177

DBMS_SERVICE

The DBMS_SERVICE package lets you create, delete, activate, and deactivate services for a single instance.

The chapter contains the following topics:

- Overview
- Security Model
- Constants
- Operating Procedures
- Exceptions
- Summary of DBMS_SERVICE Subprograms

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for administering services in Oracle Real Application Clusters.

DBMS_SERVICE Overview

DBMS_SERVICE supports the workload management of high availability, quality of service, job scheduling, and other planned operations in the RDBMS for the purposes of workload measurement, management, prioritization, and XA and distributed transaction management.

Oracle Real Application Clusters (RAC) can manage service names across instances as administered through SRVCTL. The DBMS_SERVICE package allows the creation, deletion, starting, and stopping of services in a single instance. Additionally, it provides the ability to disconnect all sessions that connect to an instance with a service name.

> **See Also:**
>
> For more information about Oracle Real Application Clusters, *Oracle Real Application Clusters Administration and Deployment Guide*.

DBMS_SERVICE Security Model

The DBMS_SERVICE package has certain security requirements.

Privileges

The client using this package must have the ALTER SYSTEM execution privilege and the V$SESSION table read privilege.

Schemas

This package must be installed under SYS schema.

Roles

The EXECUTE privilege of the package is granted to the DBA role only.

DBMS_SERVICE Constants

The DBMS_SERVICE package provides constants that can be used for specifying parameter values.

- Constants used in calling arguments are described in Table 177-1
- Constants used in connection balancing goal arguments are described in Table 177-2
- Constants used in TAF failover attribute arguments are described in Table 177-3

Table 177-1   Constants Used in Calling Arguments

Name	Type	Value	Description
GOAL_NONE	NUMBER	0	Disables Load Balancing Advisory
GOAL_SERVICE_TIME	NUMBER	1	Load Balancing Advisory is based on elapsed time for work done in the service plus available bandwidth to the service
GOAL_THROUGHPUT	NUMBER	2	Load Balancing Advisory is based on the rate that work is completed in the service plus available bandwidth to the service

Table 177-2   Constants Used in Connection Balancing Goal Arguments

Name	Type	Value	Description
CLB_GOAL_SHORT	NUMBER	1	Connection load balancing uses Load Balancing Advisory, when Load Balancing Advisory is enabled (either goal_service_time or goal_throughput). When GOAL=NONE (no load balancing advisory), connection load balancing uses an abridged advice based on CPU utilization.

Table 177-2 (Cont.) Constants Used in Connection Balancing Goal Arguments

Name	Type	Value	Description
CLB_GOAL_LONG	NUMBER	2	Balances the number of connections for each instance using session count for each service. This setting is recommended for applications with long connections such as forms. This setting can be used with Load Balancing Advisory when the connection pool is sized to accommodate gravitation within the pool itself (without adding or removing connections). The latter is the most efficient design.

Table 177-3 Constants Used in High Availability Attribute Arguments for FAN, Application Continuity, Transaction Guard and TAF

Name	Type	Value	Description
FAILOVER_METHOD_NONE	VARCHAR2	0	Server side TAF is not enabled for this service
FAILOVER_METHOD_BASIC	VARCHAR2	1	Server side TAF method is BASIC. BASIC is the only value currently supported. This means that a new connection is established at failure time.
FAILOVER_TYPE_NONE	VARCHAR		Server side TAF type is NONE
FAILOVER_TYPE_SESSION	VARCHAR		Server side TAF failover type is SESSION. At failure time, if the failover type is SESSION, TAF reconnects to a surviving node and re-establish a vanilla database session. Customizations (for example, ALTER SESSION) must be re-executed in a failover callback.
FAILOVER_TYPE_SELECT	VARCHAR		Server side TAF failover type is SELECT
FAILOVER_RETRIES	NUMBER		Number of connection attempts when failover occurs. Specifies the number of times for Application Continuity and TAF to attempt the reconnect and re-authenticate pair. The value must be an integer greater than 0. The default in Oracle Database 12c Release 1 (12.1) for Application Continuity is 30.
FAILOVER_RESTORE_NONE	CONSTANT VARCHAR2	NONE	The initial state is not restored before replaying for Application Continuity and TAF. This is recommended for OCI applications that use Application Continuity and build their own state in the request. For example, SQLPLUS.
FAILOVER_RESTORE_BASIC	VARCHAR2	LEVEL1	This is the recommended value for Java and ODP.NET applications using Application Continuity. The initial states that the user knows are restored automatically before replaying. If the user needs additional states, a callback must be registered.

