_SCOPE=S
```

See Also:

"SRVCTL Command Reference for Oracle Restart"

2.2.11 Creating and Deleting Database Services with SRVCTL

When managing a database with Oracle Restart, Oracle recommends that you use SRVCTL to create and delete database services. When you use SRVCTL to add a database service, the service is automatically added to the Oracle Restart configuration and a dependency between the service and the database is established. Thus, if you start the service, Oracle Restart first starts the database if it is not started.

When you use SRVCTL to delete a database service, the service is also removed from the Oracle Restart configuration.

To create a database service with SRVCTL:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl add service -db db_unique_name -service service_name [options]
   ```

   The database service is created and added to the Oracle Restart configuration. See the srvctl add service command for available options.

To delete a database service with SRVCTL:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Enter the following command:

   ```
   srvctl remove service -db db_unique_name -service service_name [-force]
   ```

   The database service is removed from the Oracle Restart configuration. If the `-force` flag is present, the service is removed even if it is still running. Without this flag, an error occurs if the service is running.

Example 2-15 Creating a Database Service

For the database with the `DB_UNIQUE_NAME` of `dbcrm`, this example creates a new database service named `crmbatch`.

```
srvctl add service -db dbcrm -service crmbatch
```

Example 2-16 Creating a Role-Based Database Service

This example creates the `crmbatch` database service and assigns it the Data Guard role of `PHYSICAL_STANDBY`. The service is automatically started only if the current role of the `dbcrm` database is physical standby.

```
srvctl add service -db dbcrm -service crmbatch -role PHYSICAL_STANDBY
```

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.12 Enabling FAN Events in an Oracle Restart Environment

To enable Oracle Restart to publish Fast Application Notification (FAN) events, you must create an Oracle Notification Services (ONS) network that includes the Oracle Restart servers and the integrated clients.

These clients can include Oracle Connection Manager (CMAN), Java Database Connectivity (JDBC), and Universal Connection Pool (UCP) clients. If you are using Oracle Call Interface or ODP.NET clients, then you must enable Oracle Advanced Queuing (AQ) HA notifications for your services. In addition, ONS must be running on the server.

To enable FAN events in an Oracle Restart environment:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Add the database to the Oracle Restart Configuration if it is not already managed by Oracle Restart. See "Adding Components to the Oracle Restart Configuration".
3. Add ONS to the configuration:

   ```
   srvctl add ons
   ```

   ONS is disabled when it is added.
4. Enable ONS:

   ```
   srvctl enable ons
   ```
5. Start ONS:

   ```
   srvctl start ons
   ```
6. Add the service to the Oracle Restart Configuration.

   For Oracle Call Interface and ODP.NET clients, ensure that the `-notification` option is set to `TRUE` to enable the database queue.

   See "Creating and Deleting Database Services with SRVCTL".
7. Enable each client for fast connection failover. See "Enabling Clients for Fast Connection Failover".

See Also:

"SRVCTL Command Reference for Oracle Restart"

2.2.13 Automating the Failover of Connections Between Primary and Standby Databases

In a configuration that uses Oracle Restart and Oracle Data Guard primary and standby databases, the database services fail over automatically from the primary to the standby during either a switchover or failover.

You can use Oracle Notification Services (ONS) to immediately notify clients of the failover of services between the primary and standby databases. The Oracle Data Guard Broker uses Fast Application Notification (FAN) to send notifications to clients when a failover occurs. Integrated Oracle clients automatically failover connections and applications can mask the failure from end-users.

To automate connection failover, you must create an ONS network that includes the Oracle Restart servers and the integrated clients (CMAN, listener, JDBC, and UCP). If you are using Oracle Call Interface or ODP.NET clients, you must enable the Oracle Advanced Queuing queue. The database and the services must be managed by Oracle Restart and the Oracle Data Guard Broker to automate the failover of services.

To automate the failover of services between primary and standby databases:

1. Configure the primary and standby database with the Oracle Data Guard Broker. See *Oracle Data Guard Broker*.
2. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
3. Add the primary database to the Oracle Restart configuration on the primary server if it has not been added. Ensure that you specify `PRIMARY` for the database role. See "Adding Components to the Oracle Restart Configuration".
4. Add the standby database to the Oracle Restart configuration on the standby server if it has not been added. Ensure that you specify the appropriate standby database role.
5. Enable FAN events on both the primary database server and the standby database server. "Enabling FAN Events in an Oracle Restart Environment".
6. Add the services that clients will use to connect to the databases to the Oracle Restart configuration on the primary database and the standby database. When you add a service, ensure that:
   - The `-role` option is set to the proper role for each service
   - The `-notification` option is set to `TRUE` if you are using ODP.NET or Oracle Call Interface

   See "Creating and Deleting Database Services with SRVCTL".
7. Enable each client for fast connection failover. See "Enabling Clients for Fast Connection Failover".

> **See Also:**
>
> "SRVCTL Command Reference for Oracle Restart"

2.2.14 Enabling Clients for Fast Connection Failover

Fast Connection Failover provides high availability to Fast Application Notification (FAN) integrated clients, such as clients that use JDBC, OCI, or ODP.NET. If you configure the client to use fast connection failover, then the client automatically subscribes to FAN events and can react to database UP and DOWN events. In response, Oracle Database gives the client a connection to an active instance that provides the requested database service.

- About Enabling Clients for Fast Connection Failover
  In a configuration with a standby database, after you have added Oracle Notification Services (ONS) to your Oracle Restart configurations and enabled Oracle Advanced Queuing (AQ) HA notifications for your services, you can enable clients for fast connection failover.
- Enabling Fast Connection Failover for JDBC Clients
  Enabling FAN for the Oracle Universal Connection Pool enables Fast Connection Failover (FCF) for the client. Your application can use either thick or thin JDBC clients to use FCF.
- Enabling Fast Connection Failover for Oracle Call Interface Clients
  Oracle Call Interface clients can enable Fast Connection Failover (FCF) by registering to receive notifications about Oracle Restart high availability FAN events and respond when events occur.
- Enabling Fast Connection Failover for ODP.NET Clients
  Oracle Data Provider for .NET (ODP.NET) connection pools can subscribe to notifications that indicate when services are down. After a `DOWN` event, Oracle Database cleans up sessions in the connection pool that go to the instance that stops, and ODP.NET proactively disposes connections that are no longer valid.

2.2.14.1 About Enabling Clients for Fast Connection Failover

In a configuration with a standby database, after you have added Oracle Notification Services (ONS) to your Oracle Restart configurations and enabled Oracle Advanced Queuing (AQ) HA notifications for your services, you can enable clients for fast connection failover.

The clients receive Fast Application Notification (FAN) events and can relocate connections to the current primary database after an Oracle Data Guard failover. See "Automating the Failover of Connections Between Primary and Standby Databases" for information about adding ONS.

For databases with no standby database configured, you can still configure the client FAN events. When there is a failure, you can configure the client to retry the connection to the database. Since Oracle Restart will restart the failed database, the client can reconnect when the database restarts. Ensure that you program the appropriate delay and retries on the connection string, as illustrated in the examples in this section.

2.2.14.2 Enabling Fast Connection Failover for JDBC Clients

Enabling FAN for the Oracle Universal Connection Pool enables Fast Connection Failover (FCF) for the client. Your application can use either thick or thin JDBC clients to use FCF.

To configure the JDBC client, set the `FastConnectionFailoverEnabled` property before making the first `getConnection()` request to a data source. When you enable Fast Connection Failover, the failover applies to every connection in the connection cache. If your application explicitly creates a connection cache using the Connection Cache Manager, then you must first set `FastConnectionFailoverEnabled`.

This section describes how to enable FCF for JDBC with the Universal Connection Pool. For thick JDBC clients, if you enable Fast Connection Failover, do not enable Transparent Application Failover (TAF), either on the client or for the service. Enabling FCF with thin or thick JDBC clients enables the connection pool to receive and react to all FAN events.

To enable Fast Connection Failover for JDBC clients:

1. On a cache enabled DataSource, set the DataSource property `FastConnectionFailoverEnabled` to `true` as in the following example to enable FAN for the Oracle JDBC Implicit Connection Cache:

   ```
   PoolDataSource pds = PoolDataSourceFactory.getPoolDataSource();
   pds.setONSConfiguration("nodes=primaryhost:6200,standbyhost:6200");
   pds.setFastConnectionFailoverEnabled(true);
   pds.setURL("jdbc:oracle:thin:@(DESCRIPTION=
     (LOAD_BALANCE=on)
     (ADDRESS=(PROTOCOL=TCP)(HOST=primaryhost)(PORT=1521))
     (ADDRESS=(PROTOCOL=TCP)(HOST=standbyhost)(PORT=1521))
     (CONNECT_DATA=(service_name=service_name)))");

   ......
   ```

   In this example, `primaryhost` is the server for the primary database, and `standbyhost` is the server for the standby database.

   Applications must have both ucp.jar and ons.jar in their `CLASSPATH`.

   > **Note:**
   >
   > Use the following system property to enable FAN without making data source changes: `-D oracle.jdbc.FastConnectionFailover=true`.

2. When you start the application, ensure that the ons.jar file is located on the application `CLASSPATH`. The ons.jar file is part of the Oracle client installation.

> **See Also:**
>
> - *Oracle Database JDBC Developer's Guide*
> - *Oracle Universal Connection Pool Developer’s Guide*

2.2.14.3 Enabling Fast Connection Failover for Oracle Call Interface Clients

Oracle Call Interface clients can enable Fast Connection Failover (FCF) by registering to receive notifications about Oracle Restart high availability FAN events and respond when events occur.

This improves the session failover response time in Oracle Call Interface and removes terminated connections from connection and session pools. This feature works on Oracle Call Interface applications, including those that use Transparent Application Failover (TAF), connection pools, or session pools.

First, you must enable a service for high availability events to automatically populate the Advanced Queuing `ALERT_QUEUE`. If your application is using TAF, then enable the TAF settings for the service. Configure client applications to connect to an Oracle Restart database. Clients can register callbacks that are used whenever an event occurs. This reduces the time that it takes to detect a connection failure.

During `DOWN` event processing, Oracle Call Interface:

- Terminates affected connections at the client and returns an error
- Removes connections from the Oracle Call Interface connection pool and the Oracle Call Interface session pool

  The session pool maps each session to a physical connection in the connection pool, and there can be multiple sessions for each connection.
- Fails over the connection if you have configured TAF

If TAF is not configured, then the client only receives an error.

> **Note:**
>
> Oracle Call Interface does not manage `UP` events.

To Enable Fast Connection Failover for an Oracle Call Interface client:

1. Ensure that the service that you are using has Advanced Queuing notifications enabled by setting the services' values using the SRVCTL `modify` command. For example:

   ```
   srvctl modify service -db proddb -service gl.us.example.com -notification
   true -role primary -failovertype select -failovermethod basic -failoverretry 5
   -failoverdelay 180 -clbgoal long
   ```
2. Enable `OCI_EVENTS` at environment creation time on the client as follows:

   ```
   ( OCIEnvCreate(...) )
   ```
3. Link client applications with the client thread or operating system library.
4. (Optional) Register a client `EVENT` callback.
5. Ensure that the client uses an Oracle Net connect descriptor that includes all primary and standby hosts in the `ADDRESS_LIST`. For example:

   ```
   gl =
   (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
      (ADDRESS_LIST =
   ```

```
         (ADDRESS = (PROTOCOL = TCP)(HOST = BOSTON1)(PORT = 1521))
         (ADDRESS = (PROTOCOL = TCP)(HOST = CHICAGO1)(PORT = 1521))
         (LOAD_BALANCE = yes)
    )
    (CONNECT_DATA=
      (SERVICE_NAME=gl.us.example.com)))
```

To see the alert information, query the views `DBA_OUTSTANDING_ALERTS` and `DBA_ALERT_HISTORY`.

See Also:

- *Oracle Call Interface Programmer's Guide*
- *Oracle Database Net Services Administrator's Guide* for information about configuring TAF

2.2.14.4 Enabling Fast Connection Failover for ODP.NET Clients

Oracle Data Provider for .NET (ODP.NET) connection pools can subscribe to notifications that indicate when services are down. After a `DOWN` event, Oracle Database cleans up sessions in the connection pool that go to the instance that stops, and ODP.NET proactively disposes connections that are no longer valid.

All three ODP.NET providers (core, managed, and unmanaged) support FCF.

