54
DBMS_CONNECTION_POOL

The DBMS_CONNECTION_POOL package provides an interface to manage the Database Resident Connection Pools (DRCP).

The database initialization parameter ENABLE_PER_PDB_DRCP controls whether DRCP is configured in the CDB ROOT DRCP mode or per-PDB DRCP mode. The default value is FALSE, which will configure DRCP at the CDB level.

- If the value of the database configuration parameter ENABLE_PER_PDB_DRCP is set to TRUE, then:
  - Each PDB administrator can manage their pool configuration using the DBMS_CONNECTION_POOL package. If the ROOT tries to manage the pool configuration using the DBMS_CONNECTION_POOL package, then an error is thrown.
  - The values of the num_cbrok and maxconn_cbrok pool parameters from the cpool$ table are ignored. The PDB administrator cannot modify these parameters using the DBMS_CONNECTION_POOL.ALTER_PARAM() procedure. These parameters can be set using the CONNECTION_BROKERS database parameter. Only the ROOT can alter these parameters dynamically.
- If the value of the ENABLE_PER_PDB_DRCP parameter is set to FALSE, then only the ROOT can manage the pool configuration, using the DBMS_CONNECTION_POOL package. If a PDB administrator tries to manage the pool configuration using the DBMS_CONNECTION_POOL package, then an error is thrown.
- The following DRCP parameters can have values ranging from 0 to SB4MAXVAL (2147483647) - 1:
  - minsize
  - num_cbrok
  - maxconn_cbrok

> **See Also:**
>
> *Oracle Database Concepts* for more information on "Database Resident Connection Pooling"

This chapter contains the following topic:

- Summary of DBMS_CONNECTION_POOL Subprograms

Summary of DBMS_CONNECTION_POOL Subprograms

This table lists the subprograms of the DBMS_CONNECTION_POOL package in an alphabetical order and describes them briefly.

Table 54-1 DBMS_CONNECTION_POOL Package Subprograms

Subprogram	Description
ADD_POOL Procedure	Adds a new pool to the multiple pool DRCP.
ALTER_PARAM Procedure	Alters a specific configuration parameter as a standalone unit, without affecting the other parameters.
CONFIGURE_POOL Procedure	Configures the pool with advanced options.
REMOVE_POOL Procedure	Removes a pool from the multiple pool DRCP.
RESTORE_DEFAULTS Procedure	Restores the pool to the default settings
START_POOL Procedure	Starts the pool for operations. Only after this procedure is called, the pool can be used by the connection clients for creating sessions .
STOP_POOL Procedure	Stops the pool and makes it unavailable for the registered connection clients.

ADD_POOL Procedure

You can use this procedure to add a new DRCP pool.

Syntax

```
DBMS_CONNECTION_POOL.ADD_POOL (
   pool_name                IN VARCHAR2,
   minsize                  IN PLS_INTEGER   DEFAULT 0,
   maxsize                  IN PLS_INTEGER   DEFAULT 40,
   incrsize                 IN PLS_INTEGER   DEFAULT 2,
   session_cached_cursors   IN PLS_INTEGER   DEFAULT 20,
   inactivity_timeout       IN PLS_INTEGER   DEFAULT 300,
   max_think_time           IN PLS_INTEGER   DEFAULT 120,
   max_use_session          IN PLS_INTEGER   DEFAULT 500000,
   max_lifetime_session     IN PLS_INTEGER   DEFAULT 86400,
   max_txn_think_time       IN PLS_INTEGER   DEFAULT 0);
```

Parameters

> **Note:**
>
> If you are aware of the pool configuration at the time of adding the pool, then you can specify the values for the following parameters. Otherwise, you can call the `CONFIGURE_POOL` subprogram later and specify the new pool name and the configuration options.

Table 54-2 ADD_POOL Procedure Parameters

Parameter	Description
`pool_name`	The name of the pool to be added to the DRCP.
`minsize`	The minimum number of pooled servers in the pool. The default value is 0.
`maxsize`	The maximum number of pooled servers allowed in the pool. The default value is 40.

Table 54-2 (Cont.) ADD_POOL Procedure Parameters

Parameter	Description
incrsize	Pool would increment by this number of pooled server when pooled servers are unavailable at application request time. The default value is 2.
session_cached_cursors	The number of session cursors to cache in each pooled server session. The default value is 20. Turn on SESSION_CACHED_CURSORS for all connections in the pool. This is an existing init.ora parameter.
inactivity_timeout	TTL (Time to live) for an idle session in the pool. This parameter helps to shrink the pool when it is not used to its maximum capacity. If a connection remains in the pool idle for this time, the connection is closed. The default value is 300.
max_think_time	The maximum time of inactivity, in seconds, for a client after it obtains a pooled server from the pool with no open transactions in it. After obtaining a pooled server from the pool, if the client application does not issue a database call for the time specified by MAX_THINK_TIME, the pooled server is freed and the client connection is terminated. The default value is 120.
max_use_session	Maximum number of times a connection can be taken and released to the pool. The default value is 500000.
max_lifetime_session	TTL (Time to live) in seconds for a pooled session. The default value is 86400.
max_txn_think_time	The maximum time of inactivity, in seconds, for a client after it obtains a pooled server from the pool with an open transaction. After obtaining the pooled server from the pool, if the client application does not issue a database call for the time specified by MAX_TXN_THINK_TIME, then the pooled server is freed, and the client connection is terminated. The default value of this parameter is the value of the MAX_THINK_TIME parameter. Applications can set the value of the MAX_TXN_THINK_TIME parameter to a value higher than the MAX_THINK_TIME value to allow more time for the connections with open transactions.