Table 177-3 (Cont.) Constants Used in High Availability Attribute Arguments for FAN, Application Continuity, Transaction Guard and TAF

Name	Type	Value	Description
`FAILOVER_DELAY`	`NUMBER`		Number of seconds delay between each connection attempt. This is the delay that Application Continuity and TAF waits if a reconnect and re-authentication fails. The value must be an integer greater than 0. The default in Oracle Database 12c Release 1 (12.1)is 10s when using Application Continuity. Using `FAILOVER_DELAY` the failover can be delayed until the service is next available. This can work well in conjunction with a planned outage that may make a service temporarily unavailable (such as for several minutes).
`STOP_OPTION_NONE`	`VARCHAR`		Sessions are not disconnected.
`STOP_OPTION_IMMEDIATE`	`VARCHAR`		Sessions are disconnected immediately after the `drain_timeout` expires.
`STOP_OPTION_TRANSACTIONAL`			Sessions are disconnected after the transactions during the `drain_timeout`. The sessions disconnect immediately when `drain_timeout` expires.
`DYNAMIC`	`NUMBER`		For Application Continuity, this parameter specifies whether the session state that is not transactional is changed by the application during request execution. A value of `DYNAMIC` is recommended for all applications. If you are in any doubt, or the application can be customized, you must use `DYNAMIC`.

Usage Notes

- If a TAF callback has been registered, then the failover retries and failover delay are ignored. If an error occurs, TAF continues to re-attempt the connect and authentication as long as the callback returns a value of `OCI_FO_RETRY`. Any delay must be coded into the callback logic
- Server side TAF settings override client-side counterparts that might be configured in TNS connect descriptors. If TAF is not configured on the client side, then at a minimum, the failover type must be set to enable TAF. If the failover type is set on the server side, then the failover method defaults to `BASIC`. Delay and retries are optional and may be specified independently.

DBMS_SERVICE Operating Procedures

You cannot use the following procedures with Oracle Real Applications Clusterware, Oracle Restart, and Oracle Global Data Services.

- – CREATE_SERVICE Procedure
  – DELETE_SERVICE Procedure
  – MODIFY_SERVICE Procedure
  – START_SERVICE Procedure

  - STOP_SERVICE Procedure
- With Oracle Database 12*c* release 1, you are advised to use the parameter interface in all service -related subprorgrams.
- If you wish to use DBMS_SERVICE on a pluggable database (PDB) in a single instance, you must connect to that PDB first.

DBMS_SERVICE Exceptions

This table lists the exceptions raised by the DBMS_SERVICE package.

Table 177-4 DBMS_SERVICE Exceptions

Exception	Error Code	Description
NULL_SERVICE_NAME	44301	Service name argument was found to be NULL
NULL_NETWORK_NAME	44302	Network name argument was found to be NULL
SERVICE_EXISTS	44303	Service name already exists
SERVICE_DOES_NOT_EXIST	44304	Specified service does not exist
SERVICE_IN_USE	44305	Specified service was running
SERVICE_NAME_TOO_LONG	44306	Service name was too long
NETWORK_PREFIX_TOO_LONG	44307	Network name, excluding the domain, was too long
NOT_INITIALIZED	44308	Services layer was not yet initialized
GENERAL_FAILURE	44309	An unknown failure
MAX_SERVICES_EXCEEDED	44310	Maximum number of services has been reached
SERVICE_NOT_RUNNING	44311	Specified service was not running
DATABASE_CLOSED	44312	Database was closed
INVALID_INSTANCE	44313	Instance name argument was not valid
NETWORK_EXISTS	44314	Network name already exists
NULL_ATTRIBUTES	44315	All attributes specified were NULL
INVALID_ARGUMENT	44316	Invalid argument supplied
DATABASE_READONLY	44317	Database is open read-only
MAX_SN_LENGTH	44318	Total length of all running service network names exceeded the maximum allowable length
ERR_AQ_SERVICE	44319	Cannot delete AQ service
ERR_GLB_SERVICE	44320	Cannot delete global service
ERR_INVALID_PDB_NAME	44771	Invalid name for a pluggable database
ERR_CRS_API	44772	Cluster ready services (CRS) operation failed
ERR_PDB_CLOSED	44773	Cannot perform requested service operation