To enable Fast Connection Failover for ODP.NET clients:

1. Enable Fast Application Notification (FAN) by using SRVCTL `modify service` command, as in the following example:

   ```
   srvctl modify service -db dbname -service gl -notification true
   ```

2. Enable Fast Connection Failover for ODP.NET connection pools by subscribing to FAN high availability events. Set the `HA` Events connection string attribute to true at connection time. In newer ODP.NET versions, `HA` Events is set to true by default. The pooling attribute must be set to `true`, which is the default. The following example illustrates these settings, where *`user_name`* is the name of the user and *`password`* is the user password:

   ```
   // C#
   using System;
   using Oracle.ManagedDataAccess.Client;
   //using Oracle.DataAccess.Client;

   class HAEventEnablingSample
   {
     static void Main()
     {
       OracleConnection con = new OracleConnection();

       // Open a connection using ConnectionString attributes
       // Also, enable "load balancing"
       con.ConnectionString =
         "User Id=user_name;Password=password;Data Source=oracle;" +
         "Min Pool Size=10;Connection Lifetime=120;Connection Timeout=60;" +
         "HA Events=true;Incr Pool Size=5;Decr Pool Size=2";
   ```

```
      con.Open();

      // Create more connections and perform work against the database here.

      // Dispose OracleConnection object
      con.Dispose();
    }
  }
```

3. Ensure that the client uses an Oracle Net connect descriptor that includes all primary and standby hosts in the `ADDRESS_LIST`. For example:

```
gl =
(DESCRIPTION =
 (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(HOST = BOSTON1)(PORT = 1521))
     (ADDRESS = (PROTOCOL = TCP)(HOST = CHICAGO1)(PORT = 1521))
     (LOAD_BALANCE = yes)
 )
 (CONNECT_DATA=
  (SERVICE_NAME=gl.us.example.com))
 )
)
```

See Also:

- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for information about ODP.NET
- "SRVCTL Command Reference for Oracle Restart"

2.3 Starting and Stopping Components Managed by Oracle Restart

When Oracle Restart is in use, Oracle strongly recommends that you use the SRVCTL utility to start and stop components.

Use the SRVCTL utility to start and stop components for the following reasons:

- When starting a component with SRVCTL, Oracle Restart can first start any components on which this component depends. When stopping a component with SRVCTL, Oracle Restart can stop any dependent components first.
- SRVCTL always starts a component according to its Oracle Restart configuration. Starting a component by other means may not.

  For example, if you specified a server parameter file (SPFILE) location when you added a database to the Oracle Restart configuration, and that location is not the default location for SPFILEs, if you start the database with SQL*Plus, the SPFILE specified in the configuration may not be used.

  See the srvctl add database command for a table of configuration options for a database instance.

- When you start a component with SRVCTL, environment variables stored in the Oracle Restart configuration for the component are set.

  See "Managing Environment Variables in the Oracle Restart Configuration" for more information.

You can start and stop any component managed by Oracle Restart with SRVCTL.

To start or stop a component managed by Oracle Restart with SRVCTL:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Do one of the following:
   - To start a component, enter the following command:

     ```
     srvctl start object [options]
     ```

   - To stop a component, enter the following command:

     ```
     srvctl stop object [options]
     ```

   where *object* is one of the components listed in Table 2-7. See the SRVCTL start command and the stop command for available options for each component.

Example 2-17 Starting a Database

This example starts the database with a DB_UNIQUE_NAME of dbcrm:

```
srvctl start database -db dbcrm
```

Example 2-18 Starting a Database NOMOUNT

This example starts the database instance without mounting the database:

```
srvctl start database -db dbcrm -startoption nomount
```

Example 2-19 Starting the Default Listener

This example starts the default listener:

```
srvctl start listener
```

Example 2-20 Starting a Specified Listener

This example starts the listener named crmlistener:

```
srvctl start listener -listener crmlistener
```

Example 2-21 Starting Database Services

This example starts the database services bizdev and support for the database with a DB_UNIQUE_NAME of dbcrm. If the database is not started, Oracle Restart first starts the database.

```
srvctl start service -db dbcrm -service "bizdev,support"
```

Example 2-22 Starting (Mounting) Oracle ASM Disk Groups

This example starts (mounts) the Oracle ASM disk groups data and recovery. The user running this command must be a member of the OSASM group.

```
srvctl start diskgroup -diskgroup "data,recovery"
```

Example 2-23 Shutting Down a Database

This example stops (shuts down) the database with a DB_UNIQUE_NAME of dbcrm. Because a stop option (-stopoption) is not provided, the database shuts down according to the stop option in its Oracle Restart configuration. The default stop option is IMMEDIATE.

```
srvctl stop database -db dbcrm
```

Example 2-24 Shutting Down a Database with the ABORT option

This example does a SHUTDOWN ABORT of the database with a DB_UNIQUE_NAME of dbcrm.

```
srvctl stop database -db dbcrm -stopoption abort
```

Note:

After relinking Oracle executables, use the SRVCTL utility to start and stop components when Oracle Restart is in use. Typically, relinking Oracle executables is required on a Linux or UNIX-based operating system after you apply an operating system patch or after an operating system upgrade. See *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for more information about relinking.

If you use SQL*Plus to start and stop components, then you must first run the setasmgidwrap script after relinking. See *Oracle Database Upgrade Guide* for information about running this script.

See Also:

The SRVCTL start command

2.4 Stopping and Restarting Oracle Restart for Maintenance Operations

When several components in an Oracle home are managed by Oracle Restart, you can stop Oracle Restart and the components managed by Oracle Restart in the Oracle home.

You can also disable Oracle Restart so that it is not restarted if the node reboots. You might need to do this when you are performing maintenance that includes the Oracle home, such as installing a patch. When the maintenance operation is complete, you can enable and restart Oracle Restart, and you can restart the components managed by Oracle Restart in the Oracle home.

Use both the SRVCTL utility and the CRSCTL utility for the stop and start operations:

- The stop home SRVCTL command stops all of the components that are managed by Oracle Restart in the specified Oracle home. The start home SRVCTL command starts these components. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.

When you use the home object, a state file, specified in the -statefile option, tracks the state of each component. The stop and status commands create the state file. The start command uses the state file to identify the components to restart.

In addition, you can check the status of the components managed by Oracle Restart using the status home command.

- The stop CRSCTL command stops Oracle Restart, and the disable CRSCTL command ensures that the components managed by Oracle Restart do not restart automatically. The enable CRSCTL command enables automatic restart and the start CRSCTL command restarts Oracle Restart.

To stop and start the components in an Oracle home while installing a patch:

1. Prepare to run SRVCTL as described in "Preparing to Run SRVCTL".
2. Use the SRVCTL utility to stop the components managed by Oracle Restart in an Oracle home:

   ```
   srvctl stop home -oraclehome oracle_home -statefile state_file [-stopoption
   stop_options] [-force]
   ```

   where *oracle_home* is the complete path of the Oracle home and *state_file* is the complete path to the state file. State information for the Oracle home is recorded in the specified state file. Make a note of the state file location because it must be specified in Step 7.

   Before stopping the components in an Oracle Grid Infrastructure home, ensure that you first stop the components in a dependent Oracle Database home.
3. If you are patching an Oracle Grid Infrastructure home, then disable and stop Oracle Restart. Otherwise, go to Step 4.

   To disable and stop Oracle Restart, use the CRSCTL utility to run the following commands:

   ```
   crsctl disable has

   crsctl stop has
   ```
4. Perform the maintenance operation.
5. Use the CRSCTL utility to enable automatic restart of the components managed by Oracle Restart:

   ```
   crsctl enable has
   ```
6. Use the CRSCTL utility to start Oracle Restart:

   ```
   crsctl start has
   ```
7. Use the SRVCTL utility to start the components that were stopped in Step 2:

   ```
   srvctl start home -oraclehome oracle_home -statefile state_file
   ```

   The state file must match the state file specified in Step 2.
8. (Optional) Use the SRVCTL utility to check the status of the components managed by Oracle Restart in the Oracle home:

   ```
   srvctl status home -oraclehome oracle_home -statefile state_file
   ```

Example 2-25 Stopping Components Managed by Oracle Restart in an Oracle Home

```
srvctl stop home -oraclehome /u01/app/oracle/product/database_release_number/dbhome_1 -
statefile /usr1/or_state
```

Example 2-26 Starting Components Managed by Oracle Restart in an Oracle Home

```
srvctl start home -oraclehome /u01/app/oracle/product/database_release_number/dbhome_1 -
statefile /usr1/or_state
```

Example 2-27 Displaying the Status of Components Managed by Oracle Restart in an Oracle Home

```
srvctl status home -oraclehome /u01/app/oracle/product/database_release_number/dbhome_1 -
statefile /usr1/or_state
```

See Also:

- The srvctl stop home command
- The srvctl status home command
- The srvctl start home command
- "CRSCTL Command Reference"

2.5 SRVCTL Command Reference for Oracle Restart

You can reference details about the syntax and options for SRVCTL commands specific to Oracle Restart.

See *Oracle Real Application Clusters Administration and Deployment Guide* for the full list of SRVCTL commands.

SRVCTL Command Syntax and Options Overview

SRVCTL expects the following command syntax:

```
srvctl command object options
```

where:

- `command` is a verb such as `start`, `stop`, or `remove`.
- `object` is the component on which SRVCTL performs the command, such as database, listener, and so on. You can also use component abbreviations. See Table 2-7 for a complete list of components and their abbreviations.
- `options` extend the use of a preceding command combination to include additional parameters for the command. For example, the `-db` option indicates that a database unique name follows, and the `-service` option indicates that a comma-delimited list of database service names follows.

Note:

On the Windows platform, when specifying a comma-delimited list, you must enclose the list within double-quotes ("...,..."). You must also use double-quotes on the UNIX and Linux platforms if any list member contains shell metacharacters.

Case Sensitivity

SRVCTL commands and components are case insensitive. Options are case sensitive. Database and database service names are case insensitive and case preserving.

Command Parameters Input File

You can specify command parameters in a file rather than directly on the command line. Using a command parameters input file is useful in the following situations:

- You want to run a command with very long parameter values or a command with numerous parameters
- You want to bypass shell processing of certain special characters

To specify a command parameters input file, use the `-file` parameter with a value that is the location of the command parameters file. SRVCTL processes the command parameters from the command parameters file instead of from the command line.

SRVCTL Components Summary

Table 2-7 lists the keywords that can be used for the *`object`* portion of SRVCTL commands. You can use either the full name or the abbreviation for each component keyword.

Table 2-7 Component Keywords and Abbreviations

Component	Abbreviation	Description
`asm`	`asm`	Oracle ASM instance
`database`	`db`	Database instance
`diskgroup`	`dg`	Oracle ASM disk group
`home`	`home`	Oracle home or Oracle Clusterware home
`listener`	`lsnr`	Oracle Net listener
`service`	`serv`	Database service
`ons`	`ons`	Oracle Notification Services (ONS)

- add
  The `srvctl add` command adds the specified component to the Oracle Restart configuration, and optionally sets Oracle Restart configuration parameters for the component. After a component is added, Oracle Restart begins to manage it, restarting it when required.
- config
  The `srvctl config` command displays the Oracle Restart configuration of the specified component or set of components.
- disable
  Disables a component, which suspends management of that component by Oracle Restart.
- downgrade
  The `srvctl downgrade` command downgrades the database configuration after you manually downgrade the database.

- enable
  The `srvctl enable` command reenables the specified disabled component.
- getenv
  Gets and displays environment variables and their values from the Oracle Restart configuration for a database, listener, or Oracle ASM instance.
- modify
  Modifies the Oracle Restart configuration of a component. The change takes effect when the component is next restarted.
- remove
  Removes the specified component from the Oracle Restart configuration. Oracle Restart no longer manages the component. Any environment variable settings for the component are also removed.
- setenv
  The `setenv` command sets values of environment variables in the Oracle Restart configuration for a database, a listener, or the Oracle ASM instance.
- start
  Starts the specified component or components.
- status
  Displays the running status of the specified component or set of components.
- stop
  Stops the specified component or components.
- unsetenv
  The `unsetenv` command deletes one or more environment variables from the Oracle Restart configuration for a database, a listener, or an Oracle ASM instance.
- update
  The `srvctl update` command updates the running database to switch to the specified startup option.
- upgrade
  The `srvctl upgrade` command upgrades the resources types and resources from an older version to a newer version.

> **See Also:**
>
> Table 2-1

2.5.1 add

The `srvctl add` command adds the specified component to the Oracle Restart configuration, and optionally sets Oracle Restart configuration parameters for the component. After a component is added, Oracle Restart begins to manage it, restarting it when required.

To perform `srvctl add` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

> **Note:**
>
> There is no `srvctl add` command for Oracle ASM disk groups. Disk groups are automatically added to the Oracle Restart configuration when they are first mounted. If you remove a disk group from the Oracle Restart configuration and later want to add it back, connect to the Oracle ASM instance with SQL*Plus and use an `ALTER DISKGROUP` ... `MOUNT` command.

- srvctl add asm
  Adds an Oracle ASM instance to the Oracle Restart configuration.
- srvctl add database
  Adds a database to the Oracle Restart configuration.
- srvctl add listener
  Adds a listener to the Oracle Restart configuration.
- srvctl add ons
  Adds Oracle Notification Services (ONS) to an Oracle Restart configuration.
- srvctl add service
  Adds a database service to the Oracle Restart configuration.

2.5.1.1 srvctl add asm

Adds an Oracle ASM instance to the Oracle Restart configuration.

- Syntax and Options
- Example

2.5.1.1.1 Syntax and Options

Use the `srvctl add asm` command with the following syntax:

```
srvctl add asm [-listener listener_name] [-spfile spfile]
  [-pwfile password_file_path] [-diskstring asm_diskstring]
```

Table 2-8 srvctl add asm Options

Option	Description
`-listener` *`listener_name`*	Name of the listener with which Oracle ASM should register. A weak dependency is established with this listener. (Before starting the Oracle ASM instance, Oracle Restart attempts to start the listener. If the listener does not start, the Oracle ASM instance is still started. If the listener later fails, Oracle Restart does not restart Oracle ASM.) If omitted, defaults to the listener named `listener`.
`-spfile` *`spfile`*	The full path of the server parameter file for the database. If omitted, the default SPFILE is used.
`-pwfile` *`password_file_path`*	The full path of the Oracle ASM password file.

