

# 3

## Oracle Transactional Event Queues and Advanced Queuing: Programmatic Interfaces

These topics describe the various language options and elements you must work with and issues to consider in preparing your Oracle Database Advanced Queuing (AQ) application environment.



### Note:

Java package `oracle.AQ` was deprecated in 10g Release 1 (10.1). Oracle recommends that you migrate existing Java AQ applications to Oracle JMS (or other Java APIs) and use Oracle JMS (or other Java APIs) to design your future Java AQ applications.

### Topics:

- [Programmatic Interfaces for Accessing Oracle Database Advanced Queuing](#)
- [Using PL/SQL to Access Oracle Database Advanced Queuing](#)
- [Using OCI to Access Oracle Database Advanced Queuing](#)
- [Using OCCI to Access Oracle Database Advanced Queuing](#)
- [Using Oracle Java Message Service \(Oracle JMS\) to Access Oracle Database Advanced Queuing](#)
- [Using Oracle Database Advanced Queuing XML Servlet to Access Oracle Database Advanced Queuing](#)
- [Comparing Oracle Database Advanced Queuing Programmatic Interfaces](#)

## Programmatic Interfaces for Accessing Oracle Database Advanced Queuing

The table lists Oracle Database Advanced Queuing programmatic interfaces, functions supported in each interface, and syntax references.

**Table 3-1 Oracle Database Advanced Queuing Programmatic Interfaces**

Language	Precompiler or Interface Program	Functions Supported	Syntax References
PL/SQL	DBMS_AQADM and DBMS_AQ Packages	Administrative and operational	<i>Oracle Database PL/SQL Packages and Types Reference</i>

**Table 3-1 (Cont.) Oracle Database Advanced Queuing Programmatic Interfaces**

Language	Precompiler or Interface Program	Functions Supported	Syntax References
C, Python, Javascript, Node.js	<a href="#">Oracle Call Interface</a> (OCI)	Operational only	<ul style="list-style-type: none"> <li><i>Oracle Call Interface Programmer's Guide</i></li> <li><a href="https://cx-oracle.readthedocs.io/en/latest/user_guide/aq.html">https://cx-oracle.readthedocs.io/en/latest/user_guide/aq.html</a></li> <li><a href="https://oracle.github.io/node-oracledb/doc/api.html#aq">https://oracle.github.io/node-oracledb/doc/api.html#aq</a></li> </ul>
Java (JMS)	<code>oracle.JMS</code> package using JDBC <a href="#">API</a>	Administrative and operational	<i>Oracle Database Advanced Queuing Java API Reference</i>
AQ XML <a href="#">servlet</a>	<a href="#">Internet Data Access Presentation</a> (IDAP)	Operational only	<i>Oracle XML DB Developer's Guide</i>

## Using PL/SQL to Access Oracle Database Advanced Queuing

The PL/SQL packages `DBMS_AQADM` and `DBMS_AQ` support access to Oracle Database Advanced Queuing administrative and operational functions using the native Oracle Database Advanced Queuing interface.

These functions include:

- Create [queue](#), transactional event queue, [queue table](#), [nonpersistent](#) queue, multiconsumer queue/topic, RAW [message](#), or message with structured data
- Get queue table, queue, or multiconsumer queue/topic
- Alter queue table or queue/topic
- Drop queue/topic
- Start or stop queue/topic
- Grant and revoke privileges
- Add, remove, or alter [subscriber](#)
- Add, remove, or alter an Oracle Database Advanced Queuing Internet agent
- Grant or revoke privileges of database users to Oracle Database Advanced Queuing Internet agents
- Enable, disable, or alter [propagation](#) schedule
- Enqueue messages to single [consumer](#) queue (point-to-point model)
- Publish messages to multiconsumer queue/topic ([publish/subscribe](#) model)
- Subscribe for messages in multiconsumer queue
- Browse messages in a queue
- Receive messages from queue/topic
- Register to receive messages asynchronously
- Listen for messages on multiple queues/topics
- Post messages to anonymous subscriptions
- Bind or unbind agents in a [Lightweight Directory Access Protocol](#) (LDAP) server

- Add or remove aliases to Oracle Database Advanced Queuing objects in a LDAP server
- Available PL/SQL `DBMS_AQADM` and `DBMS_AQ` functions are listed in detail in Table 3–2 through Table 3–9.

