



# Oracle Built-in Packages

SEARCH

◀ PREVIOUS

**Chapter 5**  
[Oracle Advanced](#)  
[Queuing](#)

NEXT ▶



## 5.2 Getting Started with Oracle AQ

There are five basic steps involved in using Oracle AQ:

1. Install Oracle AQ for use in your database. For this, the underlying database objects must be created and the initialization file for the instance modified.
2. Authorize administrative and operational privileges on queues.
3. Set up the message queue tables and corresponding queues.
4. Enqueue messages to a queue.
5. Dequeue messages from a queue.

In many organizations, steps 1 through 3 will be performed by a database administrator. They require specific authorization and can be a complex process, depending on the type of queues you want to create. The fourth and fifth steps will be performed from within the PL/SQL programs that you have written to take advantage of this message queuing technology.

Steps 3, 4, and 5 are explained in detail later in this chapter. For now, let's take a look at the steps you must take to install AQ and to grant the proper privileges to Oracle accounts in order to perform AQ activities.

### 5.2.1 Installing the Oracle AQ Facility

Oracle Corporation has established the following guidelines for use of Oracle AQ:

- If you have a license to the Oracle8 Enterprise Edition with the Objects Option, then you will be able to take advantage of the full functionality of Oracle AQ.
- If you have a license to the Oracle8 Enterprise Edition without the Objects option, then you will be able to use Oracle AQ with queues of type RAW only.
- If you have a license to Oracle8 without the Enterprise Edition, you are not licensed to use Oracle AQ at all.

The DBMS\_AQ and DBMS\_AQADM packages (and the database object used by them) are created when the Oracle database is installed. You should not have to take any special steps to install Oracle AQ and make it available in your environment. (You will, as covered in a later section, need to grant access to Oracle AQ to specific users.)

Before you can execute any programs in the DBMS\_AQADM or DBMS\_AQ packages, you may need to grant EXECUTE privilege on those packages explicitly (role-based privileges do not take effect inside stored code). If you have trouble executing any program in these packages, connect to the SYS account and execute these commands,

```
GRANT EXECUTE ON DBMS_AQADM TO <user>;
GRANT EXECUTE ON DBMS_AQ TO <user>;
```

where <user> is the name of the account to which you want to grant EXECUTE privilege.

### 5.2.2 Database Initialization



You will need to set one or more initialization parameters in order to obtain the desired behavior from Oracle AQ.

### 5.2.2.1 Starting the Queue Monitor

One of the features of Oracle AQ is the ability to manage the time in which messages are available for dequeuing and the time after which messages are "expired." If you want to employ this "time management" feature, you need to add a parameter to your initialization file or INIT.ORA file for your database instance. The name of this parameter is AQ\_TM\_PROCESSES, and it can be assigned a nonzero integer value. If the parameter is set to any number between 1 and 10, that number of Queue Monitor background processes will be created to monitor messages in the various queues.

The name of the process created is,

```
ora_aqtm_<oracle_SID>
```

where oracle\_SID is the System ID for the database instance being started. If the parameter is not specified or is set to 0, then the Queue Monitor background process will not be created.

Here is an example of a line in the INIT.ORA file that specifies that one Queue Monitor process be created:

```
AQ_TM_PROCESSES = 1
```

If the Queue Monitor process is not started, you will not be able to start and stop the Queue Monitor using the DBMS\_AQADM.START\_TIME\_MANAGER and DBMS\_AQADM.STOP\_TIME\_MANAGER procedures, respectively.

### 5.2.2.2 Starting propagation processes

Message propagation (an Oracle 8.0.4 AQ feature) is implemented by job queue processes. The number of these processes is defined with the JOB\_QUEUE\_PROCESSES parameter. The default value for this parameter is 0. If you want message propagation to take place, you must set this parameter to at least 1. If you plan to propagate messages from many queues in your instance or receive messages to many destination queues, you (or your DBA) should set this parameter to a higher value.

Here is an example of a setting of this parameter to three processes:

```
JOB_QUEUE_PROCESSES = 3
```

See [Chapter 12, Managing Server Resources](#), for more information about DBMS\_JOB and the setting of this parameter.

### 5.2.2.3 Setting Oracle AQ compatibility

If you want to use the Oracle AQ propagation feature, set your compatibility setting in the INIT.ORA file as follows:

```
COMPATIBLE = 8.0.4
```

This parameter will be checked under any of the following conditions:

- If a call to the DBMS\_AQADM.ADD\_SUBSCRIBER procedure includes an agent (defined with the SYS.AQ\$AGENT object type) whose address field is non-NULL.
- If an agent defined with the SYS.AQ\$AGENT object type whose address field is non-NULL is specified in the recipient list field of a message properties record in a call to the DBMS\_AQ.ENQUEUE procedure.
- If you call the DBMS\_AQADM.SCHEDULE\_PROPAGATION procedure.