Note:

The pool_name parameter has the following validation checks:

- The pool name must be unique.
- It must begin with an alphabetic character.
- It can contain alphanumeric characters in the second and subsequent positions.
- It can contain an underscore (_) in the second and subsequent positions.
- The maximum length allowed for the pool name is 128.

> **See Also:**
>
> For the list and description of all the database resident connection pooling parameters that can be configured using this procedure, see the *Oracle Database Administrator's Guide*.

Examples

```
exec dbms_connection_pool.add_pool('mypool')
```

ALTER_PARAM Procedure

This procedure alters a specific Database Resident Connection Pool (DRCP) configuration parameter as a standalone unit, and does not affect other parameters.

Syntax

```
DBMS_CONNECTION_POOL.ALTER_PARAM (
   pool_name     IN  VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL',
   param_name    IN  VARCHAR2,
   param_value   IN  VARCHAR2);
```

Parameters

Table 54-3 ALTER_PARAM Procedure Parameters

Parameter	Description
`pool_name`	Pool to be configured.
`param_name`	Any parameter name from `CONFIGURE_POOL`. For broker parameters: `NUM_CBROK` and `MAXCONN_CBROK`, these rules apply: • If per-PDB DRCP is enabled, then `ALTER_PARAM()` cannot be used to set these parameters. Using the database initialization parameter `connection_brokers` is the only way to set them. • For the CDB ROOT-level DRCP, if these parameters are not set using `connection_brokers`, then the administrator can use the `ALTER_PARAM()` procedure to set them. Using `connection_brokers` is the recommended way to set these parameters.
`param_value`	Parameter value for `param_name`.

> **See Also:**
>
> For the list and description of all the DRCP parameters that can be configured using this procedure, see the *Oracle Database Administrator's Guide*.

Exceptions

Table 54-4 ALTER_PARAM Procedure Exceptions

Exception	Description
ORA-56500	Connection pool not found
ORA-56504	Invalid connection pool configuration parameter name
ORA-56505	Invalid connection pool configuration parameter value
ORA-56507	Connection pool alter configuration failed

Examples

```
DBMS_CONNECTION_POOL.ALTER_PARAM(
   'SYS_DEFAULT_CONNECTION_POOL', 'MAX_LIFETIME_SESSION', '120');
```

CONFIGURE_POOL Procedure

This procedure configures the pool with advanced options.

Syntax

```
DBMS_CONNECTION_POOL.CONFIGURE_POOL (
   pool_name                IN VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL',
   minsize                  IN NUMBER   DEFAULT 4,
   maxsize                  IN NUMBER   DEFAULT 40,
   incrsize                 IN NUMBER   DEFAULT 2,
   session_cached_cursors   IN NUMBER   DEFAULT 20,
   inactivity_timeout       IN NUMBER   DEFAULT 300,
   max_think_time           IN NUMBER   DEFAULT 120,
   max_use_session          IN NUMBER   DEFAULT 500000,
   max_lifetime_session     IN NUMBER   DEFAULT 86400,
   max_txn_think_time       IN NUMBER);
```

Parameters

Table 54-5 CONFIGURE_POOL Procedure Parameters

Parameter	Description
pool_name	Pool to be configured.
minsize	Minimum number of pooled servers in the pool
maxsize	Maximum allowed pooled servers in the pool
incrsize	Pool would increment by this number of pooled server when pooled server are unavailable at application request time
session_cached_cursors	Turn on SESSION_CACHED_CURSORS for all connections in the pool. This is an existing init.ora parameter
inactivity_timeout	TTL (Time to live) for an idle session in the pool. This parameter helps to shrink the pool when it is not used to its maximum capacity. If a connection remains in the pool idle for this time, it is killed.

Table 54-5 (Cont.) CONFIGURE_POOL Procedure Parameters

Parameter	Description
max_think_time	The maximum time of inactivity, in seconds, for a client after it obtains a pooled server from the pool with no open transactions in it. After obtaining a pooled server from the pool, if the client application does not issue a database call for the time specified by MAX_THINK_TIME, the pooled server is freed and the client connection is terminated.
max_use_session	Maximum number of times a connection can be taken and released to the pool
max_lifetime_session	TTL (Time to live) for a pooled session
max_txn_think_time	The maximum time of inactivity, in seconds, for a client after it obtains a pooled server from the pool with an open transaction. After obtaining the pooled server from the pool, if the client application does not issue a database call for the time specified by MAX_TXN_THINK_TIME, then the pooled server is freed, and the client connection is terminated. The default value of this parameter is the value of the MAX_THINK_TIME parameter. Applications can set the value of the MAX_TXN_THINK_TIME parameter to a value higher than the MAX_THINK_TIME value to allow more time for the connections with open transactions.