**See Also:**

*Oracle Database PL/SQL Packages and Types Reference* for detailed documentation of `DBMS_AQADM` and `DBMS_AQ`, including syntax, parameters, parameter types, return values, and examples

## Using OCI and the Thin JDBC Driver to Access Oracle Database Advanced Queuing

An Oracle Call Interface (OCI) provides an interface to Oracle Database Advanced Queuing functions using the native Oracle Database Advanced Queuing interface.

The OCI interface is used to access AQ in C, Python, and Node.js. See [Table 3-1](#) for links to Python and Node.js documentation.

An OCI client can perform the following actions:

- Enqueue messages
- Dequeue messages
- Listen for messages on sets of queues
- Register to receive message notifications

In addition, OCI clients can receive [asynchronous](#) notifications for new messages in a queue using `OCISubscriptionRegister`. Transactional event queues (TxEventQ) do not support OCI clients.

### Oracle Type Translator

For queues with user-defined payload types, the Oracle type translator must be used to generate the OCI/OCCT mapping for the Oracle type. The OCI client is responsible for freeing the memory of the Oracle Database Advanced Queuing descriptors and the message payload.

**See Also:**

"OCI and Advanced Queuing" and "Publish-Subscribe Notification" in *Oracle Call Interface Programmer's Guide* for syntax details

## Python and Node.js programming interfaces for Advanced Queuing

The OCI thin driver client is used to develop the Python and Node.js implementations for operations on AQ. These calls are documented in a separate manual as shown in the links in [Table 3-1](#).

## Comparing Oracle Database Advanced Queuing Programmatic Interfaces

These topics list and compare the Oracle Database Advanced Queuing Administrative Interfaces and the Oracle Database Advanced Queuing Operational Interfaces.

Available functions for the Oracle Database Advanced Queuing programmatic interfaces are listed by use case in Table 3-2 through Table 3-9. Use cases are described in [Oracle Database Advanced Queuing Administrative Interface](#) through [Oracle Database Advanced Queuing Operations Using PL/SQL](#) and [Oracle Java Message Service Basic Operations](#) through [Oracle Java Message Service Shared Interfaces](#).

## Oracle Transactional Event Queues and Advanced Queuing Administrative Interfaces

The table lists the equivalent Oracle Transactional Event Queues(TxEventQ) and Advanced Queuing(AQ) administrative functions for the PL/SQL and Java (JMS) programmatic interfaces.

**Table 3-2 Comparison of Oracle Transactional Event Queues and Advanced Queuing Programmatic Interfaces: Administrative Interface**

Use Case	PL/SQL	Java (JMS)
Create a transactional event queue	DBMS_AQADM.CREATE_TRANSACTIONAL_EVENT_QUEUE	
Drop a transactional event queue	DBMS_AQADM.DROP_TRANSACTIONAL_EVENT_QUEUE	
Alter a transactional event queue	DBMS_AQADM.ALTER_TRANSACTIONAL_EVENT_QUEUE	
Create a connection factory	N/A	AQjmsFactory.getQueueConnectionFactory  AQjmsFactory.getTopicConnectionFactory
Register a <a href="#">ConnectionFactory</a> in an LDAP server	N/A	AQjmsFactory.registerConnectionFactory
Create a queue table	DBMS_AQADM.CREATE_QUEUE_TABLE	AQjmsSession.createQueueTable
Get a queue table	Use <i>schema.queue_table_name</i>	AQjmsSession.getQueueTable
Alter a queue table	DBMS_AQADM.ALTER_QUEUE_TABLE	AQQueueTable.alter
Drop a queue table	DBMS_AQADM.DROP_QUEUE_TABLE	AQQueueTable.drop

**Table 3-2 (Cont.) Comparison of Oracle Transactional Event Queues and Advanced Queuing Programmatic Interfaces: Administrative Interface**