Table 2-8 (Cont.) srvctl add asm Options

Option	Description
`-diskstring` *`asm_diskstring`*	Oracle ASM disk group discovery string. An Oracle ASM discovery string is a comma-delimited list of strings that limits the set of disks that an Oracle ASM instance discovers. The discovery strings can include wildcard characters. Only disks that match one of the strings are discovered.

2.5.1.1.2 Example

An example of this command is:

```
srvctl add asm -listener crmlistener
```

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle ASM disk group discovery strings

2.5.1.2 srvctl add database

Adds a database to the Oracle Restart configuration.

After adding a database to the Oracle Restart configuration, if the database then accesses data in an Oracle ASM disk group, a dependency between the database that disk group is created. Oracle Restart then ensures that the disk group is mounted before attempting to start the database.

However, if the database and Oracle ASM instance are not running when you add the database to the Oracle Restart configuration, you must manually establish the dependency between the database and its disk groups by specifying the `-diskgroup` option in the SRVCTL command. See the example later in this section.

> **Note:**
>
> When you manually add a database to the Oracle Restart configuration, you must also add the Oracle grid infrastructure software owner as a member of the OSDBA group of that database. This is because the grid infrastructure components must be able to connect to the database as `SYSDBA` to start and stop the database.
>
> For example, if the host user who installed the grid infrastructure home is named `grid` and the OSDBA group of the new database is named `dba`, then user `grid` must be a member of the `dba` group.

- Syntax and Options
- Examples

2.5.1.2.1 Syntax and Options

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

```
srvctl add database -db db_unique_name -oraclehome oracle_home
  [-domain domain_name] [-dbname db_name] [-instance instance_name]
  [-spfile spfile][-pwfile password_file_path] [-startoption start_options]
  [-stopoption stop_options]
  [-role {PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY |
         SNAPSHOT_STANDBY | FAR_SYNC}]
  [-policy {AUTOMATIC | MANUAL | NORESTART}] [-diskgroup disk_group_list]
  [-verbose]
```

Table 2-9   srvctl add database Options

Syntax	Description
`-db` *db_unique_name*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-oraclehome` *oracle_home*	The full path of Oracle home for the database
`-domain` *domain_name*	The domain for the database. Must match the `DB_DOMAIN` initialization parameter.
`-dbname` *db_name*	If provided, must match the `DB_NAME` initialization parameter setting. You must include this option if `DB_NAME` is different from the unique name given by the `-db` option
`-instance` *instance_name*	The instance name. You must include this option if the instance name is different from the unique name given by the `-db` option. For example, if the unique name includes an underscore, and the instance name omits the underscore, then use this parameter to specify the instance name.
`-spfile` *spfile*	The full path of the server parameter file for the database. If omitted, the default SPFILE is used.
`-pwfile` *password_file_path*	The full path of the database password file.
`-startoption` *start_options*	Startup options for the database (`OPEN`, `MOUNT`, or `NOMOUNT`). If omitted, defaults to `OPEN`. **See Also:** *SQL*Plus User's Guide and Reference* for more information about startup options
`-stopoption` *stop_options*	Shutdown options for the database (`NORMAL`, `IMMEDIATE`, `TRANSACTIONAL`, or `ABORT`). If omitted, defaults to `IMMEDIATE`. **See Also:** *SQL*Plus User's Guide and Reference* for more information about shutdown options
`-role {PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY | SNAPSHOT_STANDBY | FAR_SYNC}`	The current role of the database (`PRIMARY`, `PHYSICAL_STANDBY`, `LOGICAL_STANDBY`, `SNAPSHOT_STANDBY`, or `FAR_SYNC`). The default is `PRIMARY`. Applicable in Oracle Data Guard environments only. **See Also:** *Oracle Data Guard Concepts and Administration* for more information about database roles

Table 2-9 (Cont.) srvctl add database Options

Syntax	Description
`-policy {AUTOMATIC | MANUAL | NORESTART}`	Management policy for the database. • `AUTOMATIC` (default): The database is automatically restored to its previous running condition (started or stopped) upon restart of the database host computer. • `MANUAL`: The database is never automatically restarted upon restart of the database host computer. A `MANUAL` setting does not prevent Oracle Restart from monitoring the database while it is running and restarting it if a failure occurs. • `NORESTART`: Similar to the `MANUAL` setting, the database is never automatically restarted upon restart of the database host computer. A `NORESTART` setting, however, never restarts the database even if a failure occurs.
`-diskgroup` *`disk_group_list`*	Comma separated list of disk groups upon which the database is dependent. When starting the database, Oracle Restart first ensures that these disk groups are mounted. This option is required only if the database instance and the Oracle ASM instance are not started when adding the database. Otherwise, the dependency is recorded automatically between the database and its disk groups.
`-verbose`	Verbose output

2.5.1.2.2 Examples

This example adds the database with the `DB_UNIQUE_NAME dbcrm`:

```
srvctl add database -db dbcrm -oraclehome /u01/app/oracle/product/
database_release_number/dbhome_1
```

This example adds the same database and also establishes a dependency between the database and the disk groups `DATA` and `RECOVERY`.

```
srvctl add database -db dbcrm -oraclehome /u01/app/oracle/product/
database_release_number/dbhome_1
  -diskgroup "DATA,RECOVERY"
```

See Also:

- "Oracle Restart Integration with Oracle Data Guard"
- *Oracle Data Guard Concepts and Administration*

2.5.1.3 srvctl add listener

Adds a listener to the Oracle Restart configuration.

- Syntax and Options
- Example

2.5.1.3.1 Syntax and Options

Use the `srvctl add listener` command with the following syntax:

```
srvctl add listener [-listener listener_name] [-endpoints endpoints] [-skip]
  [-oraclehome oracle_home]
```

Table 2-10 srvctl add listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, defaults to `LISTENER`
`-endpoints` *`endpoints`*	Comma separated TCP ports or listener endpoints. If omitted, defaults to TCP:1521. *endpoints* syntax is: `"[TCP:]port[, ...] [/IPC:key] [/NMP:pipe_name] [/TCPS:s_port] [/SDP:port]"`
`-skip`	Skip checking for port conflicts with the supplied endpoints
`-oraclehome` *`oracle_home`*	Oracle home for the listener. If omitted, the Oracle Grid Infrastructure home is assumed.

2.5.1.3.2 Example

The following command adds a listener (named `LISTENER`) running out of the database Oracle home and listening on TCP port 1522:

```
srvctl add listener -endpoints TCP:1522
  -oraclehome /u01/app/oracle/product/database_release_number/dbhome_1
```

2.5.1.4 srvctl add ons

Adds Oracle Notification Services (ONS) to an Oracle Restart configuration.

ONS must be added to an Oracle Restart configuration to enable the sending of Fast Application Notification (FAN) events after an Oracle Data Guard failover.

When ONS is added to an Oracle Restart configuration, it is initially disabled. You can enable it with the `srvctl enable ons` command.

- Syntax and Options

See Also:

"srvctl enable ons"

2.5.1.4.1 Syntax and Options

Use the `srvctl add ons` command with the following syntax:

```
srvctl add ons [-emport em_port] [-onslocalport ons_local_port]
  [-onsremoteport ons_remote_port] [-remoteservers host[:port],[host[:port]...]]
  [-verbose]
```

Table 2-11 srvctl add ons Options

Option	Description
`-emport` *`em_port`*	ONS listening port for Oracle Enterprise Manager Cloud Control (Cloud Control). The default is 2016.
`-onslocalport` *`ons_local_port`*	ONS listening port for local client connections. The default is 6100.
`-onsremoteport` *`ons_remote_port`*	ONS listening port for connections from remote hosts. The default is 6200.
`-remoteservers host[:`*`port`*`],[`*`host`*`[:`*`port`*`],...`	A list of *`host:port`* pairs of remote hosts that are part of the ONS network **Note:** If *`port`* is not specified for a remote host, then `ons_remote_port` is used.
`-verbose`	Verbose output

2.5.1.5 srvctl add service

Adds a database service to the Oracle Restart configuration.

Creates the database service if it does not exist. This method of creating a service is preferred over using the `DBMS_SERVICE` PL/SQL package.

- Syntax and Options
- Example

2.5.1.5.1 Syntax and Options

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

```
srvctl add service -db db_unique_name -service service_name
  [-role [PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY][,SNAPSHOT_STANDBY]]
  [-policy {AUTOMATIC | MANUAL}]
  [-failovertype {NONE | SESSION | SELECT | TRANSACTION}]
  [-failovermethod {NONE | BASIC}] [-failoverdelay integer]
  [-failoverretry integer] [-clbgoal {SHORT | LONG}]
  [-rlbgoal {SERVICE_TIME | THROUGHPUT | NONE}] [-notification {TRUE | FALSE}]
  [-edition edition_name] [-pdb pluggable_database]
  [-sql_translation_profile sql_translation_profile]
  [-commit_outcome {TRUE | FALSE}] [-retention retention]
  [-replay_init_time replay_init_time] [-drain_timeout timeout]
  [-stopoption stop_option] [-session_state {STATIC | DYNAMIC}]
  [-global {TRUE | FALSE}] [-maxlag max_lag_time] [-force] [-verbose]
```

Table 2-12 srvctl add service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database The name must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service` *`service_name`*	The database service name

Table 2-12 (Cont.) srvctl add service Options

Option	Description
`-role [PRIMARY] [,PHYSICAL_STANDBY] [,LOGICAL_STANDBY] [,SNAPSHOT_STANDBY]`	A list of service roles This option is applicable in Oracle Data Guard environments only. When this option is present, upon database open, the service is started only when one of its service roles matches the current database role. **See Also:** *Oracle Data Guard Concepts and Administration* for more information about database roles
`-policy {AUTOMATIC | MANUAL}`	Management policy for the service If `AUTOMATIC` (the default), the service is automatically started upon restart of the database, either by a planned restart (with SRVCTL) or after a failure. Automatic restart is also subject to the service role, however (the `-role` option). If `MANUAL`, the service is never automatically restarted upon planned restart of the database (with SRVCTL). A `MANUAL` setting does not prevent Oracle Restart from monitoring the service when it is running and restarting it if a failure occurs.
`-failovertype {NONE |SESSION | SELECT | TRANSACTION}`	To enable Application Continuity for OCI and Java, use `TRANSACTION`. If the failover type is `TRANSACTION`, then OCI and Java attempt to recover the in progress transaction upon receipt of a recoverable error. When failover type is `TRANSACTION`, the `-commit_outcome` option must be set to `TRUE`. To enable Transparent Application Failover (TAF) for OCI, use `SELECT` or `SESSION`.
`-failovermethod {NONE | BASIC}`	TAF failover method for backward compatibility only If the failover type (`-failovertype`) is set to a value other than `NONE`, then use `BASIC` for this option.
`-failoverdelay` *`integer`*	For Application Continuity and TAF, the time delay, in seconds, between reconnect attempts for each incident at failover
`-failoverretry` *`integer`*	For Application Continuity and TAF, the number of attempts to connect after an incident
`-clbgoal {SHORT | LONG}`	Connection load balancing goal Use `SHORT` for run-time load balancing. Use `LONG` for long running connections, such as batch jobs.
`-rlbgoal {SERVICE_TIME | THROUGHPUT | NONE}`	Run-time load balancing goal Use `SERVICE_TIME` to balance connections by response time. Use `THROUGHPUT` to balance connections by throughput.
`-notification {TRUE | FALSE}`	Enable Fast Application Notification (FAN) for OCI connections
`-edition` *`edition_name`*	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. SRVCTL does not validate the specified edition name. During connection, the connect user must have `USE` privilege on the specified edition. If the edition does not exist or if the connect user does not have `USE` privilege on the specified edition, then an error is raised.

Table 2-12 (Cont.) srvctl add service Options

Option	Description
`-pdb` *`pluggable_database`*	In a multitenant container database (CDB), the name of the pluggable database (PDB) to associate with the service If this option is set to an empty string, then the service is associated with root.
`-sql_translation_profile` *`sql_translation_profile`*	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 This parameter corresponds to the SQL translation profile parameter in the `DBMS_SERVICE` service attribute. **Notes:** • Before using the SQL translation framework, you must migrate all server-side application objects and data to the Oracle database. • Use the `srvctl config service` command to display the SQL translation profile. **See Also:** *Oracle Database SQL Translation and Migration Guide* for more information about using a SQL translation profile
`-commit_outcome` `{TRUE | FALSE}`	For Transaction Guard, when `TRUE` a transaction's commit outcome is accessible after the transaction's session fails due to a recoverable outage. If `FALSE`, the default, then a transaction's commit outcome is not retained. When this option is set to `TRUE`, the outcome of a transaction's commit is durable, and an applications can determine the commit status of a transaction after an outage. You can set `commit_outcome` to `TRUE` for a user-defined service. The `commit_outcome` setting has no effect on Oracle Active Data Guard and read-only databases. **See Also:** See *Oracle Database Development Guide* for more information.
`-retention` *`retention`*	If `commit_outcome` is set to `TRUE`, then this option determines the amount of time, in seconds, that the commit outcome is retained. The default is 24 hours (86400). If `commit_outcome` is set to `FALSE`, then this option cannot be set.
`-replay_init_time` *`replay_init_time`*	For Application Continuity, this option specifies the difference between the time, in seconds, of original execution of the first operation of a request and the time that the replay is ready to start after a successful reconnect. Application Continuity will not replay after the specified amount of time has passed. This option is intended to avoid the unintentional execution of a transaction when a system is recovered after a long period of time. The default is 5 minutes (300). The maximum value is 24 hours (86400). If `failovertype` is not set to `TRANSACTION`, then this option is not used.
`-drain_timeout` *`timeout`*	This option specifies the time allowed for resource draining to be completed in seconds. Permitted values are `NULL`, `0`, or any positive integer. The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. How draining works depends on the setting of the `-stopoption` option. The default value is `NULL`, which means that this option is not set. If the option is not set, and `-drain_timeout` has been set on the service, then this value is used. If it is set to `0`, then draining does not occur.