Table 177-4 (Cont.) DBMS_SERVICE Exceptions

Exception	Error Code	Description
`ERR_PDB_INVALID`	44774	Pluggable database attribute cannot be changed
`ERR_PDB_NAME`	44775	Pluggable database service cannot be created
`ERR_PDB_EXP`	44776	Pluggable database service cannot be deleted
`ERR_PDB_FAIL`	44777	Pluggable database service cannot be started

Summary of DBMS_SERVICE Subprograms

This table lists the `DBMS_SERVICE` subprograms and briefly describes them.

Table 177-5 DBMS_SERVICE Package Subprograms

Subprogram	Description
CREATE_SERVICE Procedure	Creates service
DELETE_SERVICE Procedure	Deletes service
DISCONNECT_SESSION Procedure	Disconnects sessions running under this service
MODIFY_SERVICE Procedure	Modifies service
START_SERVICE Procedure	Activates service
STOP_SERVICE Procedure	Stops service

CREATE_SERVICE Procedure

This procedure creates a service name in the data dictionary. Services are also created in the data dictionary implicitly when you set the service in the `service_name` parameter or by means of the `ALTER SYSTEM SET SERVICE_NAMES` command.

> **Note:**
>
> The service attribute values `FAILOVER_TYPE = TRANSACTION` with `SESSION_STATE_CONSISTENCY = STATIC` are no longer a supported service attribute combination.
> In previous releases, you could use the service parameter `SESSION_STATE_CONSISTENCY` to manage session state automatically using Application Continuity by setting `SESSION_STATE_CONSISTENCY` to `DYNAMIC` or `STATIC`. However, starting with Oracle Database 23ai, you can no longer use the `STATIC` option. Instead, use one of the following failover options:
>
> - `FAILOVER_TYPE = AUTO` with `SESSION_STATE_CONSISTENCY = AUTO`
> - `FAILOVER_TYPE = TRANSACTION` with `SESSION_STATE_CONSISTENCY = DYNAMIC`
>
> These configurations enforce session state tracking in Oracle Database, ensuring that session state is preserved at session migration and session failover.

Syntax

```
DBMS_SERVICE.CREATE_SERVICE(
   service_name                IN VARCHAR2,
   network_name                IN VARCHAR2,
   parameter_array             IN TABLE OF VARCHAR2(100));
```