Exceptions

Table 54-6 CONFIGURE_POOL Procedure Exceptions

Exception	Description
ORA-56500	Connection pool not found
ORA-56507	Connection pool alter configuration failed

Usage Notes

- All expressions of time are in seconds
- All of the parameters should be set based on statistical request patterns.
- minsize should be set keeping in mind that it puts a lower bound on server resource consumption. This is to prevent the timeout from dragging the pool too low, because of a brief period of inactivity.
- maxsize should be set keeping in mind that it puts an upper bound on concurrency and response-times and also server resource consumption.
- session_cached_cursors is typically set to the number of most frequently used statements. It occupies cursor resource on the server
- In doubt, do not set the increment and inactivity_timeout. The pool will have reasonable defaults.
- max_use_session and max_lifetime_session allow for software rejuvenation or defensive approaches to potential bugs, leaks, accumulations, and like problems, by getting brand new sessions once in a while.

- The connection pool reserves 5% of the pooled servers for authentication, and at least one pooled server is always reserved for authentication. When setting the `maxsize` parameter, ensure that there are enough pooled servers for both authentication and connections.

REMOVE_POOL Procedure

You can use this procedure to remove a pool from DRCP.

Syntax

```
DBMS_CONNECTION_POOL.REMOVE_POOL (
   pool_name      IN  VARCHAR2);
```

Parameters

Table 54-7 REMOVE_POOL Procedure Parameters

Parameter	Description
`pool_name`	The pool to be removed.

> **Note:**
>
> This procedure generates an error if:
>
> - The specified pool does not exist.
> - You try to remove the default pool.
> - You try to remove an active pool without stopping it.

Exceptions

Table 54-8 REMOVE_POOL Procedure Exceptions

Exception	Description
`ORA-56620`	Removing the pool <pool_name> failed because pool doesn't exist.
`ORA-56620`	Removing the pool <pool_name> failed because database is read only.
`ORA-56620`	Removing the pool <pool_name> failed because pool is active.
`ORA-56620`	Removing the pool <pool_name==NULL> failed because pool name cannot be empty.
`ORA-56620`	Removing the pool `SYS_DEFAULT_CONNECTION_POOL` failed because default pool cannot be removed.

> **Note:**
>
> The same error number is used for all exception errors because the error message is a dynamic string.

Examples

```
exec dbms_connection_pool.remove_pool('mypool')
```

RESTORE_DEFAULTS Procedure

This procedure restores the pool to default settings.

Syntax

```
DBMS_CONNECTION_POOL.RESTORE_DEFAULTS (
   pool_name    IN  VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL');
```

Parameters

Table 54-9 RESTORE_DEFAULTS Procedure Parameters

Parameter	Description
pool_name	Pool to be restored.

Exceptions

Table 54-10 RESTORE_DEFAULTS Procedure Exceptions

Exception	Description
ORA-56500	Connection pool not found
ORA-56507	Connection pool alter configuration failed

START_POOL Procedure

This procedure starts the pool for operations. It is only after this call that the pool could be used by connection classes for creating sessions.

Syntax

```
DBMS_CONNECTION_POOL.START_POOL (
   pool_name  IN  VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL');
```

Parameters

Table 54-11 START_POOL Procedure Parameters

Parameter	Description
pool_name	Pool to be started.

Exceptions

Table 54-12 START_POOL Procedure Exceptions

Exception	Description
ORA-56500	Connection pool not found
ORA-56501	Connection pool startup failed

Usage Notes

If the instance is restarted (shutdown followed by startup), DRCP pools are automatically started.

STOP_POOL Procedure

This procedure stops the pool and makes it unavailable for the registered connection classes.

Syntax

```
DBMS_CONNECTION_POOL.STOP_POOL (
   pool_name    IN   VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL',
   draintime    IN   PLS_INTEGER DEFAULT 2147483647);
```

Parameters

Table 54-13 STOP_POOL Procedure Parameters

Parameter	Description
pool_name	Pool to be stopped.
draintime	Allows active DRCP pools to be closed after a specified connection drain time, or be closed immediately without waiting for the connections to be idle. draintime can have the following integer values: • 0 to close the bound pooled servers immediately. • x to close the bound pooled servers after x seconds.

> **Note:**
>
> Do not pass the draintime parameter if you want to retain the old behavior of closing the pools only after they become idle and inactive.

Exceptions

Table 54-14 STOP_POOL Procedure Exceptions

Exception	Description
ORA-56500	Connection pool not found
ORA-56506	Connection pool shutdown failed

Usage Notes

This stops the pool and takes it offline. This does not destroy the persistent data (such as, the pool name and configuration parameters) associated with the pool.