Table 2-12 (Cont.) srvctl add service Options

Option	Description
`-stopoption` *`stop_option`*	This option specifies the mode in which the service is stopped. The following values are permitted: • `IMMEDIATE` specifies that sessions are permitted to drain before the service is stopped. • `TRANSACTIONAL` specifies that sessions are permitted to drain for the amount of time specified in the `–drain_timeout` option. The service is stopped when the time limit is reached, and any remaining sessions are terminated. • `NONE` is the default.
`-session_state {STATIC | DYNAMIC}`	For Application Continuity, this parameter specifies whether the session state that is not transactional is changed by the application. Oracle recommends a setting of `DYNAMIC` for most applications. **Note:** This parameter is considered only if `-failovertype` is set to `TRANSACTION` for Application Continuity. It describes how non-transactional is changed during a request. Examples of session state are NLS settings, optimizer preferences, event settings, PL/SQL global variables, temporary tables, advanced queues, LOBs, and result cache. If non-transactional values change after the request starts, then use the default, `DYNAMIC`. Most applications should use `DYNAMIC` mode. If you are unsure, then use `DYNAMIC` mode.
`-global {TRUE | FALSE}`	If `TRUE`, then the service is a Global Data Services (GDS) service and is managed by the Global Services Manager (GSM). If `FALSE`, the default, then the service is not a GDS service. The global attribute of a service cannot be changed after the service is added. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.
`-maxlag` *`maximum_lag_time`*	Maximum replication lag time in seconds. Must be a non-negative integer. The default value is `ANY`.
`-force`	Force the add operation even though a listener is not configured for a network.
`-verbose`	Verbose output

2.5.1.5.2 Example

This example adds the `sales` service for the database with `DB_UNIQUE_NAME` `dbcrm`. The service is started only when `dbcrm` is in `PRIMARY` mode.

```
srvctl add service -db dbcrm -service sales -role PRIMARY
```

See Also:

- The section in *Oracle Database PL/SQL Packages and Types Reference* on the `DBMS_SERVICE` package for more information about the options for this command
- "Oracle Restart Integration with Oracle Data Guard"
- *Oracle Data Guard Concepts and Administration*
- *Oracle Multitenant Administrator's Guide* for information about creating, modifying, or removing a service for a pluggable database (PDB)

2.5.2 config

The `srvctl config` command displays the Oracle Restart configuration of the specified component or set of components.

- srvctl config asm
  Displays the Oracle Restart configuration information for the Oracle ASM instance.
- srvctl config database
  Displays the Oracle Restart configuration information for the specified database, or lists all databases managed by Oracle Restart.
- srvctl config listener
  Displays the Oracle Restart configuration information for all Oracle Restart–managed listeners or for the specified listener.
- srvctl config ons
  Displays the current configuration information for Oracle Notification Services (ONS).
- srvctl config service
  For the specified database, displays the Oracle Restart configuration information for the specified database service or for all Oracle Restart–managed database services.

2.5.2.1 srvctl config asm

Displays the Oracle Restart configuration information for the Oracle ASM instance.

- Syntax and Options
- Example

2.5.2.1.1 Syntax and Options

Use the `srvctl config asm` command with the following syntax:

```
srvctl config asm [-all]
```

Table 2-13 srvctl config asm Options

Option	Description
`-all`	Display enabled/disabled status also

2.5.2.1.2 Example

An example of this command is:

```
srvctl config asm -all

asm home: /u01/app/oracle/product/database_release_number/grid
ASM is enabled.
```

2.5.2.2 srvctl config database

Displays the Oracle Restart configuration information for the specified database, or lists all databases managed by Oracle Restart.

- Syntax and Options
- Example

2.5.2.2.1 Syntax and Options

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

```
srvctl config database [-db db_unique_name [-all]] [-verbose]
```

Table 2-14   srvctl config database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-all`	Display detailed configuration information
`-verbose`	Verbose output

2.5.2.2.2 Example

An example of this command to list all Oracle Restart–managed databases is:

```
srvctl config database

dbcrm
orcl
```

An example of this command to display configuration and enabled/disabled status for the database with the `DB_UNIQUE_ID` `orcl` is:

```
srvctl config database -db orcl -all

Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/database_release_number/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain: us.example.com
```

```
Start options: open
Stop options: immediate
Database role:
Management policy: automatic
Disk Groups: DATA
Services: mfg,sales
Database is enabled
```

2.5.2.3 srvctl config listener

Displays the Oracle Restart configuration information for all Oracle Restart–managed listeners or for the specified listener.

- Syntax and Options
- Example

2.5.2.3.1 Syntax and Options

Use the `srvctl config listener` command with the following syntax:

```
srvctl config listener [-listener listener_name]
```

Table 2-15 srvctl config listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, configuration information for all Oracle Restart–managed listeners is displayed.

2.5.2.3.2 Example

This example displays the configuration information and enabled/disabled status for the default listener:

```
srvctl config listener

Name: LISTENER
Home: /u01/app/oracle/product/database_release_number/dbhome_1
End points: TCP:1521
Listener is enabled.
```

2.5.2.4 srvctl config ons

Displays the current configuration information for Oracle Notification Services (ONS).

- Syntax and Options

2.5.2.4.1 Syntax and Options

Use the `srvctl config ons` command with the following syntax:

```
srvctl config ons
```

2.5.2.5 srvctl config service

For the specified database, displays the Oracle Restart configuration information for the specified database service or for all Oracle Restart–managed database services.

- Syntax and Options
- Example

2.5.2.5.1 Syntax and Options

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

```
srvctl config service -db db_unique_name [-service service_name] [-verbose]
```

Table 2-16 srvctl config service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service` *`service_name`*	Database service name. If omitted, SRVCTL displays configuration information for all Oracle Restart–managed services for the database.
`-verbose`	Verbose output

2.5.2.5.2 Example

An example of this command is:

```
srvctl config service -db dbcrm -service sales

Service name: sales
Service is enabled
Cardinality: SINGLETON
Disconnect: true
Service role: PRIMARY
Management policy: automatic
DTP transaction: false
AQ HA notifications: false
Failover type: NONE
Failover method: NONE
TAF failover retries: 0
TAF failover delay: 0
Connection Load Balancing Goal: NONE
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
Edition: e2
```

2.5.3 disable

Disables a component, which suspends management of that component by Oracle Restart.

The `srvctl disable` command is intended to be used when a component must be repaired or shut down for maintenance, and should not be restarted automatically. When you disable a component:

- It is no longer automatically restarted.
- It is no longer automatically started through a dependency.
- It cannot be started with SRVCTL.

To perform `srvctl disable` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

- srvctl disable asm
  Disables the Oracle ASM instance.
- srvctl disable database
  Disables the specified database.
- srvctl disable diskgroup
  Disables an Oracle ASM disk group.
- srvctl disable listener
  Disables the specified listener or all listeners.
- srvctl disable ons
  Disables Oracle Notification Services (ONS).
- srvctl disable service
  Disables one or more database services.

> **See Also:**
>
> The enable command

2.5.3.1 srvctl disable asm

Disables the Oracle ASM instance.

- Syntax and Options

2.5.3.1.1 Syntax and Options

Use the `srvctl disable asm` command with the following syntax:

```
srvctl disable asm
```

2.5.3.2 srvctl disable database

Disables the specified database.

- Syntax and Options
- Example

2.5.3.2.1 Syntax and Options

Use the `srvctl disable database` command with the following syntax:

```
srvctl disable database -db db_unique_name
```

Table 2-17 srvctl disable database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.

2.5.3.2.2 Example

An example of this command is:

```
srvctl disable database -db dbcrm
```

2.5.3.3 srvctl disable diskgroup

Disables an Oracle ASM disk group.

- Syntax and Options
- Example

2.5.3.3.1 Syntax and Options

Use the `srvctl disable diskgroup` command with the following syntax:

```
srvctl disable diskgroup -diskgroup diskgroup_name
```

Table 2-18 srvctl disable diskgroup Options

Option	Description
`-diskgroup` *`diskgroup_name`*	Disk group name

2.5.3.3.2 Example

An example of this command is:

```
srvctl disable diskgroup -diskgroup DATA
```

2.5.3.4 srvctl disable listener

Disables the specified listener or all listeners.

- Syntax and Options
- Example

2.5.3.4.1 Syntax and Options

Use the `srvctl disable listener` command with the following syntax:

```
srvctl disable listener [-listener listener_name]
```

Table 2-19 srvctl disable listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, all listeners are disabled.

2.5.3.4.2 Example

An example of this command is:

```
srvctl disable listener -listener crmlistener
```

2.5.3.5 srvctl disable ons

Disables Oracle Notification Services (ONS).

- Syntax and Options

2.5.3.5.1 Syntax and Options

Use the `srvctl disable ons` command with the following syntax:

```
srvctl disable ons [-verbose]
```

Table 2-20 srvctl disable ons Options

Option	Description
`-verbose`	Verbose output

2.5.3.6 srvctl disable service

Disables one or more database services.

- Syntax and Options
- Example

2.5.3.6.1 Syntax and Options

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

```
srvctl disable service -db db_unique_name -service service_name_list
  [-global_override
```

Table 2-21 srvctl disable service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.

Table 2-21 (Cont.) srvctl disable service Options

Option	Description
`-service` *`service_name_list`*	Comma-delimited list of database service names
`-global_override`	If the service is a Global Data Services (GDS) service, then this option must be specified to disable the service. An error is returned if you attempt to disable a GDS service and –`global_override` is not included. This option is ignored if the service is not a GDS service. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.

2.5.3.6.2 Example

The following example disables the database service `sales` and `mfg`:

```
srvctl disable service -db dbcrm -service sales,mfg
```

2.5.4 downgrade

The `srvctl downgrade` command downgrades the database configuration after you manually downgrade the database.

- srvctl downgrade database
  The `srvctl downgrade database` command downgrades the configuration of a database and its services from its current version to the specified lower version.

2.5.4.1 srvctl downgrade database

The `srvctl downgrade database` command downgrades the configuration of a database and its services from its current version to the specified lower version.

- Syntax and Options

2.5.4.1.1 Syntax and Options

Use the `srvctl downgrade database` command with the following syntax:

```
srvctl downgrade database -db db_unique_name -oraclehome oracle_home
       -targetversion to_version
```

Table 2-22 srvctl downgrade database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-oraclehome` *`oracle_home`*	The full path of Oracle home for the database

Table 2-22 (Cont.) srvctl downgrade database Options

Option	Description
`-targetversion` *`to_version`*	The version to which to downgrade

2.5.5 enable

The `srvctl enable` command reenables the specified disabled component.

When you enable a component:

- Oracle Restart can automatically restart it.
- It can be automatically started through a dependency.
- You can start it manually with SRVCTL.

If the component is already enabled, then the command is ignored.

When you add a component to the Oracle Restart configuration, it is enabled by default.

To perform `srvctl enable` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

- srvctl enable asm
  Enables an Oracle ASM instance.
- srvctl enable database
  Enables the specified database.
- srvctl enable diskgroup
  Enables an Oracle ASM disk group.
- srvctl enable listener
  Enables the specified listener or all listeners.
- srvctl enable ons
  Enables Oracle Notification Services (ONS).
- srvctl enable service
  Enables one or more database services for the specified database.

> **See Also:**
>
> The disable command

2.5.5.1 srvctl enable asm

Enables an Oracle ASM instance.

- Syntax and Options

2.5.5.1.1 Syntax and Options

Use the `srvctl enable asm` command with the following syntax:

```
srvctl enable asm
```

2.5.5.2 srvctl enable database

Enables the specified database.

- Syntax and Options
- Example

2.5.5.2.1 Syntax and Options

Use the `srvctl enable database` command with the following syntax:

```
srvctl enable database -db db_unique_name
```

Table 2-23 srvctl enable database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.

2.5.5.2.2 Example

An example of this command is:

```
srvctl enable database -db dbcrm
```

2.5.5.3 srvctl enable diskgroup

Enables an Oracle ASM disk group.