This overload is maintained for backward compatibility:

```
DBMS_SERVICE.CREATE_SERVICE(
   service_name                IN VARCHAR2,
   network_name                IN VARCHAR2,
   goal                        IN NUMBER DEFAULT NULL,
   dtp                         IN BOOLEAN DEFAULT NULL,
   true_cache_service          IN VARCHAR2,
   aq_ha_notifications         IN BOOLEAN DEFAULT NULL,
   failover_method             IN VARCHAR2 DEFAULT NULL,
   failover_type               IN VARCHAR2 DEFAULT NULL,
   failover_retries            IN NUMBER DEFAULT NULL,
   failover_delay              IN NUMBER DEFAULT NULL,
   clb_goal                    IN NUMBER DEFAULT NULL,
   edition                     IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 177-6 CREATE_SERVICE Procedure Parameters

Parameter	Description
`service_name`	Name of the service, limited to 64 characters in the Data Dictionary

Table 177-6 (Cont.) CREATE_SERVICE Procedure Parameters

Parameter	Description
`network_name`	Network name of the service as used in SQLNet connect descriptors for client connections. This is limited to the NET `service_names` character set (see *Oracle Database Net Services Reference*).
`parameter_array`	Associative array with name-value pairs of the service attributes. Supported names: • `goal` • `dtp` • `true_cache_service` • `aq_ha_notifications` • `failover_method` • `failover_type` • `failover_retries` • `failover_restore` • `failover_delay` • `clb_goal` • `edition` • `commit_outcome` • `reset_state` • `retention_timeout` • `replay_initiation_timeout` • `session_state_consistency` • `sql_translation_profile` • `drain_timeout` • `stop_option` • `placement_policy`
`goal`	Workload management goal directive for the service. Valid values: • `DBMS_SERVICE.GOAL_SERVICE_TIME` • `DBMS_SERVICE.GOAL_THROUGHPUT` • `DBMS_SERVICE.GOAL_NONE`
`dtp`	Declares the service to be for X/Open Distributed Transaction Processing (DTP) or any distributed transaction (especially XA)
`true_cache_service`	Name of the True Cache service being registered with the primary service.
`aq_ha_notifications`	Determines whether Fast Application Notification (FAN) is enabled for OCI/OCCI/ODP. In Oracle Database12*c*, FAN uses Oracle Notification Services (ONS). This parameter is still used to enable FAN. FAN is recommended for all High Availability systems, and is on by default for Application Continuity
`failover_method`	Failover `TYPE` for the service for Application Continuity and TAF. If the `failover_type` is set to `TRANSACTION` on the service, this automatically sets `COMMIT_OUTCOME` to `TRUE`. JDBC Replay Driver uses the `FAILOVER_TYPE` service attribute setting of `TRANSACTION` for `TRANSACTION` failover. OCI uses the older settings of `SELECT` and `SESSION`. The server only accepts `FAILOVER_METHOD` = `BASIC` with the `TRANSACTION` setting.
`failover_type`	`TRANSACTION` for Application Continuity. `SELECT` or `BASIC` for TAF.

Table 177-6 (Cont.) CREATE_SERVICE Procedure Parameters

Parameter	Description
`failover_retries`	Number of connection retries for Application Continuity and TAF. Using the `failover_retries` and `failover_delay` parameters, the failover can be delayed until the service is next available. This parameter is for connecting. It does not control the number of failovers, which is 3 for each incident for Application Continuity.
`failover_delay`	Delay in seconds between connection retries for Application Continuity and TAF. The default is 10 seconds for Application Continuity. Do not use a 0-second delay if the service needs time to failover and register. Long delays are good for planned outages and to failover to Data Guard. Short delays work well with RAC when the service is already available.
`edition`	If this argument has a non-`NULL` value, this provides the initial session edition for subsequent database connections using this service that do not specify an edition. If no value is specified, this argument has no effect. During service creation or modification, no validation is performed on this parameter. At connection time, if the connecting user does not have `USE` privilege on the edition, or the edition does not exist, this raises the error `ORA-38802` (edition does not exist).
`drain_timeout`	If this parameter is defined, all sessions connected to that service are drained by the client drivers and pools using Fast Connection Failover (FCF). The `drain_timeout` can be set on the service, so stopping and relocating drains for this time by default.
`stop_option`	Provides options to terminate a session. The supported values are: • `TRANSACTIONAL` `drain_timeout` applies to the transactions. After the `drain_timeout` expire, the sessions are immediately killed. • `IMMEDIATE` The sessions are killed after `drain_timeout` expires. • `NONE` Sessions are not terminated. However, these parameter values can be overridden using the command line.
`commit_outcome`	Determines whether transaction `COMMIT` outcome is accessible after the `COMMIT` has executed. While the database guarantees that `COMMIT` is durable, this ensures that the outcome of the `COMMIT` is durable. Applications use the feature to probe the status of the commit last executed after an outage, and is available to applications to determine an outcome. Note: • Invoking the GET_LTXID_OUTCOME Procedure of the DBMS_APP_CONT package requires that the `commit_outcome` attribute be set. • `commit_outcome` has no effect on active Data Guard and read-only databases. • `commit_outcome` is only allowed on the database service and on user-defined database services
`reset_state`	Clears the session state usage between requests, so that each new request starts clean (usage web and stateless applications).

Table 177-6 (Cont.) CREATE_SERVICE Procedure Parameters

Parameter	Description
`retention_timeout`	Used in conjunction with `commit_outcome`, it determines the amount of time (in seconds) that the `COMMIT_OUTCOME` is retained. Default is 24 hours (86400). Maximum value is 30 days (2592000).
`replay_initiation_timeout`	For Application Continuity, `replay_initiation_timeout` is the difference between the time of original execution of first operation of a request, and the time that the replay is ready to start after a successful reconnect. Replay initiation time is measured from the time that the request was originally submitted until the time that replay has connected and is ready to replay. When replay is expected, keep this value high. Default is 900 seconds.
`session_state_consistency`	Describes how nontransactional is changed during a request (`DYNAMIC`). This parameter is considered only if `failover_type` is set to `TRANSACTION` for Application Continuity. Examples of session state are NLS settings, optimizer preferences, event settings, PL/SQL global variables, temporary tables, advanced queues, LOBs, and result cache. If these values change after the request starts, set to `DYNAMIC` (default). Almost all applications should use `DYNAMIC` mode. If you are unsure, use `DYNAMIC` mode.
`sql_translation_profile`	Name of SQL translation profile.
`clb_goal`	Method used for Connection Load Balancing (see Table 177-2)
`placement_policy`	`NUMBER` Placement policy for the service. Possible values: • `0: PDB-NONE` • `1: PDB-SINGLETON` • `2: PDB-UNIFORM` **Note:** Values other than `0` are applicable only in the ATP-Dedicated Cloud in an Oracle RAC environment.

Examples

