D

Advanced Queuing Sharded Queues

Table D-1 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces for Sharded Queues: Administrative Interface

Use Case	PL/SQL	Java (JMS)
Create a sharded queue	`DBMS_AQADM.CREATE_SHARDED_QUEUE`	`AQjmsDestination.createJMSShardedQueue`
Drop a sharded queue	`DBMS_AQADM.DROP_SHARDED_QUEUE`	`AQjmsDestination.dropJMSShardedQueue`
Alter a sharded queue	`DBMS_AQADM.ALTER_SHARDED_QUEUE`	None. Use PL/SQL API.

Managing Sharded Queues

These topics describe how to manage sharded queues.

> **Note:**
>
> Starting and stopping a sharded queue use the same APIs as non-sharded queues.

- Creating a Sharded Queue
- Dropping a Sharded Queue
- Altering a Sharded Queue
- Setting a Queue Parameter
- Unsetting a Queue Parameter
- Getting a Queue Parameter
- Creating an Exception Queue

Creating a Sharded Queue

The `CREATE_SHARDED_QUEUE` API creates a sharded queue.

```
PROCEDURE CREATE_SHARDED_QUEUE (
    queue_name             IN VARCHAR2,
    storage_clause         IN VARCHAR2       DEFAULT NULL,
    multiple_consumers     IN BOOLEAN        DEFAULT FALSE,
    max_retries            IN NUMBER         DEFAULT NULL,
    comment                IN VARCHAR2       DEFAULT NULL,
    queue_payload_type     IN VARCHAR2       DEFAULT JMS_TYPE,
    queue_properties       IN QUEUE_PROPS_T  DEFAULT NULL,
    replication_mode       IN BINARY_INTEGER DEFAULT NONE);
```

It has the following parameters:

Parameter	Description
queue_name	This required parameter specifies the name of the new queue. Maximum of 128 characters allowed.
storage_clause	The storage parameter is included in the CREATE TABLE statement when the queue table is created. The storage_clause argument can take any text that can be used in a standard CREATE TABLE storage_clause argument. The storage parameter can be made up of any combinations of the following parameters: PCTFREE, PCTUSED, INITRANS, MAXTRANS, TABLESPACE, LOB, and a table storage clause. If a tablespace is not specified here, then the queue table and all its related objects are created in the default user tablespace. If a tablespace is specified here, then the queue table and all its related objects are created in the tablespace specified in the storage clause. See *Oracle Database SQL Language Reference* for the usage of these parameters.
multiple_consumers	FALSE means queues can only have one consumer for each message. This is the default. TRUE means queues created in the table can have multiple consumers for each message.
max_retries	This optional parameter limits the number of times that a dequeue can reattempted on a message after a failure. The maximum value of max_retries is 2**31 -1. After the retry limit has been exceeded, the message will be purged from the queue. RETRY_COUNT is incremented when the application issues a rollback after executing the dequeue. If a dequeue transaction fails because the server process dies (including ALTER SYSTEM KILL SESSION) or SHUTDOWN ABORT on the instance, then RETRY_COUNT is not incremented.
comment	This optional parameter is a user-specified description of the queue table. This user comment is added to the queue catalog.
queue_payload_type	Payload can be RAW, DBMS_AQADM.JMS_TYPE, or an object type. Default is DBMS_AQADM.JMS_TYPE.
queue_properties	Properties such as Normal or Exception Queue, Retry delay, retention time, sort list and cache hint. See also *Oracle Database PL/SQL Packages and Types Reference* for more information about queue_properties.
replication_mode	Reserved for future use. DBMS_AQADM.REPLICATION_MODE if queue is being created in the Replication Mode or else DBMS_AQADM.NONE. Default is DBMS_AQADM.NONE.

Dropping a Sharded Queue

This procedure drops an existing sharded queue from the database queuing system. You must stop the queue before calling DROP_SHARDED_QUEUE. User must stop the queue explicitly if force is set to FALSE before calling DROP_SHARDED_QUEUE. If force is set to TRUE then queue will be stopped internally and then dropped.