Use Case	PL/SQL	Java (JMS)
Create a queue	DBMS_AQADM.CREATE_QUEUE	AQjmsSession.createQueue
Get a queue	Use <i>schema.queue_name</i>	AQjmsSession.getQueue
Create a multiconsumer queue/topic in a queue table with multiple consumers enabled	DBMS_AQADM.CREATE_QUEUE	AQjmsSession.createTopic
Get a multiconsumer queue/topic	Use <i>schema.queue_name</i>	AQjmsSession.getTopic
Alter a queue/topic	DBMS_AQADM.ALTER_QUEUE	AQjmsDestination.alter
Start a queue/topic	DBMS_AQADM.START_QUEUE	AQjmsDestination.start
Stop a queue/topic	DBMS_AQADM.STOP_QUEUE	AQjmsDestination.stop
Drop a queue/topic	DBMS_AQADM.DROP_QUEUE	AQjmsDestination.drop
Grant system privileges	DBMS_AQADM.GRANT_SYSTEM _PRIVILEGE	AQjmsSession.grantSystem Privilege
Revoke system privileges	DBMS_AQADM.REVOKE_SYSTE M_ PRIVILEGE	AQjmsSession.revokeSystem Privilege
Grant a queue/topic privilege	DBMS_AQADM.GRANT_QUEUE_ PRIVILEGE	AQjmsDestination.grantQueue Privilege  AQjmsDestination.grantTopic Privilege
Revoke a queue/topic privilege	DBMS_AQADM.REVOKE_QUEUE_ PRIVILEGE	AQjmsDestination.revokeQueue Privilege  AQjmsDestination.revokeTopic Privilege
Verify a queue type	DBMS_AQADM.VERIFY_QUEUE _TYPES	Not supported
Add a subscriber	DBMS_AQADM.ADD_SUBSCRIB ER	See <a href="#">Table 3-6</a>

**Table 3-2 (Cont.) Comparison of Oracle Transactional Event Queues and Advanced Queuing Programmatic Interfaces: Administrative Interface**

Use Case	PL/SQL	Java (JMS)
Alter a subscriber	DBMS_AQADM.ALTER_SUBSCRIBER	See <a href="#">Table 3-6</a>
Remove a subscriber	DBMS_AQADM.REMOVE_SUBSCRIBER	See <a href="#">Table 3-6</a>
Schedule propagation	DBMS_AQADM.SCHEDULE_PROPAGATION	AQjmsDestination.schedulePropagation
Enable a propagation schedule	DBMS_AQADM.ENABLE_PROPAGATION_SCHEDULE	AQjmsDestination.enablePropagationSchedule
Alter a propagation schedule	DBMS_AQADM.ALTER_PROPAGATION_SCHEDULE	AQjmsDestination.alterPropagationSchedule
Disable a propagation schedule	DBMS_AQADM.DISABLE_PROPAGATION_SCHEDULE	AQjmsDestination.disablePropagationSchedule
Unschedule a propagation	DBMS_AQADM.UNSCHEDULE_PROPAGATION	AQjmsDestination.unschedulePropagation
Create an Oracle Database Advanced Queuing Internet Agent	DBMS_AQADM.CREATE_AQ_AGENT	Not supported
Alter an Oracle Database Advanced Queuing Internet Agent	DBMS_AQADM.ALTER_AQ_AGENT	Not supported
Drop an Oracle Database Advanced Queuing Internet Agent	DBMS_AQADM.DROP_AQ_AGENT	Not supported
Grant database user privileges to an Oracle Database Advanced Queuing Internet Agent	DBMS_AQADM.ENABLE_AQ_AGENT	Not supported
Revoke database user privileges from an Oracle Database Advanced Queuing Internet Agent	DBMS_AQADM.DISABLE_AQ_AGENT	Not supported
Add alias for queue, agent, ConnectionFactory in a LDAP server	DBMS_AQADM.ADD_ALIAS_TO_LDAP	Not supported
Delete alias for queue, agent, ConnectionFactory in a LDAP server	DBMS_AQADM.DEL_ALIAS_FROM_LDAP	Not supported

## Oracle Database Advanced Queuing Operational Interfaces

These tables list equivalent Oracle Database Advanced Queuing operational functions for the programmatic interfaces PL/SQL, OCI, Oracle Database Advanced Queuing XML Servlet, and JMS, for various use cases.