```
DBMS_SERVICE.CREATE_SERVICE('ernie.example.com','ernie.example.com');

DECLARE
   params dbms_service.svc_parameter_array;
   BEGIN
      params('TRUE_CACHE_SERVICE')        :='C_SALES';
      params('FAILOVER_TYPE')             :='TRANSACTION';
      params('REPLAY_INITIATION_TIMEOUT'):=1800;
      params('RETENTION_TIMEOUT')         :=86400;
      params('FAILOVER_DELAY')            :=10;
      params('FAILOVER_RETRIES')          :=30;
      params('DRAIN_TIMEOUT')             :=60;
      params('STOP_OPTION')               :='DBMS_SERVICE.STOP_OPTION_IMMEDIATE';
      params('FAILOVER_RESTORE')          :='DBMS_SERVICE.FAILOVER_RESTORE_BASIC';
      params('commit_outcome')            :='true';
      params('aq_ha_notifications')      :='true';
      DBMS_SERVICE.MODIFY_SERVICE('GOLD',params);
   END;
```

DELETE_SERVICE Procedure

This procedure deletes a service from the data dictionary.

> **Note:**
>
> Starting with Oracle Database 19c, customer use of the `SERVICE_NAME` parameter is deprecated. It can be desupported in a future release. It must not be used for high availability (HA) deployments. It is not supported to use service name parameter for any HA operations. This restriction includes FAN, load balancing, `FAILOVER_TYPE`, `FAILOVER_RESTORE`, `SESSION_STATE_CONSISTENCY`, and any other uses.
>
> You cannot use this subprogram if your services are managed by Oracle Clusterware, Oracle Restart, or Oracle Global Data Services.

Syntax

```
DBMS_SERVICE.DELETE_SERVICE(
   service_name   IN VARCHAR2);
```

Parameters

Table 177-7 DELETE_SERVICE Procedure Parameters

Parameter	Description
`service_name`	Name of the service, limited to 64 characters in the Data Dictionary

Examples