Syntax

```
DBMS_AQADM.DROP_SHARDED_QUEUE(
       queue_name IN VARCHAR2,
       force      IN BOOLEAN DEFAULT FALSE )
```

Parameters

Table D-2 CREATE_SHARDED_QUEUE Procedure Parameters

Parameter	Description
queue_name	This required parameter specifies the name of the sharded queue.
force	The sharded queue is dropped even if the queue is not stopped.

Altering a Sharded Queue

This procedure provides user the ability to alter *queue_properties* of a sharded queue.

Syntax

```
PROCEDURE ALTER_SHARDED_QUEUE(
    queue_name             IN VARCHAR2,
    max_retries            IN NUMBER         DEFAULT NULL,
    comment                IN VARCHAR2       DEFAULT NULL,
    queue_properties       IN QUEUE_PROPS_T  DEFAULT NULL,
    replication_mode       IN BINARY_INTEGER DEFAULT NULL);
```

Parameters

Table D-3 ALTER_SHARDED_QUEUE Procedure Parameters

Parameter	Description
queue_name	This parameter specifies the name of the sharded queue. A maximum of 128 characters are allowed.
max_retries	The maximum number of retries allowed.
comment	The parameter comment.
queue_properties	Properties such as Normal or Exception Queue, Retry delay, retention time, sort list and cache hint. See also *Oracle Database PL/SQL Packages and Types Reference* for more information about queue_properties.
replication_mode	Reserved for future use. DBMS_AQADM.REPLICATION_MODE if queue is being altered to be in the Replication Mode or else DBMS_AQADM.NONE. Default is NULL.

Setting a Queue Parameter

This procedure allows user to set different parameters for sharded queues at queue or database level. For database level the queue_name should be NULL. Note that queue overrides database level parameter values.

Syntax

```
PROCEDURE  SET_QUEUE_PARAMETER(
  queue_name          IN VARCHAR2,
  param_name          IN VARCHAR2,
  param_value         IN NUMBER);
```

Parameters

Table D-4 SET_QUEUE_PARAMETER Procedure Parameters

Parameter	Description
queue_name	The name of the sharded queue.
param_name	The name of the parameter.
param_value	The value of the parameter.

Unsetting a Queue Parameter

This procedure allows user to unset different parameters for sharded queues at queue or database level. For database level the queue_name should be NULL. Note that queue overrides database level parameter values.

Syntax

```
PROCEDURE  UNSET_QUEUE_PARAMETER(
  queue_name          IN VARCHAR2,
  param_name          IN VARCHAR2);
```

Parameters

Table D-5 UNSET_QUEUE_PARAMETER Procedure Parameters

Parameter	Description
queue_name	The name of the sharded queue.
param_name	The name of the parameter.

Getting a Queue Parameter

This procedure allows user to get different parameters for sharded queues at queue or database level. For database level the queue_name should be NULL. Note that queue overrides database level parameter values.

Syntax

```
PROCEDURE  GET_QUEUE_PARAMETER(
  queue_name          IN VARCHAR2,
  param_name          IN VARCHAR2,
  param_value         OUT NUMBER);
```

Parameters

Table D-6 GET_QUEUE_PARAMETER Procedure Parameters

Parameter	Description
queue_name	The name of the sharded queue.
param_name	The name of the parameter.

Table D-6 (Cont.) GET_QUEUE_PARAMETER Procedure Parameters

Parameter	Description
param_value	The value of the parameter.

Creating an Exception Queue

This procedure allows a user to create an exception queue for a sharded queue.

Syntax

```
PROCEDURE CREATE_EXCEPTION_QUEUE(
  sharded_queue_name     IN VARCHAR2,
  exception_queue_name   IN VARCHAR2 DEFAULT NULL
  );
```

Parameters

Table D-7 CREATE_EXCEPTION_QUEUE Procedure Parameters

Parameter	Description
sharded_queue_name	The name of the sharded queue.
exception_queue_name	The name of the exception queue.