**Table 3-3 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Create Connection, Session, Message Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Create a connection	N/A	OCIServer Attach	Open an HTTP connection after authenticating with the Web server	AQjmsQueueConnectionFactory .createQueueConnection  AQjmsTopicConnectionFactory .createTopicConnection
Create a session	N/A	OCISession Begin	An HTTP servlet session is automatically started with the first SOAP request	QueueConnection.createQueueSession  TopicConnection.createTopicSession
Create a RAW message	Use SQL RAW type for message	Use OCIRaw for Message	Supply the hex representation of the message payload in the XML message. For example, <code>&lt;raw&gt;023f4523&lt;/raw&gt;</code>	Not supported
Create a message with structured data	Use SQL Oracle <a href="#">object type</a> for message	Use SQL Oracle object type for message	For Oracle object type queues that are not JMS queues (that is, they are not type <code>AQ\$_JMS_*</code> ), the XML specified in <code>&lt;message payload&gt;</code> must map to the SQL type of the payload for the queue table.  For JMS queues, the XML specified in the <code>&lt;message_payload&gt;</code> must be one of the following: <code>&lt;jms_text_message&gt;</code> , <code>&lt;jms_map_message&gt;</code> , <code>&lt;jms_bytes_message&gt;</code> , <code>&lt;jms_object_message&gt;</code>	Session.createTextMessage Session.createObjectMessage Session.createMapMessage Session.createBytesMessage Session.createStreamMessage AQjmsSession.createAdtMessage
Create a message <a href="#">producer</a>	N/A	N/A	N/A	QueueSession.createSender TopicSession.createPublisher

**Table 3-4 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Enqueue Messages to a Single-Consumer Queue, Point-to-Point Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Enqueue a message to a single-consumer queue	DBMS_AQ.enqueue	OCIAQEnq	<AQXmlSend>	QueueSender.send

**Table 3-4 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Enqueue Messages to a Single-Consumer Queue, Point-to-Point Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Enqueue a message to a queue and specify visibility options	DBMS_AQ.enqueue  Specify visibility in ENQUEUE_OPTIONS	OCIAQEnq  Specify OCI_ATTR_VISIBILITY in OCIAQEnqOptions  OCIAQEnqOptions	<AQXmlSend>  Specify <visibility> in  <producer_options>	Not supported
Enqueue a message to a single-consumer queue and specify message properties priority and expiration	DBMS_AQ.enqueue  Specify priority, expiration in MESSAGE_PROPERTIES	OCIAQEnq  Specify OCI_ATTR_PRIORITY, OCI_ATTR_EXPIRATION in OCIAQMsgProperties	<AQXmlSend>  Specify <priority>, <expiration> in  <message_header>	Specify priority and TimeToLive during QueueSender.send  or  .setTimeToLive  and  MessageProducer. setPriority  followed by  QueueSender.send
Enqueue a message to a single-consumer queue and specify message properties correlationID, delay, and exception queue	DBMS_AQ.enqueue  Specify correlation, delay, exception_queue in MESSAGE_PROPERTIES	OCIAQEnq  Specify OCI_ATTR_CORRELATION, OCI_ATTR_DELAY, OCI_ATTR_EXCEPTION_QUEUE in OCIAQMsgProperties	<AQXmlSend>  Specify <correlation_id>, <delay>, <exception_queue> in  <message_header>	Message.setJMSCorrelationID  Delay and exception queue specified as provider specific message properties  JMS_OracleDelay JMS_OracleExcpQ  followed by  QueueSender.send