- Syntax and Options
- Example

2.5.5.3.1 Syntax and Options

Use the `srvctl enable diskgroup` command with the following syntax:

```
srvctl enable diskgroup -diskgroup diskgroup_name
```

Table 2-24 srvctl enable diskgroup Options

Option	Description
`-diskgroup` *`diskgroup_name`*	Disk group name

2.5.5.3.2 Example

An example of this command is:

```
srvctl enable diskgroup -diskgroup DATA
```

2.5.5.4 srvctl enable listener

Enables the specified listener or all listeners.

- Syntax and Options
- Example

2.5.5.4.1 Syntax and Options

Use the `srvctl enable listener` command with the following syntax:

```
srvctl enable listener [-listener listener_name]
```

Table 2-25 srvctl enable listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, all listeners are enabled.

2.5.5.4.2 Example

An example of this command is:

```
srvctl enable listener -listener crmlistener
```

2.5.5.5 srvctl enable ons

Enables Oracle Notification Services (ONS).

- Syntax and Options

2.5.5.5.1 Syntax and Options

Use the `srvctl enable ons` command with the following syntax:

```
srvctl enable ons [-verbose]
```

Table 2-26 srvctl enable ons Options

Option	Description
`-verbose`	Verbose output

2.5.5.6 srvctl enable service

Enables one or more database services for the specified database.

- Syntax and Options
- Example

2.5.5.6.1 Syntax and Options

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

```
srvctl enable service -db db_unique_name -service service_name_list
  [-global_override]
```

Table 2-27   srvctl enable service Options

Option	Description
`-db db_unique_name`	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service service_name_list`	Comma-delimited list of database service names
`-global_override`	If the service is a Global Data Services (GDS) service, then this option must be specified to enable the service. An error is returned if you attempt to enable a GDS service and –`global_override` is not included. This option is ignored if the service is not a GDS service. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.

2.5.5.6.2 Example

The following example enables the database services `sales` and `mfg` in the database with `DB_UNIQUE_NAME dbcrm`:

```
srvctl enable service -db dbcrm -service "sales,mfg"
```

2.5.6 getenv

Gets and displays environment variables and their values from the Oracle Restart configuration for a database, listener, or Oracle ASM instance.

- srvctl getenv asm
  Displays the configured environment variables for the Oracle ASM instance.
- srvctl getenv database
  Displays the configured environment variables for the specified database.
- srvctl getenv listener
  Displays the configured environment variables for the specified listener.

See Also:

- setenv command
- unsetenv command
- "Managing Environment Variables in the Oracle Restart Configuration"

2.5.6.1 srvctl getenv asm

Displays the configured environment variables for the Oracle ASM instance.

- Syntax and Options
- Example

2.5.6.1.1 Syntax and Options

Use the `srvctl getenv asm` command with the following syntax:

```
srvctl getenv asm [-envs name_list]
```

Table 2-28 srvctl getenv asm Options

Options	Description
`-envs name_list`	Comma-delimited list of names of environment variables to display. If omitted, SRVCTL displays all configured environment variables for Oracle ASM.

2.5.6.1.2 Example

The following example displays all configured environment variables for the Oracle ASM instance:

```
srvctl getenv asm
```

2.5.6.2 srvctl getenv database

Displays the configured environment variables for the specified database.

- Syntax and Options
- Example

2.5.6.2.1 Syntax and Options

Use the `srvctl getenv database` command with the following syntax:

```
srvctl getenv database -db db_unique_name [-envs name_list]
```

Table 2-29 srvctl getenv database Options

Options	Description
`-db db_unique_name`	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-envs name_list`	Comma-delimited list of names of environment variables to display. If omitted, SRVCTL displays all configured environment variables.

2.5.6.2.2 Example

The following example displays all configured environment variables for the database with `DB_UNIQUE_NAME dbcrm`:

```
srvctl getenv database -db dbcrm
```

2.5.6.3 srvctl getenv listener

Displays the configured environment variables for the specified listener.

- Syntax and Options
- Example

2.5.6.3.1 Syntax and Options

Use the `srvctl getenv listener` command with the following syntax:

```
srvctl getenv listener [-listener listener_name] [-envs name_list]
```

Table 2-30 srvctl getenv listener Options

Options	Description
`-listener` *`listener_name`*	Listener name. If omitted, SRVCTL lists environment variables for all listeners.
`-envs` *`name_list`*	Comma-delimited list of names of environment variables to display. If omitted, SRVCTL displays all configured environment variables.

2.5.6.3.2 Example

The following example displays all configured environment variables for the listener named `crmlistener`:

```
srvctl getenv listener -listener crmlistener
```

2.5.7 modify

Modifies the Oracle Restart configuration of a component. The change takes effect when the component is next restarted.

To perform `srvctl modify` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

- srvctl modify asm
  Modifies the Oracle Restart configuration for the Oracle ASM instance.
- srvctl modify database
  Modifies the Oracle Restart configuration for a database.
- srvctl modify listener
  Modifies the Oracle Restart configuration for the specified listener or all listeners.
- srvctl modify ons
  Modifies Oracle Notification Services (ONS).

- srvctl modify service
  Modifies the Oracle Restart configuration of a database service.

2.5.7.1 srvctl modify asm

Modifies the Oracle Restart configuration for the Oracle ASM instance.

- Syntax and Options
- Example

2.5.7.1.1 Syntax and Options

Use the `srvctl modify asm` command with the following syntax:

```
srvctl modify asm [-listener listener_name] [-spfile spfile]
  [-pwfile password_file_path] [-diskstring asm_diskstring]
```

Table 2-31 srvctl modify asm Options

Option	Description
`-listener` *`listener_name`*	Name of the listener with which Oracle ASM must register. A weak dependency is established with this listener. (Before Oracle ASM is started, Oracle Restart ensures that this listener is started.)
`-spfile` *`spfile`*	The full path of the server parameter file for the database. If omitted, the default SPFILE is used.
`-pwfile` *`password_file_path`*	The full path of the Oracle ASM password file.
`-diskstring` *`asm_diskstring`*	Oracle ASM disk group discovery string. An Oracle ASM discovery string is a comma-delimited list of strings that limits the set of disks that an Oracle ASM instance discovers. The discovery strings can include wildcard characters. Only disks that match one of the strings are discovered.

2.5.7.1.2 Example

An example of this command is:

```
srvctl modify asm -listener crmlistener
```

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle ASM disk group discovery strings

2.5.7.2 srvctl modify database

Modifies the Oracle Restart configuration for a database.

- Syntax and Options
- Example

2.5.7.2.1 Syntax and Options

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

```
srvctl modify database -db db_unique_name [-oraclehome oracle_home]
  [-user oracle_user] [-domain domain_name] [-dbname db_name]
  [-instance instance_name] [-instance instance_name] [-spfile spfile]
  [-pwfile password_file_path] [-startoption start_options]
  [-stopoption stop_options]
  [-role {PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY | SNAPSHOT_STANDBY}]
  [-policy {AUTOMATIC | MANUAL | NORESTART}]
  [{-diskgroup "diskgroup_list" | -nodiskgroup}] [-force]
```

Table 2-32   srvctl modify database Options

Option	Description
`-db` *db_unique_name*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-user` *oracle_user*	Name of the Oracle user who owns the Oracle home directory
`-diskgroup` *disk_group_list*	Comma separated list of disk groups upon which the database is dependent. When starting the database, Oracle Restart first ensures that these disk groups are mounted. This option is required only if the database instance and the Oracle ASM instance are not started when adding the database. Otherwise, the dependency is recorded automatically between the database and its disk groups.
`-nodiskgroup`	Remove the database's dependency on Oracle ASM disk groups
`-force`	Force the operation even though the some resources might be stopped.
`(Other options)`	See Table 2-9

2.5.7.2.2 Example

The following example changes the role of the database with `DB_UNIQUE_NAME dbcrm` to `LOGICAL_STANDBY`:

```
srvctl modify database -db dbcrm -role logical_standby
```

See Also:

- "Oracle Restart Integration with Oracle Data Guard"
- *Oracle Data Guard Concepts and Administration*

2.5.7.3 srvctl modify listener

Modifies the Oracle Restart configuration for the specified listener or all listeners.

- Syntax and Options
- Example

2.5.7.3.1 Syntax and Options

Use the `srvctl modify listener` command with the following syntax:

```
srvctl modify listener [-listener listener_name] [-endpoints endpoints]
  [-oraclehome oracle_home]
```

Table 2-33 srvctl modify listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, all listener configurations are modified.
`-endpoints` *`endpoints`*	Comma separated TCP ports or listener endpoints. *endpoints* syntax is: `"[TCP:]port[, ...] [/IPC:key] [/NMP:pipe_name] [/TCPS:s_port] [/SDP:port]"`
`-oraclehome` *`oracle_home`*	New Oracle home for the listener

2.5.7.3.2 Example

This example modifies the TCP port on which the listener named `crmlistener` listens:

```
srvctl modify listener -listener crmlistener -endpoints TCP:1522
```

2.5.7.4 srvctl modify ons

Modifies Oracle Notification Services (ONS).

- Syntax and Options

2.5.7.4.1 Syntax and Options

Use the `srvctl modify ons` command with the following syntax:

```
srvctl modify ons [-emport em_port] [-onslocalport ons_local_port]
  [-onsremoteport ons_remote_port] [-remoteservers host[:port],[host[:port]...]]
  [-verbose]
```

Table 2-34 srvctl modify ons Options

Option	Description
`-emport` *`em_port`*	ONS listening port for Cloud Control. The default is 2016.
`-onslocalport` *`ons_local_port`*	ONS listening port for local client connections
`-onsremoteport` *`ons_remote_port`*	ONS listening port for connections from remote hosts
`-remoteservers host[:port],[host[:port],...`	A list of `host:port` pairs of remote hosts that are part of the ONS network **Note:** If `port` is not specified for a remote host, then `ons_remote_port` is used.
`-verbose`	Verbose output

2.5.7.5 srvctl modify service

Modifies the Oracle Restart configuration of a database service.

> **Note:**
>
> Oracle recommends that you limit configuration changes to the minimum requirement and that you not perform other service operations while the online service modification is in progress.

- Syntax and Options
- Example

2.5.7.5.1 Syntax and Options

Use the `srvctl modify service` command with the following syntax:

```
srvctl modify service -db db_unique_name -service service_name
  [-role [PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY][,SNAPSHOT_STANDBY]]
  [-policy {AUTOMATIC | MANUAL}]
  [-failovertype {NONE | SESSION | SELECT | TRANSACTION}]
  [-failovermethod {NONE | BASIC}] [-failoverdelay integer]
  [-failoverretry integer] [-clbgoal {SHORT | LONG}]
  [-rlbgoal {SERVICE_TIME | THROUGHPUT | NONE}] [-notification {TRUE | FALSE}]
  [-edition edition_name] [-pdb pluggable_database]
  [-sql_translation_profile sql_translation_profile]
  [-commit_outcome {TRUE | FALSE}] [-retention retention]
  [-replay_init_time replay_init_time] [-drain_timeout timeout]
  [-stopoption stop_option] [-session_state {STATIC | DYNAMIC}]
  [-global_override] [-verbose]
```

Table 2-35 srvctl modify service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database The name must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service` *`service_name`*	Service name
`-role [PRIMARY] [,PHYSICAL_STANDBY] [,LOGICAL_STANDBY] [,SNAPSHOT_STANDBY]`	A list of service roles This option is applicable in Oracle Data Guard environments only. When this option is present, upon database startup, the service is started only when one of its service roles matches the current database role. **See Also:** *Oracle Data Guard Concepts and Administration* for more information about database roles

Table 2-35 (Cont.) srvctl modify service Options

Option	Description
`-policy {AUTOMATIC | MANUAL}`	Management policy for the service If `AUTOMATIC` (the default), the service is automatically started upon restart of the database, either by a planned restart (with SRVCTL) or after a failure. Automatic restart is also subject to the service role, however (the `-role` option). If `MANUAL`, the service is never automatically restarted upon planned restart of the database (with SRVCTL). A `MANUAL` setting does not prevent Oracle Restart from monitoring the service when it is running and restarting it if a failure occurs.
`-failovertype {NONE |SESSION | SELECT | TRANSACTION}`	To enable Application Continuity for OCI and Java, use `TRANSACTION`. If the failover type is `TRANSACTION`, then OCI and Java attempt to recover the inflight transaction upon receipt of a recoverable error. When failover type is `TRANSACTION`, the `commit_outcome` option must be set to `TRUE`. To enable Transparent Application Failover (TAF) for OCI, use `SELECT` or `SESSION`.
`-failovermethod {NONE | BASIC}`	TAF failover method for backward compatibility only If the failover type (`-failovertype`) is set to a value other than `NONE`, then use `BASIC` for this option.
`-failoverdelay` *`integer`*	For Application Continuity and TAF, the time delay, in seconds, between reconnect attempts for each incident at failover
`-failoverretry` *`integer`*	For Application Continuity and TAF, the number of attempts to connect after an incident
`-clbgoal {SHORT | LONG}`	Connection load balancing goal Use `SHORT` for run-time load balancing. Use `LONG` for long running connections, such as batch jobs.
`-rlbgoal {SERVICE_TIME | THROUGHPUT | NONE}`	Run-time load balancing goal Use `SERVICE_TIME` to balance connections by response time. Use `THROUGHPUT` to balance connections by throughput.
`-notification {TRUE | FALSE}`	Enable Fast Application Notification (FAN) for OCI connections
`-edition` *`edition_name`*	The initial session edition of the service If this option is not specified, then the edition is not modified for the service. If this option is specified but *`edition_name`* is empty, then the edition is set to `NULL`. A `NULL` edition has no effect. 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. SRVCTL does not validate the specified edition name. During connection, the connect user must have `USE` privilege on the specified edition. If the edition does not exist or if the connect user does not have `USE` privilege on the specified edition, then an error is raised.
`-pdb` *`pluggable_database`*	In a CDB, the name of the PDB to associate with the service If this option is set to an empty string, then the service is associated with root.