```
DBMS_SERVICE.DELETE_SERVICE('ernie.example.com');
```

DISCONNECT_SESSION Procedure

This procedure disconnects sessions with the named service at the current instance.

Syntax

```
DBMS_SERVICE.DISCONNECT_SESSION(
   service_name          IN VARCHAR2,
   disconnect_option     IN NUMBER DEFAULT POST_TRANSACTION;
```

Parameters

Table 177-8 DISCONNECT_SESSION Procedure Parameters

Parameter	Description
`service_name`	Name of the service, limited to 64 characters in the Data Dictionary

Table 177-8 (Cont.) DISCONNECT_SESSION Procedure Parameters

Parameter	Description
disconnect_option	The options, package constants, are expressed as NUMBER: • POST_TRANSACTION = 0: session disconnects after the current transaction commits or rolls back • IMMEDIATE = 1: session disconnects immediately • NOREPLAY = 2: session disconnects immediately and be flagged to not be replayed by application continuity, that is IMMEDIATE and NOREPLAY together **Note**: IMMEDIATE or POST_TRANSACTION and NOREPLAY is automatically translated as 1 or 0 or 2 respectively. However, passing a string literal (quoted using either the ' or " characters, such as "IMMEDIATE" or 'POST_TRANSACTION' or 'NOREPLAY') raises an error.

Usage Notes

- This procedure can be used in the context of a single instance as well as with Oracle Real Application Clusters.
- This subprogram does not return until all corresponding sessions are disconnected. Therefore, use the DBMS_JOB package or put the SQL session in background if the caller does not want to wait for all corresponding sessions to be disconnected.

Examples

This disconnects sessions with service_name 'ernie.example.com'.

```
DBMS_SERVICE.DISCONNECT_SESSION('ernie.example.com');
```

If a service is using application continuity, and you do not want the sessions replayed but simply terminated, use the following:

```
EXECUTE DBMS_SERVICE.DISCONNECT_SESSION('service name', DBMS_SERVICE.NOREPLAY);
```

MODIFY_SERVICE Procedure

This procedure modifies an existing service.

> **Note:**
>
> You cannot use the second version of subprogram if your services are managed by Oracle Clusterware, Oracle Restart, or Oracle Global Data Services. The version with the parameter array interface applies to databases that are not managed by Oracle Clusterware, Oracle Restart or Oracle Global Data Services. New attributes are only available using the parameter interface.

Syntax

```
DBMS_SERVICE.MODIFY_SERVICE(
   service_name            IN VARCHAR2,
   parameter_array         IN svc_parameter_array);
```

This overload is maintained for backward compatibility:

```
DBMS_SERVICE.MODIFY_SERVICE(
   service_name              IN VARCHAR2,
   goal                      IN NUMBER DEFAULT NULL,
   dtp                       IN BOOLEAN DEFAULT NULL,
   true_cache_service        IN VARCHAR2,
   aq_ha_notifications       IN BOOLEAN DEFAULT NULL,
   failover_method           IN VARCHAR2 DEFAULT NULL,
   failover_type             IN VARCHAR2 DEFAULT NULL,
   failover_retries          IN NUMBER DEFAULT NULL,
   failover_delay            IN NUMBER DEFAULT NULL,
   clb_goal                  IN NUMBER DEFAULT NULL,
   edition                   IN VARCHAR2 DEFAULT NULL,
   modify_edition            IN BOOLEAN DEFAULT FALSE;
```

Parameters

Table 177-9   MODIFY_SERVICE Procedure Parameters

Parameter	Description
service_name	Name of the service, limited to 64 characters in the Data Dictionary
parameter_array	Associative array with name/value pairs of the service attributes. Supported names: • goal • dtp • true_cache_service • aq_ha_notifications • failover_method • failover_type • failover_restore • failover_retries • failover_delay • drain_timeout • stop_option • edition • commit_outcome • retention_timeout • replay_initiation_timeout • session_state_consistency • sql_translation_profile • placement_policy
goal	Workload management goal directive for the service. Valid values: • DBMS_SERVICE.GOAL_SERVICE_TIME • DBMS_SERVICE.GOAL_THROUGHPUT • DBMS_SERVICE.GOAL_NONE
dtp	Declares the service to be for X/Open Distributed Transaction Processing (DTP) or any distributed transaction (especially XA)
true_cache_service	Name of the True Cache service being registered with the primary service.

Table 177-9 (Cont.) MODIFY_SERVICE Procedure Parameters

Parameter	Description
aq_ha_notifications	Determines whether Fast Application Notification (FAN) is enabled for OCI/OCCI/ODP. In Oracle Database12*c*, FAN uses Oracle Notification Services (ONS). This parameter is still used to enable FAN. FAN is recommended for all High Availability systems, and is on by default for Application Continuity
failover_method	Failover TYPE for the service for Application Continuity and TAF. If the failover_type is set to TRANSACTION on the service, this automatically sets COMMIT_OUTCOME to TRUE. JDBC Replay Driver uses the FAILOVER_TYPE service attribute setting of TRANSACTION for TRANSACTION failover. OCI uses the older settings of SELECT and SESSION. The server only accepts FAILOVER_METHOD = BASIC with the TRANSACTION setting.
failover_type	Failover TYPE for the service for Application Continuity and TAF.
failover_restore	For Application Continuity, when the failover_restore parameter is set, the session states are restored before replaying for ODP.NET and Java. Use LEVEL1 for ODP.NET and Java with Application Continuity to restore the initial state. For AC OCI, use NONE for applications that are not STATIC.
failover_retries	Number of connection retries for Application Continuity and TAF. Using the failover_retries and failover_delay parameters, the failover can be delayed until the service is next available. This parameter is for connecting. It does not control the number of failovers, which is 3 for each incident for Application Continuity.
failover_delay	Delay in seconds between connection retries for Application Continuity and TAF. The default is 10 seconds for Application Continuity. Do not use a 0-second delay if the service needs time to failover and register. Long delays are good for planned outages and to failover to Data Guard. Short delays work well with Oracle RAC when the service is already available.
drain_timeout	When this parameter is set, all sessions connected to that service are drained by the client drivers and pools using Fast Connection Failover (FCF). The drain_timeout can be set on the service, to stop and relocate drains for this time by default.
edition	If this argument has a non-NULL value, this provides the initial session edition for subsequent database connections using this service that do not specify an edition. If no value is specified, this argument has no effect. During service creation or modification, no validation is performed on this parameter. At connection time, if the connecting user does not have USE privilege on the edition, or the edition does not exist, this raises the error ORA-38802 (edition does not exist).

Table 177-9 (Cont.) MODIFY_SERVICE Procedure Parameters

Parameter	Description
commit_outcome	Determines whether transaction COMMIT outcome is accessible after the COMMIT has executed. While the database guarantees that COMMIT is durable, this ensures that the outcome of the COMMIT is durable.Applications use the feature to probe the status of the commit last executed after an outage, and is available to applications to determine an outcome. Note: • Invoking the GET_LTXID_OUTCOME Procedure of the DBMS_APP_CONT package requires that the commit_outcome attribute be set. • commit_outcome has no effect on active Data Guard and read-only databases. • commit_outcome is allowed only on user-defined database services
retention_timeout	Used in conjunction with commit_outcome, it determines the amount of time (in seconds) that the COMMIT_OUTCOME is retained. Default is 24 hours (86400). Maximum value is 30 days (2592000).
replay_initiation_timeout	For Application Continuity, replay_initiation_timeout is the difference between the time of original execution of first operation of a request, and the time that the replay is ready to start after a successful reconnect. Replay initiation time is measured from the time that the request was originally submitted until the time that replay has connected and is ready to replay. When replay is expected, keep this value high. Default is 900 seconds.
session_state_consistency	Describes how nontransactional is changed during a request. This parameter is considered only if failover_type is set to TRANSACTION for Application Continuity. Examples of session state are NLS settings, optimizer preferences, event settings, PL/SQL global variables, temporary tables, advanced queues, LOBs, and result cache. If these values change after the request starts, set to DYNAMIC (default). Almost all applications should use DYNAMIC mode. If you are unsure, use DYNAMIC mode.
sql_translation_profile	Name of SQL translation profile.
modify_edition	If TRUE, the edition service attribute is updated to use the edition argument value. If FALSE or NULL, the edition attribute is not updated.
clb_goal	Method used for Connection Load Balancing (see Table 177-2)
placement_policy	NUMBER Placement policy for the service. Possible values: • 0: PDB-NONE • 1: PDB-SINGLETON • 2: PDB-UNIFORM **Note:** Values other than 0 are applicable only in the ATP-Dedicated Cloud in an Oracle RAC environment.

Usage Notes

- If you are using Clustered Managed Services with Oracle Clusterware, or using Oracle Restart with your single instance database, you must modify services using the `srvctl` command rather than `DBMS_SERVICE`. When the service is started by Oracle Clusterware or Oracle Restart, the service is modified in the database to match the resource defined to either Oracle Clusterware or Oracle Restart. Any changes made with `DBMS_SERVICE` are lost unless they are also made with the corresponding `srvctl` command. Starting with 11.2.0.2, service attribute modifications take effect immediately when the service is started or modified by `srvctl`.
- Although users can modify the edition attribute while the service is up and running, it may not be safe to do so. Users must proceed with caution because this causes new connections to be connected at the new edition, while the existing connection is not affected. This can cause mid-tier operations to connect to the wrong edition.

START_SERVICE Procedure

This procedure starts a service. In Oracle RAC, implementing this option acts on the instance specified.

> **Note:**
>
> You cannot use this subprogram if your services are managed by Oracle Clusterware, Oracle Restart or Oracle Global Data Services.

Syntax