**Table 3-4 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Enqueue Messages to a Single-Consumer Queue, Point-to-Point Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Enqueue a message to a single-consumer queue and specify user-defined message properties	Not supported Properties should be part of payload	Not supported Properties should be part of payload	<AQXmlSend>  Specify <name> and <int_value>, <string_value>, <long_value>, and so on in  <user_properties>	Message.setIntProperty  Message.setStringProperty  Message.setBooleanProperty  and so forth, followed by  QueueSender.send
Enqueue a message to a single-consumer queue and specify message transformation	DBMS_AQ.enqueue  Specify transformation in  ENQUEUE_OPTIONS	OCIAQEnq  Specify OCI_ATTR_TRANSFORMATION in  OCIAQEnqOptions	<AQXmlSend>  Specify <transformation> in  <producer_options>	AQjmsQueueSender.setTransformation  followed by  QueueSender.send

**Table 3-5 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Publish Messages to a Multiconsumer Queue/Topic, Publish/Subscribe Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Publish a message to a multiconsumer queue/topic using default subscription list	DBMS_AQ.enqueue  Set recipient_list to NULL in  MESSAGE_PROPERTIES	OCIAQEnq  Set OCI_ATTR_RECIPIENT_LIST to NULL in  OCIAQMsgProperties	<AQXmlPublish>	TopicPublisher.publish
Publish a message to a multiconsumer queue/topic using specific recipient list See footnote-1	DBMS_AQ.enqueue  Specify recipient list in  MESSAGE_PROPERTIES	OCIAQEnq  Specify OCI_ATTR_RECIPIENT_LIST in  OCIAQMsgProperties	<AQXmlPublish>  Specify <recipient_list> in  <message_header>	AQjmsTopicPublisher.publish  Specify recipients as an array of AQjmsAgent

**Table 3-5 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Publish Messages to a Multiconsumer Queue/Topic, Publish/Subscribe Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Publish a message to a multiconsumer queue/topic and specify message properties priority and expiration	DBMS_AQ.enqueue  Specify priority, expiration in MESSAGE_PROPERTIES	OCIAQEnq  Specify OCI_ATTR_PRIORITY, OCI_ATTR_EXPIRATION in OCIAQMsgProperties	<AQXmlPublish>  Specify <priority>, <expiration> in <message_header>	Specify priority and TimeToLive during TopicPublisher. publish  or MessageProducer. setTimeToLive  and MessageProducer. setPriority  followed by TopicPublisher. publish
Publish a message to a multiconsumer queue/topic and specify send options correlationID, delay, and exception queue	DBMS_AQ.enqueue  Specify correlation, delay, exception_queue in MESSAGE_PROPERTIES	OCIAQEnq  Specify OCI_ATTR_CORRELATION, OCI_ATTR_DELAY, OCI_ATTR_EXCEPTION_QUEUE in OCIAQMsgProperties	<AQXmlPublish>  Specify <correlation_id>, <delay>, <exception_queue> in <message_header>	Message.setJMSCorrelationID  Delay and exception queue specified as provider-specific message properties JMS_OracleDelay JMS_OracleExcpQ  followed by TopicPublisher. publish
Publish a message to a topic and specify user-defined message properties	Not supported Properties should be part of payload	Not supported Properties should be part of payload	<AQXmlPublish>  Specify <name> and <int_value>, <string_value>, <long_value>, and so on in <user_properties>	Message.setIntProperty  Message.setStringProperty  Message.setBooleanProperty  and so forth, followed by TopicPublisher. publish

**Table 3-5 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Publish Messages to a Multiconsumer Queue/Topic, Publish/Subscribe Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Publish a message to a topic and specify message transformation	DBMS_AQ.enqueue  Specify transformation in  ENQUEUE_OPTIONS	OCIAQEnq  Specify OCI_ATTR_TRANSFORM ATION in  OCIAQEnqOptions	<AQXmlPublish>  Specify <transformation> in  <producer_options>	AQjmsTopic Publisher.set Transformation  followed by  TopicPublisher. publish

**Table 3-6 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Subscribing for Messages in a Multiconsumer Queue/Topic, Publish/Subscribe Model Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Add a subscriber	See administrative interfaces	Not supported	Not supported	TopicSession. createDurable Subscriber  AQjmsSession. createDurable Subscriber
Alter a subscriber	See administrative interfaces	Not supported	Not supported	TopicSession. createDurable Subscriber  AQjmsSession. createDurable Subscriber  using the new selector
Remove a subscriber	See administrative interfaces	Not supported	Not supported	AQjmsSession. unsubscribe