You can also downgrade to 8.0.3 after you have used the Oracle 8.0.4 features by using the following command:

```
ALTER DATABASE RESET COMPATIBILITY
```

Users will not be able to restart the database in 8.0.3 compatibility mode under the following conditions:

- If you have messages in queues that have not yet been propagated to their destination queues.
- If there are propagation schedules still pending for execution. If this is the case, then you will probably need to run `DBMS_AQADM.UNSCHEDULE_PROPAGATION` to remove those schedules.
- If you have queues with subscribers that specify remote destination (with a non-NULL address field in the agent). In this scenario, you will want to run `DBMS_AQADM.REMOVE_SUBSCRIBER` to remove the remote subscribers.

If you have been using Oracle AQ in Oracle 8.0.3 and are now upgrading to 8.0.4, check the online Oracle documentation for upgrade steps you must take.

### 5.2.3 Authorizing Accounts to Use Oracle AQ

When working with AQ, you will perform either administrative or operational activities. Administrative tasks include creating queue tables and queues, and starting and stopping queues. Operational tasks include different aspects of using existing queues (i.e., queuing messages to them and dequeuing messages from them). Access to these operations is granted to users through database roles. There are two such roles:

#### AQ\_ADMINISTRATOR\_ROLE

This role grants EXECUTE privileges for all programs in `DBMS_AQ` and `DBMS_AQADM`. The user SYS must grant this role to the account(s) defined to be AQ administrators.

#### AQ\_USER\_ROLE

This role grants EXECUTE privileges for all programs in `DBMS_AQ`. The AQ administrator (an account to which the `AQ_ADMINISTRATOR_ROLE` privilege has been previously granted) must grant this role to the account(s) defined to be AQ users.

**NOTE:** Administrator and user privileges are granted at the database level, and not on specific objects or within schemas. As a result, any Oracle account with the `AQ_USER_ROLE` can perform enqueue and dequeue operations on *any* queue in the database instance. Given this situation, how can you minimize the chance of messages being incorrectly queued or dequeued? You are always best off building a package around your AQ usage that hides the names of your queue tables and queues. See the [Section 5.7, "Oracle AQ Examples"](#) section for many illustrations of this technique.

Here is an example of the steps you might perform from SQL\*Plus to set up an AQ administrator:

```
SQL> CONNECT SYS/CHANGE_ON_INSTALL
SQL> GRANT AQ_ADMINISTRATOR_ROLE TO AQADMIN;
```

The AQADMIN account can now set up that old standby SCOTT as an AQ user account as follows:

```
SQL> CONNECT aqadmin/top_secret
SQL> GRANT AQ_USER_ROLE TO scott;
```

If you further wish to create and manipulate queue tables that are enabled for multiple dequeuing (in other words, queues that use subscriber lists to dequeue a message to multiple consumers), you must also execute the `GRANT_TYPE_ACCESS` procedure of the `DBMS_AQADM` package. The SYS account must do this for the AQ administrator account; that administrator can then do the same for AQ user accounts. Here are the

steps:

1. Enable AQADMIN for multiple dequeues from SYS:

```
SQL> CONNECT SYS/CHANGE_ON_INSTALL
SQL> exec DBMS_AQADM.GRANT_TYPE_ACCESS ('aqadmin');
```

2. Enable SCOTT for multiple dequeues from AQADMIN:

```
SQL> CONNECT aqadmin/top_secret
SQL> exec DBMS_AQADM.GRANT_TYPE_ACCESS ('scott');
```

Now we have two accounts that are ready, willing, and able to do some queuing! These steps are also performed for you by the files *aqadmset.sql* (for AQ administrators) and *aqfor.sql* (for AQ users).

**TIP:** If you are running Oracle 8.0.3, you might encounter problems with this approach. Under this first production release of Oracle8, you should instead perform all grant operations from the SYS or SYSTEM accounts.

◀ PREVIOUS

5.1 Oracle AQ Concepts

HOME

BOOK INDEX

NEXT ▶

5.3 Oracle AQ  
Nonprogram Elements

Library  
Home

Oracle PL/SQL  
Programming,  
Second Edition

Oracle PL/SQL  
Programming:  
Guide to Oracle8i Features

Oracle  
Built-in  
Packages

Advanced PL/SQL  
Programming  
with Packages

Oracle Web Applications:  
PL/SQL Developer's  
Introduction

Oracle PL/SQL  
Language  
Pocket Reference

Oracle PL/SQL  
Built-ins  
Pocket Reference

[Copyright \(c\) 2000](#) O'Reilly & Associates. All rights reserved.



bigmir.net  
8497 14696

Участник  
Rambler's  
TOP 100

164  
8419  
14427

+13180  
(8054)

ПЕРИМЕТР 60080598  
13560  
8024

62355908  
14553  
8492