Table 2-35 (Cont.) srvctl modify service Options

Option	Description
`-sql_translation_profile` *`sql_translation_profile`*	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 **Note:** Before using the SQL translation framework, you must migrate all server-side application objects and data to the Oracle database. **See Also:** *Oracle Database SQL Translation and Migration Guide* for more information about using a SQL translation profile
`-commit_outcome {TRUE | FALSE}`	For Transaction Guard, when `TRUE` a transaction's commit outcome is accessible after the transaction's session fails due to a recoverable outage. If `FALSE`, the default, then a transaction's commit outcome is not retained. When this option is set to `TRUE`, the outcome of a transaction's commit is durable, and an applications can determine the commit status of a transaction after an outage. You can set `commit_outcome` to `TRUE` for a user-defined service. The `commit_outcome` setting has no effect on Oracle Active Data Guard and read-only databases. **See Also:** See *Oracle Database Development Guide* for more information.
`-retention` *`retention`*	If `commit_outcome` is set to `TRUE`, then this option determines the amount of time, in seconds, that the commit outcome is retained. The default is 24 hours (86400). If `commit_outcome` is set to `FALSE`, then this option cannot be set.
`-replay_init_time` *`replay_init_time`*	For Application Continuity, this option specifies the difference between the time, in seconds, of original execution of the first operation of a request and the time that the replay is ready to start after a successful reconnect. Application Continuity will not replay after the specified amount of time has passed. This option is intended to avoid the unintentional execution of a transaction when a system is recovered after a long period of time. The default is 5 minutes (300). The maximum value is 24 hours (86400). If `failovertype` is not set to `TRANSACTION`, then this option is not used.
`-drain_timeout` *`timeout`*	This option specifies the time allowed for resource draining to be completed in seconds. Permitted values are `NULL`, `0`, or any positive integer. The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. How draining works depends on the setting of the `-stopoption` option. The default value is `NULL`, which means that this option is not set. If the option is not set, and `-drain_timeout` has been set on the service, then this value is used. If it is set to `0`, then draining does not occur.
`-stopoption` *`stop_option`*	This option specifies the mode in which the service is stopped. The following values are permitted: • `IMMEDIATE` specifies that sessions are permitted to drain before the service is stopped. • `TRANSACTIONAL` specifies that sessions are permitted to drain for the amount of time specified in the `–drain_timeout` option. The service is stopped when the time limit is reached, and any remaining sessions are terminated. • `NONE` is the default.

Table 2-35 (Cont.) srvctl modify service Options

Option	Description
`-session_state {STATIC | DYNAMIC}`	For Application Continuity, this parameter specifies whether the session state that is not transactional is changed by the application. Oracle recommends a setting of `DYNAMIC` for most applications. **Note:** This parameter is considered only if `-failovertype` is set to `TRANSACTION` for Application Continuity. It describes how non-transactional is changed during a request. Examples of session state are NLS settings, optimizer preferences, event settings, PL/SQL global variables, temporary tables, advanced queues, LOBs, and result cache. If non-transactional values change after the request starts, then use the default, `DYNAMIC`. Most applications should use `DYNAMIC` mode. If you are unsure, then use `DYNAMIC` mode.
`-global_override`	If the service is a Global Data Services (GDS) service, then this option must be specified to modify any of the following service attributes: • `-role` • `-policy` • `-failovertype` • `-failovermethod` • `-failoverdelay` • `-failoverretry` • `-edition` • `-clbgoal` • `-rlbgoal` • `-notification` An error is returned if you attempt to modify one of these options for a GDS service and `-global_override` is not included. This option is ignored if the service is not a GDS service. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.
`-verbose`	Verbose output

2.5.7.5.2 Example

For the database with a `DB_UNIQUE_NAME` of `dbcrm`, the following command changes the Oracle Data Guard role of the database service named `support` to `standby`:

```
srvctl modify service -db dbcrm -service support -role standby
```

> **See Also:**
>
> *Oracle Multitenant Administrator's Guide* for information about managing services associated with PDBs

2.5.8 remove

Removes the specified component from the Oracle Restart configuration. Oracle Restart no longer manages the component. Any environment variable settings for the component are also removed.

Before you remove a component from the Oracle Restart configuration, you must use SRVCTL to stop it. Oracle recommends that you disable the component before removing it, but this is not required.

To perform `srvctl remove` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

- srvctl remove asm
  Removes an Oracle ASM instance.
- srvctl remove database
  Removes a database. Prompts for confirmation first.
- srvctl remove diskgroup
  Removes an Oracle ASM disk group.
- srvctl remove listener
  Removes the specified listener or all listeners.
- srvctl remove ons
  Removes Oracle Notification Services (ONS).
- srvctl remove service
  Removes the specified database service.

> **See Also:**
>
> - stop command
> - disable command

2.5.8.1 srvctl remove asm

Removes an Oracle ASM instance.

- Syntax and Options
- Example

2.5.8.1.1 Syntax and Options

Use the `srvctl remove asm` command with the following syntax:

```
srvctl remove asm [-force]
```

Table 2-36 srvctl remove asm Options

Options	Description
`-force`	Force remove, even when disk groups and databases that use Oracle ASM exist or when the Oracle ASM instance is running.

2.5.8.1.2 Example

An example of this command is:

```
srvctl remove asm
```

2.5.8.2 srvctl remove database

Removes a database. Prompts for confirmation first.

- Syntax and Options
- Example

2.5.8.2.1 Syntax and Options

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

> **Note:**
>
> After running this command, ensure that the password file is in the default location if you want to connect to the database as the `SYS` user with the `SYS` user's password.

```
srvctl remove database -db db_unique_name [-force] [-noprompt] [-verbose]
```

Table 2-37 srvctl remove database Options

Options	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-force`	Force. Removes the database even if it is running.
`-noprompt`	Suppresses the confirmation prompt and removes immediately
`-verbose`	Verbose output. A success or failure message is displayed.

2.5.8.2.2 Example

An example of this command is:

```
srvctl remove database -db dbcrm
```

2.5.8.3 srvctl remove diskgroup

Removes an Oracle ASM disk group.

- Syntax and Options
- Example

2.5.8.3.1 Syntax and Options

Use the `srvctl remove diskgroup` command with the following syntax:

```
srvctl remove diskgroup -diskgroup diskgroup_name [-force]
```

Table 2-38 srvctl remove diskgroup Options

Option	Description
`-diskgroup` *`diskgroup_name`*	Disk group name
`-force`	Force. Removes the disk group even if files are open on it.

2.5.8.3.2 Example

This example removes the disk group named `DATA`. An error is returned if files are open on this disk group.

```
srvctl remove diskgroup -diskgroup DATA
```

2.5.8.4 srvctl remove listener

Removes the specified listener or all listeners.

- Syntax and Options
- Example

2.5.8.4.1 Syntax and Options

Use the `srvctl remove listener` command with the following syntax:

```
srvctl remove listener [-listener listener_name | -all] [-force]
```

Table 2-39 srvctl remove listener Options

Options	Description
`-listener` *`listener_name`*	Name of the listener that you want to remove. If omitted, then the default is `LISTENER`.
`-all`	Remove all listeners
`-force`	Force. Removes the listener even if databases are using it.

2.5.8.4.2 Example

The following command removes the listener lsnr01:

```
srvctl remove listener -listener lsnr01
```

2.5.8.5 srvctl remove ons

Removes Oracle Notification Services (ONS).

- Syntax and Options

2.5.8.5.1 Syntax and Options

Use the srvctl remove ons command as follows:

```
srvctl remove ons [-force] [-verbose]
```

Table 2-40 srvctl remove ons Options

Options	Description
-force	Force. Removes ONS even if it is enabled.
-verbose	Verbose output

2.5.8.6 srvctl remove service

Removes the specified database service.

- Syntax and Options
- Example

2.5.8.6.1 Syntax and Options

Use the srvctl remove service command as follows:

```
srvctl remove service -db db_unique_name -service service_name [-global_override]
```

Table 2-41 srvctl remove service Options

Options	Description
-db *db_unique_name*	Unique name for the database. Must match the DB_UNIQUE_NAME initialization parameter setting. If DB_UNIQUE_NAME is unspecified, then this option must match the DB_NAME initialization parameter setting. The default setting for DB_UNIQUE_NAME uses the setting for DB_NAME.
-service *service_name*	Service name

Table 2-41 (Cont.) srvctl remove service Options

Options	Description
`-global_override`	If the service is a Global Data Services (GDS) service, then this option must be specified to remove the service. An error is returned if you attempt to remove a GDS service and –`global_override` is not included. This option is ignored if the service is not a GDS service. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.

2.5.8.6.2 Example

An example of this command is:

```
srvctl remove service -db dbcrm -service sales
```

2.5.9 setenv

The `setenv` command sets values of environment variables in the Oracle Restart configuration for a database, a listener, or the Oracle ASM instance.

To perform `srvctl setenv` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

- srvctl setenv asm
  Sets the values of environment variables in the Oracle Restart configuration for the Oracle ASM instance. Before starting the instance, Oracle Restart sets environment variables to the values stored in the configuration.
- srvctl setenv database
  Sets the values of environment variables in the Oracle Restart configuration for a database instance. Before starting the instance, Oracle Restart sets environment variables to the values stored in the configuration.
- srvctl setenv listener
  Sets the values of environment variables in the Oracle Restart configuration for a listener. Before starting the listener, Oracle Restart sets environment variables to the values stored in the configuration.

> **See Also:**
>
> - getenv command
> - unsetenv command
> - "Managing Environment Variables in the Oracle Restart Configuration"

2.5.9.1 srvctl setenv asm

Sets the values of environment variables in the Oracle Restart configuration for the Oracle ASM instance. Before starting the instance, Oracle Restart sets environment variables to the values stored in the configuration.

- Syntax and Options
- Example

2.5.9.1.1 Syntax and Options

Use the `srvctl setenv asm` command with the following syntax:

```
srvctl setenv asm {-envs name=val[,name=val,...] | -env name=val}
```

Table 2-42 srvctl setenv database Options

Options	Description
`-envs name=val[,name=val,...]`	Comma-delimited list of name/value pairs of environment variables
`-env name=val`	Enables single environment variable to be set to a value that contains commas or other special characters

2.5.9.1.2 Example

The following example sets the AIX operating system environment variable `AIXTHREAD_SCOPE` in the Oracle ASM instance configuration:

```
srvctl setenv asm -envs AIXTHREAD_SCOPE=S
```

2.5.9.2 srvctl setenv database

Sets the values of environment variables in the Oracle Restart configuration for a database instance. Before starting the instance, Oracle Restart sets environment variables to the values stored in the configuration.

- Syntax and Options
- Example

2.5.9.2.1 Syntax and Options

Use the `srvctl setenv database` command with the following syntax:

```
srvctl setenv database -db db_unique_name
  {-envs name=val[,name=val,...] | -env name=val}
```

Table 2-43 srvctl setenv database Options

Options	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-envs` *`name`*`=`*`val`*`[,`*`name`*`=`*`val`*`,...]`	Comma-delimited list of name/value pairs of environment variables
`-env` *`name`*`=`*`val`*	Enables single environment variable to be set to a value that contains commas or other special characters

2.5.9.2.2 Example

The following example sets the `LANG` environment variable in the configuration of the database with a `DB_UNIQUE_NAME` of `dbcrm`:

```
srvctl setenv database -db dbcrm -envs LANG=en
```

2.5.9.3 srvctl setenv listener

Sets the values of environment variables in the Oracle Restart configuration for a listener. Before starting the listener, Oracle Restart sets environment variables to the values stored in the configuration.

- Syntax and Options
- Example

2.5.9.3.1 Syntax and Options

Use the `srvctl setenv listener` command with the following syntax:

```
srvctl setenv listener [-listener listener_name]
  {-envs name=val[,name=val,...] | -env name=val}
```

Table 2-44 srvctl setenv listener Options

Options	Description
`-listener` *`listener_name`*	Listener name. If omitted, sets the specified environment variables in all listener configurations.
`-envs` *`name`*`=`*`val`*`[,`*`name`*`=`*`val`*`,...]`	Comma-delimited list of name/value pairs of environment variables
`-env` *`name`*`=`*`val`*	Enables single environment variable to be set to a value that contains commas or other special characters

2.5.9.3.2 Example

The following example sets the AIX operating system environment variable `AIXTHREAD_SCOPE` in the configuration of the listener named `crmlistener`:

```
srvctl setenv listener -listener crmlistener -envs AIXTHREAD_SCOPE=S
```

2.5.10 start

Starts the specified component or components.