**Table 3-7 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Browse Messages in a Queue Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Browse messages in a queue/topic	DBMS_AQ.dequeue	OCIAQDeq	<AQXmlReceive>	QueueSession.createBrowser
	Set dequeue_mode to BROWSE in	Set OCI_ATTR_DEQ_MODE to BROWSE in	Specify <dequeue_mode> BROWSE in	QueueBrowser.getEnumeration
	DEQUEUE_OPTIONS	OCIAQDeqOptions	<consumer_options>	Not supported on topics  oracle.jms.AQjmsSession.createBrowser  oracle.jms.TopicBrowser.getEnumeration
Browse messages in a queue/topic and lock messages while browsing	DBMS_AQ.dequeue	OCIAQDeq	<AQXmlReceive>	AQjmsSession.createBrowser
	Set dequeue_mode to LOCKED in	Set OCI_ATTR_DEQ_MODE to LOCKED in	Specify <dequeue_mode> LOCKED in	set locked to TRUE. QueueBrowser.getEnumeration
	DEQUEUE_OPTIONS	OCIAQDeqOptions	<consumer_options>	Not supported on topics  oracle.jms.AQjmsSession.createBrowser  oracle.jms.TopicBrowser.getEnumeration

**Table 3-8 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Receive Messages from a Queue/Topic Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Start a connection for receiving messages	N/A	N/A	N/A	Connection.start
Create a message consumer	N/A	N/A	N/A	QueueSession.createQueueReceiver  TopicSession.createDurableSubscriber  AQjmsSession.createTopicReceiver
Dequeue a message from a queue/topic and specify visibility	DBMS_AQ.dequeue  Specify visibility in DEQUEUE_OPTIONS	OCIAQDeq  Specify OCI_ATTR_VISIBILITY in  OCIAQDeqOptions	<AQXmlReceive>  Specify <visibility> in  <consumer_options>	Not supported

**Table 3-8 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Receive Messages from a Queue/Topic Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Dequeue a message from a queue/topic and specify transformation	DBMS_AQ.dequeue  Specify transformation in  DEQUEUE_OPTIONS	OCIAQDeq  Specify OCI_ATTR_TRANSFORM ATION in  OCIAQDeqOptions	<AQXmlReceive>  Specify <transformation> in  <consumer_options>	AQjmsQueueReceiver. setTransformation  AQjmsTopicSubscriber. setTransformation  AQjmsTopicReceiver. setTransformation
Dequeue a message from a queue/topic and specify navigation mode	DBMS_AQ.dequeue  Specify navigation in  DEQUEUE_OPTIONS	OCIAQDeq  Specify OCI_ATTR_NAVIGATIO N in  OCIAQDeqOptions	<AQXmlReceive>  Specify <navigation> in  <consumer_options>	AQjmsQueueReceiver. setNavigationMode  AQjmsTopicSubscriber. setNavigationMode  AQjmsTopicReceiver. setNavigationMode
Dequeue a message from a single-consumer queue	DBMS_AQ.dequeue  Set dequeue_mode to REMOVE in  DEQUEUE_OPTIONS	OCIAQDeq  Set OCI_ATTR_DEQ_MODE to REMOVE in  OCIAQDeqOptions	<AQXmlReceive>	QueueReceiver.receive  or  QueueReceiver.receive NoWait  or  AQjmsQueueReceiver. receiveNoData
Dequeue a message from a multiconsumer queue/topic using subscription name	DBMS_AQ.dequeue  Set dequeue_mode to REMOVE and set consumer_name to subscription name in  DEQUEUE_OPTIONS	OCIAQDeq  Set OCI_ATTR_DEQ_MODE to REMOVE and set OCI_ATTR_CONSUMER_ NAME to subscription name in  OCIAQDeqOptions	<AQXmlReceive>  Specify <consumer_name> in  <consumer_options>	Create a durable TopicSubscriber on the topic using the subscription name, then  TopicSubscriber. receive  or  TopicSubscriber. receiveNoWait  or  AQjmsTopicSubscriber. receiveNoData