```
DBMS_SERVICE.START_SERVICE(
   service_name  IN VARCHAR2,
   instance_name IN VARCHAR2);
```

Parameters

Table 177-10 *START_SERVICE Procedure Parameters*

Parameter	Description
`service_name`	Name of the service limited to 64 characters in the Data Dictionary
`instance_name`	Name of the instance where the service must be activated (optional). `NULL` results in starting of the service on the local instance. In single instance, this can only be the current instance or `NULL`. Specify `DBMS_SERVICE.ALL_INSTANCES` to start the service on all configured instances.

Examples

```
DBMS_SERVICE.START_SERVICE('ernie.example.com');
```

STOP_SERVICE Procedure

This procedure stops a service.

> **Note:**
>
> You cannot use this subprogram if your services are managed by Oracle Clusterware, Oracle Restart or Oracle Global Data Services.

Syntax

```
DBMS_SERVICE.STOP_SERVICE(
   service_name   IN VARCHAR2,
   instance_name  IN VARCHAR2  DEFAULT NULL,
   stop_option    IN VARCHAR2  DEFAULT NULL,
   drain_timeout  IN NUMBER    DEFAULT NULL,
   replay         IN BOOLEAN   DEFAULT TRUE);
```

Parameters

Table 177-11   STOP_SERVICE Procedure Parameters

Parameter	Description
`service_name`	Name of the service limited to 64 characters in the Data Dictionary
`instance_name`	Name of the instance where the service must be stopped (optional). `NULL` results in stopping of the service locally. In single instance, this can only be the current instance or `NULL`. The default in Oracle RAC and exclusive case is `NULL`. Specify `DBMS_SERVICE.ALL_INSTANCES` to stop the service on all configured instances.
`stop_option`	To specify how sessions are stopped with draining. The possible values are as follows: • `IMMEDIATE`: sessions are aborted immediately after the time specified in `drain_timeout`. • `TRANASCTIONAL`: applies for transactions. After the transaction expires, the sessions are immediately terminated. • `NONE`: sessions are not terminated. These values can be overridden on the command line using `SRVCTL`.
`drain_timeout`	The time in seconds for the session to drain.
`replay`	Enable application continuity replay.

Examples

```
DBMS_SERVICE.STOP_SERVICE('ernie.example.com');
```