- srvctl start asm
  Starts the Oracle ASM instance.
- srvctl start database
  Starts the specified database instance.
- srvctl start diskgroup
  Starts (mounts) an Oracle ASM disk group.
- srvctl start home
  Starts all of the components that are managed by Oracle Restart in the specified Oracle home. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.
- srvctl start listener
  Starts the specified listener or all listeners.
- srvctl start ons
  Starts Oracle Notification Services (ONS).
- srvctl start service
  Starts the specified database service or services.

> **See Also:**
>
> "Starting and Stopping Components Managed by Oracle Restart"

2.5.10.1 srvctl start asm

Starts the Oracle ASM instance.

For this command, SRVCTL connects "/ as sysasm" to perform the operation. To run such operations, the owner of the executables in the Oracle Grid Infrastructure home must be a member of the OSASM group, and users running the commands must also be in the OSASM group.

- Syntax and Options
- Example

2.5.10.1.1 Syntax and Options

Use the `srvctl start asm` command with the following syntax:

```
srvctl start asm [-startoption start_options]
```

Table 2-45 srvctl start asm Option

Option	Description
`-startoption` *`start_options`*	Comma-delimited list of options for the startup command (`OPEN`, `MOUNT`, `NOMOUNT`, or `FORCE`). If omitted, defaults to normal startup (`OPEN`). **See Also:** *SQL*Plus User's Guide and Reference* for more information about startup options

2.5.10.1.2 Example

This example starts the Oracle ASM instance, which then mounts any disk groups named in the `ASM_DISKGROUPS` initialization parameter:

```
srvctl start asm
```

This example starts the Oracle ASM instance without mounting any disk groups:

```
srvctl start asm -startoption nomount
```

2.5.10.2 srvctl start database

Starts the specified database instance.

For this command, SRVCTL connects "/ as sysdba" to perform the operation. To run such operations, the owner of the Oracle executables in the database Oracle home must be a member of the OSDBA group (for example, the `dba` group on UNIX and Linux), and users running the commands must also be in the OSDBA group.

- Syntax and Options
- Example

2.5.10.2.1 Syntax and Options

Use the `srvctl start database` command with the following syntax:

```
srvctl start database -db db_unique_name [-startoption start_options] [-verbose]
```

Table 2-46 srvctl start database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.

Table 2-46 (Cont.) srvctl start database Options

Option	Description
`-startoption` *`start_options`*	Comma-delimited list of options for the startup command (for example: `OPEN`, `MOUNT`, `NOMOUNT`, `RESTRICT`, and so on) **Notes:** • This command parameter does not support the `PFILE` option or the `QUIET` option, but it supports all other database startup options. • For multi-word startup options, such as `read only` and `read write`, separate the words with a space and enclose in single quotation marks (`''`). For example, `'read only'`. **See Also:** *SQL*Plus User's Guide and Reference* for more information about startup options
`-verbose`	Verbose output

2.5.10.2.2 Example

An example of this command is:

```
srvctl start database -db dbcrm -startoption nomount
```

2.5.10.3 srvctl start diskgroup

Starts (mounts) an Oracle ASM disk group.

- Syntax and Options
- Example

2.5.10.3.1 Syntax and Options

Use the `srvctl start diskgroup` command with the following syntax:

```
srvctl start diskgroup -diskgroup diskgroup_name
```

Table 2-47 srvctl start diskgroup Options

Option	Description
`-diskgroup` *`diskgroup_name`*	Disk group name

2.5.10.3.2 Example

An example of this command is:

```
srvctl start diskgroup -diskgroup DATA
```

2.5.10.4 srvctl start home

Starts all of the components that are managed by Oracle Restart in the specified Oracle home. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.

This command starts the components that were stopped by a `srvctl stop home`. This command uses the information in the specified state file to identify the components to start.

> **Note:**
>
> Use this command to restart components after you install a patch in an Oracle home.

- Syntax and Options

2.5.10.4.1 Syntax and Options

Use the `srvctl start home` command with the following syntax:

```
srvctl start home -oraclehome oracle_home -statefile state_file
```

Table 2-48 srvctl start home Options

Option	Description
`-oraclehome` *`oracle_home`*	Complete path of the Oracle home
`-statefile` *`state_file`*	Complete path of the state file. The state file contains the current state information for the components in the Oracle home and is created when the `srvctl stop home` command or the `srvctl status home` command is run.

2.5.10.5 srvctl start listener

Starts the specified listener or all listeners.

- Syntax and Options
- Example

2.5.10.5.1 Syntax and Options

Use the `srvctl start listener` command with the following syntax:

```
srvctl start listener [-listener listener_name]
```

Table 2-49 srvctl start listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, all Oracle Restart–managed listeners are started.

2.5.10.5.2 Example

An example of this command is:

```
srvctl start listener -listener listener
```

2.5.10.6 srvctl start ons

Starts Oracle Notification Services (ONS).

- Syntax and Options

2.5.10.6.1 Syntax and Options

Use the `srvctl start ons` command with the following syntax:

```
srvctl start ons [-verbose]
```

Table 2-50 srvctl start ons Options

Option	Description
`-verbose`	Verbose output

2.5.10.7 srvctl start service

Starts the specified database service or services.

- Syntax and Options
- Example

2.5.10.7.1 Syntax and Options

Use the `srvctl start service` command with the following syntax:

```
srvctl start service -db db_unique_name [-service service_name_list |
  -pdb pluggable_database] [-startoption start_options] [-global_override] [-verbose]
```

Table 2-51 srvctl start service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service` *`service_name_list`*	Comma-delimited list of service names. The service name list is optional and, if not provided, SRVCTL starts all of the database's services.
`-pdb` *`pluggable_database`*	In a CDB, the name of the PDB associated with the service If this option is set to an empty string, then the service is associated with root.
`-startoption` *`start_options`*	Options for database startup (for example: `OPEN`, `MOUNT`, `NOMOUNT` and so on) if the database must be started first **See Also:** *SQL*Plus User's Guide and Reference* for more information about startup options
`-global_override`	If the service is a Global Data Services (GDS) service, then this option must be specified to start the service. An error is returned if you attempt to start a GDS service and –`global_override` is not included. This option is ignored if the service is not a GDS service. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.
`-verbose`	Verbose output

2.5.10.7.2 Example

For the database with a `DB_UNIQUE_NAME` of `dbcrm`, the following example starts the `sales` database service:

```
srvctl start service -db dbcrm -service sales
```

2.5.11 status

Displays the running status of the specified component or set of components.

- srvctl status asm
  Displays the running status of the Oracle ASM instance.
- srvctl status database
  Displays the running status of the specified database.
- srvctl status diskgroup
  Displays the running status of an Oracle ASM disk group.
- srvctl status home
  Displays the running status of all of the components that are managed by Oracle Restart in the specified Oracle home. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.
- srvctl status listener
  Displays the running status of the specified listener or of all Oracle Restart–managed listeners.
- srvctl status ons
  Displays the running status of Oracle Notification Services (ONS).
- srvctl status service
  Displays the running status of one or more database services.

2.5.11.1 srvctl status asm

Displays the running status of the Oracle ASM instance.

- Syntax and Options
- Example

2.5.11.1.1 Syntax and Options

Use the `srvctl status asm` command with the following syntax:

```
srvctl status asm [-all] [-verbose]
```

Table 2-52 srvctl status asm Options

Option	Description
`-all`	Display enabled/disabled status also
`-verbose`	Verbose output

2.5.11.1.2 Example

An example of this command is:

```
srvctl status asm

ASM is running on dbhost
```

2.5.11.2 srvctl status database

Displays the running status of the specified database.

- Syntax and Options
- Example

2.5.11.2.1 Syntax and Options

Use the `srvctl status database` command with the following syntax:

```
srvctl status database -db db_unique_name [-force] [-verbose]
```

Table 2-53 srvctl status database Options

Option	Description
`-db db_unique_name`	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-force`	Display a message if the database is disabled
`-verbose`	Verbose output. Lists the database services that are running.

2.5.11.2.2 Example

An example of this command is:

```
srvctl status database -db dbcrm -verbose

Database dbcrm is running with online services mfg,sales
```

2.5.11.3 srvctl status diskgroup

Displays the running status of an Oracle ASM disk group.

- Syntax and Options
- Example

2.5.11.3.1 Syntax and Options

Use the `srvctl status diskgroup` command with the following syntax:

```
srvctl status diskgroup -diskgroup diskgroup_name [-all] [-verbose]
```

Table 2-54   srvctl status diskgroup Options

Option	Description
`-diskgroup` *`diskgroup_name`*	Disk group name
`-all`	Display enabled/disabled status also
`-verbose`	Verbose output. Lists the database services that are running.

2.5.11.3.2 Example

An example of this command is:

```
srvctl status diskgroup -diskgroup DATA

Disk Group DATA is running on dbhost
```

2.5.11.4 srvctl status home

Displays the running status of all of the components that are managed by Oracle Restart in the specified Oracle home. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.

This command writes the current status of the components to the specified state file.

- Syntax and Options

2.5.11.4.1 Syntax and Options

Use the `srvctl status home` command with the following syntax:

```
srvctl status home -oraclehome oracle_home -statefile state_file
```

Table 2-55   srvctl status home Options

Option	Description
`-oraclehome` *`oracle_home`*	Complete path of the Oracle home
`-statefile` *`state_file`*	Complete path of the state file

2.5.11.5 srvctl status listener

Displays the running status of the specified listener or of all Oracle Restart–managed listeners.

- Syntax and Options
- Example

2.5.11.5.1 Syntax and Options

Use the `srvctl status listener` command with the following syntax:

```
srvctl status listener [-listener listener_name] [-verbose]
```

Table 2-56   srvctl status listener Options

Option	Description
`-listener listener_name`	Listener name. If omitted, the status of all listeners is displayed.
`-verbose`	Verbose output. Lists the database services that are running.

2.5.11.5.2 Example

An example of this command is:

```
srvctl status listener -listener crmlistener

Listener CRMLISTENER is running on dbhost
```

2.5.11.6 srvctl status ons

Displays the running status of Oracle Notification Services (ONS).

- Syntax and Options

2.5.11.6.1 Syntax and Options

Use the `srvctl status ons` command with the following syntax:

```
srvctl status ons [-verbose]
```

Table 2-57   srvctl status ons Options

Option	Description
`-verbose`	Verbose output. Lists the database services that are running.

2.5.11.7 srvctl status service

Displays the running status of one or more database services.

- Syntax and Options
- Example

2.5.11.7.1 Syntax and Options

Use the `srvctl status service` command with the following syntax:

```
srvctl status service -db db_unique_name
  [-service service_name_list | -pdb pluggable_database]
  [-force] [-verbose]
```

Table 2-58 srvctl status service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service` *`service_name_list`*	Comma-delimited list of service names. If omitted, status is listed for all database services for the designated database.
`-pdb` *`pluggable_database`*	In a multitenant container database (CDB), the name of the pluggable database (PDB) associated with the service If this option is set to an empty string, then the service is associated with root.
`-force`	Display a message if a service is disabled
`-verbose`	Verbose output

2.5.11.7.2 Example

For the database with the `DB_UNIQUE_NAME` of `dbcrm`, the following example displays the running status of the service `sales`:

```
srvctl status service -db dbcrm -service sales

Service sales is running on dbhost
```

2.5.12 stop

Stops the specified component or components.

If you want a component to remain stopped after you issue a `srvctl stop` command, disable the component. See the disable command.

> **Note:**
>
> If a component is stopped and is not disabled, it could restart as a result of another planned operation. That is, although a stopped component will not restart as a result of a failure, it might be started if a dependent component is started with a `srvctl start` command.

- srvctl stop asm
  Stops the Oracle ASM instance.
- srvctl stop database
  Stops a database and its services.
- srvctl stop diskgroup
  Stops (dismounts) an Oracle ASM disk group.

- srvctl stop home
  Stops all of the components that are managed by Oracle Restart in the specified Oracle home. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.
- srvctl stop listener
  Stops the designated listener or all Oracle Restart–managed listeners. Stopping a listener does not cause databases that are registered with the listener to be stopped.
- srvctl stop ons
  Stops Oracle Notification Services (ONS).
- srvctl stop service
  Stops one or more database services.

> **See Also:**
>
> "Starting and Stopping Components Managed by Oracle Restart"

2.5.12.1 srvctl stop asm

Stops the Oracle ASM instance.

- Syntax and Options
- Example

2.5.12.1.1 Syntax and Options

Use the `srvctl stop asm` command with the following syntax:

```
srvctl stop asm [-stopoption stop_options] [-force]
```

Table 2-59   srvctl stop asm Option

Option	Description
`-stopoption` *`stop_options`*	Options for the shutdown operation, for example, `NORMAL`, `TRANSACTIONAL`, `IMMEDIATE`, or `ABORT` **See Also:** *SQL*Plus User's Guide and Reference* for more information about shutdown options
`-force`	Force. Must be present if disk groups are currently started (mounted). This option enables SRVCTL to stop the disk groups before stopping Oracle ASM. Each dependent database instance is also stopped according to its stop options, or with the `ABORT` option if the configured stop options fail.

2.5.12.1.2 Example

An example of this command is:

```
srvctl stop asm -stopoption abort -force
```

2.5.12.2 srvctl stop database

Stops a database and its services.