**Table 3-8 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Receive Messages from a Queue/Topic Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Dequeue a message from a multiconsumer queue/topic using recipient name	DBMS_AQ.dequeue  Set dequeue_mode to REMOVE and set consumer_name to recipient name in DEQUEUE_OPTIONS	OCIAQDeq  Set OCI_ATTR_DEQ_MODE to REMOVE and set OCI_ATTR_CONSUMER_NAME to recipient name in OCIAQDeqOptions	<AQXmlReceive>  Specify <consumer_name> in <consumer_options>	Create a TopicReceiver on the topic using the recipient name, then  AQjmsSession.createTopicReceiver  AQjmsTopicReceiver.receive  or  AQjmsTopicReceiver.receiveNoWait  or  AQjmsTopicReceiver.receiveNoData

**Table 3-9 Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Register to Receive Messages Asynchronously from a Queue/Topic Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Receive messages asynchronously from a single-consumer queue	Define a PL/SQL callback procedure Register it using  DBMS_AQ.REGISTER	OCISubscriptionRegister  Specify queue_name as subscription name  OCISubscriptionEnable	<AQXmlRegister>  Specify queue name in <destination> and notification mechanism in  <notify_url>	Create a QueueReceiver on the queue, then  QueueReceiver.setMessageListener
Receive messages asynchronously from a multiconsumer queue/topic	Define a PL/SQL callback procedure Register it using  DBMS_AQ.REGISTER	OCISubscriptionRegister  Specify queue:OCI_ATTR_CONSUMER_NAME as subscription name  OCISubscriptionEnable	<AQXmlRegister>  Specify queue name in <destination>, consumer in <consumer_name> and notification mechanism in  <notify_url>	Create a TopicSubscriber or TopicReceiver on the topic, then  TopicSubscriber.setMessageListener
Listen for messages on multiple queues/topics	-	-	-	-

**Table 3-9 (Cont.) Comparison of Oracle Database Advanced Queuing Programmatic Interfaces: Operational Interface—Register to Receive Messages Asynchronously from a Queue/Topic Use Cases**

Use Case	PL/SQL	OCI	AQ XML Servlet	JMS
Listen for messages on one (many) single-consumer queues	DBMS_AQ.LISTEN  Use agent_name as NULL for all agents in agent_list	OCIAQListen  Use agent_name as NULL for all agents in agent_list	Not supported	Create multiple QueueReceivers on a QueueSession, then  QueueSession.set MessageListener
Listen for messages on one (many) multiconsumer queues/Topics	DBMS_AQ.LISTEN  Specify agent_name for all agents in agent_list	OCIAQListen  Specify agent_name for all agents in agent_list	Not supported	Create multiple TopicSubscribers or TopicReceivers on a TopicSession, then  TopicSession.set MessageListener

## Using OCCI to Access Oracle Database Advanced Queuing

C++ applications can use OCCI, which has a set of Oracle Database Advanced Queuing interfaces that enable messaging clients to access Oracle Database Advanced Queuing.

OCCI AQ supports all the operational functions required to send/receive and publish/subscribe messages in a message-enabled database. Synchronous and asynchronous message consumption is available, based on a message selection rule. Transactional event queues (TxEventQ) do not support OCCI clients.



### See Also:

"Oracle Database Advanced Queuing" in *Oracle C++ Call Interface Programmer's Guide*

## Using Oracle Java Message Service (Oracle JMS) to Access Oracle Database Advanced Queuing

Java Message Service (JMS) is a messaging standard defined by Sun Microsystems, Oracle, IBM, and other vendors. JMS is a set of interfaces and associated semantics that define how a JMS client accesses the facilities of an enterprise messaging product. Oracle Java Message Service (Oracle JMS) provides a Java API for Oracle Database Advanced Queuing based on the JMS standard.

[Oracle Java Message Service](#) (Oracle JMS) supports the standard JMS interfaces and has extensions to support administrative operations and other features that are not a part of the standard.