- Syntax and Options
- Example

2.5.12.2.1 Syntax and Options

Use the `srvctl stop database` command with the following syntax:

```
srvctl stop database -db db_unique_name [-stopoption stop_options]
  [-drain_timeout timeout] [-force] [-verbose]
```

Table 2-60   srvctl stop database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-stopoption` *`stop_options`*	`SHUTDOWN` command options (for example: `NORMAL`, `TRANSACTIONAL`, `IMMEDIATE`, or `ABORT`). Default is `IMMEDIATE`.
`-drain_timeout` *`timeout`*	This option specifies the time allowed for resource draining to be completed in seconds. Permitted values are `NULL`, `0`, or any positive integer. The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. How draining works depends on the setting of the `-stopoption` option. The default value is `NULL`, which means that this option is not set. If the option is not set, and `-drain_timeout` has been set on the service, then this value is used. If it is set to `0`, then draining does not occur.
`-force`	Stops the database, its services, and any resources that depend on the services
`-verbose`	Verbose output

2.5.12.2.2 Example

An example of this command is:

```
srvctl stop database -db dbcrm
```

2.5.12.3 srvctl stop diskgroup

Stops (dismounts) an Oracle ASM disk group.

- Syntax and Options
- Example

2.5.12.3.1 Syntax and Options

Use the `srvctl stop diskgroup` command with the following syntax:

```
srvctl stop diskgroup -diskgroup diskgroup_name [-force]
```

Table 2-61 srvctl stop diskgroup Options

Option	Description
`-diskgroup` *`diskgroup_name`*	Disk group name
`-force`	Force. Dismount the disk group even if some files in the disk group are open.

2.5.12.3.2 Example

This example stops the disk group named `DATA`. An error is returned if files are open on this disk group.

```
srvctl stop diskgroup -diskgroup DATA
```

2.5.12.4 srvctl stop home

Stops all of the components that are managed by Oracle Restart in the specified Oracle home. The Oracle home can be an Oracle Database home or an Oracle Grid Infrastructure home.

This command identifies the components that it stopped in the specified state file.

> **Note:**
>
> - Before stopping the components in an Oracle Grid Infrastructure home, stop the components in a dependent Oracle Database home.
> - Use this command to stop components before you install a patch in an Oracle home.

- Syntax and Options

2.5.12.4.1 Syntax and Options

Use the `srvctl stop home` command with the following syntax:

```
srvctl stop home -oraclehome oracle_home -statefile state_file
  [-stopoption stop_options] [-force]
```

Table 2-62 srvctl stop home Options

Option	Description
`-oraclehome` *`oracle_home`*	Complete path of the Oracle home
`-statefile` *`state_file`*	Complete path to where you want the state file to be written
`-stopoption` *`stop_options`*	`SHUTDOWN` command options for the database (for example: `NORMAL`, `TRANSACTIONAL`, `IMMEDIATE`, or `ABORT`). Default is `IMMEDIATE`. **See Also:** *SQL*Plus User's Guide and Reference* for more information about shutdown options

Table 2-62 (Cont.) srvctl stop home Options

Option	Description
`-force`	Force stop each component

2.5.12.5 srvctl stop listener

Stops the designated listener or all Oracle Restart–managed listeners. Stopping a listener does not cause databases that are registered with the listener to be stopped.

- Syntax and Options
- Example

2.5.12.5.1 Syntax and Options

Use the `srvctl stop listener` command with the following syntax:

```
srvctl stop listener [-listener listener_name] [-force]
```

Table 2-63 srvctl stop listener Options

Option	Description
`-listener` *`listener_name`*	Listener name. If omitted, all Oracle Restart–managed listeners are stopped.
`-force`	Force. Passes the stop command with the `-f` option to Oracle Clusterware. See *Oracle Clusterware Administration and Deployment Guide* for more information about the Oracle Clusterware `-f` option.

2.5.12.5.2 Example

An example of this command is:

```
srvctl stop listener -listener crmlistener
```

2.5.12.6 srvctl stop ons

Stops Oracle Notification Services (ONS).

- Syntax and Options

2.5.12.6.1 Syntax and Options

Use the `srvctl stop ons` command with the following syntax:

```
srvctl stop ons [-verbose]
```

Table 2-64 srvctl stop ons Options

Option	Description
`-verbose`	Verbose output

2.5.12.7 srvctl stop service

Stops one or more database services.

- Syntax and Options
- Example

2.5.12.7.1 Syntax and Options

Use the `srvctl stop service` command with the following syntax:

```
srvctl stop service -db db_unique_name [-service service_name_list |
  -pdb pluggable_database] [-drain_timeout timeout] [-stopoption stop_option]
  [-global_override] [-wait wait_option] [-force] [-verbose]
```

Table 2-65 srvctl stop service Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-service` *`service_name_list`*	Comma-delimited list of database service names. If you do not provide a service name list, then SRVCTL stops all services on the database
`-pdb` *`pluggable_database`*	In a CDB, the name of the PDB associated with the service If this option is set to an empty string, then the service is associated with root.
`-drain_timeout` *`timeout`*	This option specifies the time allowed for resource draining to be completed in seconds. Permitted values are `NULL`, `0`, or any positive integer. The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. How draining works depends on the setting of the `-stopoption` option. The default value is `NULL`, which means that this option is not set. If the option is not set, and `-drain_timeout` has been set on the service, then this value is used. If it is set to `0`, then draining does not occur.
`-stopoption` *`stop_option`*	This option specifies the mode in which the service is stopped. The following values are permitted: • `IMMEDIATE` specifies that sessions are permitted to drain before the service is stopped. • `TRANSACTIONAL` specifies that sessions are permitted to drain for the amount of time specified in the `–drain_timeout` option. The service is stopped when the time limit is reached, and any remaining sessions are terminated. • `NONE` is the default.

Table 2-65 (Cont.) srvctl stop service Options

Option	Description
`-global_override`	If the service is a Global Data Services (GDS) service, then this option must be specified to stop the service. An error is returned if you attempt to stop a GDS service and `-global_override` is not included. This option is ignored if the service is not a GDS service. See *Oracle Database Global Data Services Concepts and Administration Guide* for more information.
`-wait` *`wait_option`*	This option specifies whether to wait until service draining is completed before stopping the service. Specify `YES` to wait or `NO` to stop the service without waiting.
`-force`	Force. This option disconnects all of the stopped services' sessions immediately. Uncommitted transactions are rolled back. If this option is omitted, active sessions remain connected to the services, but no further connections to the services can be made.
`-verbose`	Verbose output

2.5.12.7.2 Example

The following example stops the `sales` database service on the database with a `DB_UNIQUE_NAME` of `dbcrm`:

```
srvctl stop service -db dbcrm -service sales
```

2.5.13 unsetenv

The `unsetenv` command deletes one or more environment variables from the Oracle Restart configuration for a database, a listener, or an Oracle ASM instance.

To perform `srvctl unsetenv` operations, you must be logged in to the database host computer with the proper user account. See "Preparing to Run SRVCTL" for more information.

- srvctl unsetenv asm
  Removes the specified environment variables from the Oracle Restart configuration for the Oracle ASM instance.
- srvctl unsetenv database
  Removes the specified environment variables from the Oracle Restart configuration for the specified database.
- srvctl unsetenv listener
  Removes the specified environment variables from the Oracle Restart configuration for the specified listener or all listeners.

See Also:

- setenv command
- getenv command
- "Managing Environment Variables in the Oracle Restart Configuration"

2.5.13.1 srvctl unsetenv asm

Removes the specified environment variables from the Oracle Restart configuration for the Oracle ASM instance.

- Syntax and Options
- Example

2.5.13.1.1 Syntax and Options

Use the `srvctl unsetenv asm` command with the following syntax:

```
srvctl unsetenv asm -envs name_list
```

Table 2-66 srvctl unsetenv asm Options

Options	Description
`-envs name_list`	Comma-delimited list of environment variables to remove

2.5.13.1.2 Example

The following example removes the AIX operating system environment variable `AIXTHREAD_SCOPE` from the Oracle ASM instance configuration:

```
srvctl unsetenv asm -envs AIXTHREAD_SCOPE
```

2.5.13.2 srvctl unsetenv database

Removes the specified environment variables from the Oracle Restart configuration for the specified database.

- Syntax and Options
- Example

2.5.13.2.1 Syntax and Options

Use the `srvctl unsetenv database` command as follows:

```
srvctl unsetenv database -db db_unique_name -envs name_list
```

Table 2-67 srvctl unsetenv database Options

Options	Description
`-db db_unique_name`	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-envs name_list`	Comma-delimited list of environment variables to remove

2.5.13.2.2 Example

The following example deletes the `AIXTHREAD_SCOPE` environment variable from the Oracle Restart configuration for the database with a `DB_UNIQUE_NAME` of `dbcrm`:

```
srvctl unsetenv database -db dbcrm -envs AIXTHREAD_SCOPE
```

2.5.13.3 srvctl unsetenv listener

Removes the specified environment variables from the Oracle Restart configuration for the specified listener or all listeners.

- Syntax and Options
- Example

2.5.13.3.1 Syntax and Options

Use the `srvctl unsetenv listener` command with the following syntax:

```
srvctl unsetenv listener [-listener listener_name] -envs name_list
```

Table 2-68 srvctl unsetenv listener Options

Options	Description
`-listener listener_name`	Listener name. If omitted, the specified environment variables are removed from the configurations of all listeners.
`-envs name_list`	Comma-delimited list of environment variables to remove

2.5.13.3.2 Example

The following example removes the AIX operating system environment variable `AIXTHREAD_SCOPE` from the listener configuration for the listener named `crmlistener`:

```
srvctl unsetenv listener -listener crmlistener -envs AIXTHREAD_SCOPE
```

2.5.14 update

The `srvctl update` command updates the running database to switch to the specified startup option.

- srvctl update database
  The `srvctl update database` command changes the open mode of the database.

2.5.14.1 srvctl update database

The `srvctl update database` command changes the open mode of the database.

- Syntax and Options

2.5.14.1.1 Syntax and Options

Use the `srvctl update database` command as follows:

```
srvctl update database -db db_unique_name --startoption start_options
```

Table 2-69 srvctl upgrade database Options

Option	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.
`-startoption` *`start_options`*	Startup options for the database. Examples of startup options are `OPEN`, `MOUNT`, or `"READ ONLY"`.

2.5.15 upgrade

The `srvctl upgrade` command upgrades the resources types and resources from an older version to a newer version.

- srvctl upgrade database
  The `srvctl upgrade database` command upgrades the configuration of a database and all of its services to the version of the database home from where this command is run.

2.5.15.1 srvctl upgrade database

The `srvctl upgrade database` command upgrades the configuration of a database and all of its services to the version of the database home from where this command is run.

- Syntax and Options

2.5.15.1.1 Syntax and Options

Use the `srvctl upgrade database` command as follows:

```
srvctl upgrade database -db db_unique_name -oraclehome oracle_home
```

Table 2-70 srvctl upgrade database Options

Parameter	Description
`-db` *`db_unique_name`*	Unique name for the database. Must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then this option must match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`.

Table 2-70 (Cont.) srvctl upgrade database Options

Parameter	Description
`-oraclehome` *`oracle_home`*	The full path of Oracle home for the database

2.6 CRSCTL Command Reference

You can reference details about the syntax for the CRSCTL commands that are relevant for Oracle Restart.

> **Note:**
>
> You must be the root user or Oracle grid infrastructure software owner to run these CRSCTL commands.

CRSCTL Command Syntax Overview

CRSCTL expects the following command syntax:

```
crsctl command has
```

where *`command`* is a verb such as `start`, `stop`, or `enable`. The `has` object indicates Oracle high availability services.

Case Sensitivity

CRSCTL commands and components are case insensitive.

- check
  Displays the Oracle Restart status.
- config
  Displays the Oracle Restart configuration.
- disable
  Disables automatic restart of Oracle Restart.
- enable
  Enables automatic restart of Oracle Restart.
- start
  Starts Oracle Restart.
- stop
  Stops Oracle Restart.

2.6.1 check

Displays the Oracle Restart status.

Syntax and Options

```
crsctl check has
```

2.6.2 config

Displays the Oracle Restart configuration.

Syntax and Options

```
crsctl config has
```

2.6.3 disable

Disables automatic restart of Oracle Restart.

Syntax and Options

```
crsctl disable has
```

2.6.4 enable

Enables automatic restart of Oracle Restart.

Syntax and Options

```
crsctl enable has
```

2.6.5 start

Starts Oracle Restart.

Syntax and Options

```
crsctl start has
```

2.6.6 stop

Stops Oracle Restart.

Syntax and Options

```
crsctl stop has [-f]
```

Table 2-71 crsctl stop has Options

Options	Description
`-f`	Force. If any resources that are managed by Oracle Restart are still running, then try to stop these resources gracefully. If a resource cannot be stopped gracefully, then try to force the resource to stop. For example, if an Oracle ASM instance is running, then `SHUTDOWN IMMEDIATE` attempts to stop the Oracle ASM instance gracefully, while `SHUTDOWN ABORT` attempts to force the Oracle ASM instance to stop. When the `-f` option *is not* specified, this command tries to stop resources managed by Oracle Restart gracefully but does not try to force them to stop. **Note:** For a database resource, this command always uses `SHUTDOWN ABORT`, regardless of whether the `-f` option is specified.