Standard [Java Message Service](#)(JMS) features include:

- Point-to-point model of communication using queues

- Publish/subscribe model of communication using topics
- `ObjectMessage`, `StreamMessage`, `TextMessage`, `BytesMessage`, and `MapMessage` message types
- Asynchronous and [synchronous](#) delivery of messages
- Message selection based on message header fields or properties

Oracle JMS extensions include:

- Administrative API to create queue tables, queues and topics
- Point-to-multipoint communication using [recipient](#) lists for topics
- Message propagation between destinations, which allows the application to define remote subscribers
- Support for transactional sessions, enabling JMS and SQL operations in one transaction
- Message retention after messages have been dequeued
- Message delay, allowing messages to be made visible after a certain delay
- Exception handling, allowing messages to be moved to exception queues if they cannot be processed successfully
- Support for `AdtMessage`

These are stored in the database as Oracle objects, so the payload of the message can be queried after it is enqueued. Subscriptions can be defined on the contents of these messages as opposed to just the message properties.

- Topic browsing

This allows durable subscribers to browse through the messages in a publish/subscribe (topic) destination. It optionally allows these subscribers to purge the browsed messages, so they are no longer retained by Oracle Database Advanced Queuing for that subscriber.

#### See Also:

- *Java Message Service Specification*, version 1.1, March 18, 2002, Sun Microsystems, Inc.
- *Oracle Database Advanced Queuing Java API Reference*

## Accessing Standard and Oracle JMS Applications

Standard JMS interfaces are in the `javax.jms` package. Oracle JMS interfaces are in the `oracle.jms` package. You must have `EXECUTE` privilege on the `DBMS_AQIN` and `DBMS_AQJMS` packages to use the Oracle JMS interfaces. You can also acquire these rights through the `AQ_USER_ROLE` or the `AQ_ADMINISTRATOR_ROLE`. You also need the appropriate system and queue or topic privileges to [send](#) or receive messages.

Because Oracle JMS uses [Java Database Connectivity](#) (JDBC) to connect to the database, its applications can run outside the database using the JDBC OCI driver or JDBC thin driver.

## Using JDBC OCI Driver or JDBC Thin Driver

To use JMS with clients running outside the database, you must include the appropriate [JDBC driver](#), [Java Naming and Directory Interface](#) (JNDI) jar files, and Oracle Database Advanced Queuing jar files in your CLASSPATH.



Note that the Oracle Database does not support JDK 1.2, JDK 1.3, JDK 1.4, JDK5.n and all classes12\*.*.\** files. You need to use the `ojdbc6.jar` and `ojdbc7.jar` files with JDK 6.n and JDK 7.n, respectively. The following jar and zip files should be in the `CLASSPATH` based on the release of JDK you are using.

For JDK 1.5.x, the `CLASSPATH` must contain:

`ORACLE_HOME/jdbc/lib/ojdbc6.jar`

For JDK 1.6.x, the `CLASSPATH` must contain:

`ORACLE_HOME/jdbc/lib/ojdbc7.jar`

The following files are used for either JDK version:

`ORACLE_HOME/lib/jta.jar`  
`ORACLE_HOME/xdk/lib/xmlparserv2.jar`  
`ORACLE_HOME/rdbms/jlib/xdm.jar`  
`ORACLE_HOME/rdbms/jlib/aqapi.jar`  
`ORACLE_HOME/rdbms/jlib/jmscommon.jar`

### Using Oracle Server Driver in JServer

If your application is running inside the **JServer**, then you should be able to access the Oracle JMS classes that have been automatically loaded when the JServer was installed. If these classes are not available, then you must load `jmscommon.jar` followed by `aqapi.jar` using the `$ORACLE_HOME/rdbms/admin/initjms` SQL script.

## Using Oracle Database Advanced Queuing XML Servlet to Access Oracle Database Advanced Queuing

You can use Oracle Database Advanced Queuing XML servlet to access Oracle Database Advanced Queuing over HTTP using Simple Object Access Protocol (SOAP) and an Oracle Database Advanced Queuing XML message format called Internet Data Access Presentation (IDAP).

Using the Oracle Database Advanced Queuing servlet, a client can perform the following actions:

- Send messages to single-consumer queues
- Publish messages to multiconsumer queues/topics
- Receive messages from queues
- Register to receive message notifications