

Oracle Database Advanced Queuing (AQ) Types

This chapter describes the types used with Oracle Database Advanced Queuing (AQ) packages for PL/SQL, `DBMS_AQ`, and `DBMS_AQADM`.

This chapter contains the following topics:

- [Security Model](#)
- [Summary of Types](#)



See Also:

Oracle Database Advanced Queuing User's Guide for information about using Oracle Database Advanced Queuing.

Advanced Queuing (AQ) Types Security Model

`PUBLIC` is granted `EXECUTE` privilege on the types described in this chapter.

Oracle Database Advanced Queuing (AQ) Types —Summary of Types

Oracle Database Advanced Queuing (AQ) uses types to specify certain types of information.

- [AQ\\$_AGENT Type](#)
- [AQ\\$_AGENT_LIST_T Type](#)
- [AQ\\$_DESCRIPTOR Type](#)
- [AQ\\$_NTFN_DESCRIPTOR Type](#)
- [AQ\\$_NTFN_MSGID_ARRAY Type](#)
- [AQ\\$_POST_INFO Type](#)
- [AQ\\$_POST_INFO_LIST Type](#)
- [AQ\\$_PURGE_OPTIONS_T Type](#)
- [AQ\\$_RECIPIENT_LIST_T Type](#)
- [AQ\\$_REG_INFO Type](#)
- [AQ\\$_REG_INFO_LIST Type](#)
- [AQ\\$_SUBSCRIBER_LIST_T Type](#)
- [DEQUEUE_OPTIONS_T Type](#)

- [ENQUEUE_OPTIONS_T Type](#)
- [QUEUE_PROPS_T Type](#)
- [SEEK_INPUT_T Type](#)
- [SEEK_OUTPUT_T Type](#)
- [SYS.MSG_PROP_T Type](#)
- [MESSAGE_PROPERTIES_T Type](#)
- [MESSAGE_PROPERTIES_ARRAY_T Type](#)
- [MSGID_ARRAY_T Type](#)

AQ\$_AGENT Type

This type identifies a producer or a consumer of a message.

Syntax

```
TYPE SYS.AQ$_AGENT IS OBJECT (
    name          VARCHAR2(512),
    address        VARCHAR2(1024),
    protocol       NUMBER);
```

Attributes

Table 309-1 AQ\$_AGENT Attributes

Attribute	Description
name	Name of a producer or consumer of a message. The name must follow object name guidelines in the <i>Oracle Database SQL Language Reference</i> with regard to reserved characters.
address	Protocol-specific address of the recipient. If the protocol is 0, then the address is of the form <code>[schema.]queue[@dblink]</code> . For example, a queue named <code>emp_messages</code> in the HR queue at the site <code>dbs1.net</code> has the address: <code>hr.emp_messages@dbs1.net</code>
protocol	Protocol to interpret the address and propagate the message. Protocols 1-127 are reserved for internal use. If the protocol number is in the range 128 - 255, the address of the recipient is not interpreted by Oracle Database Advanced Queuing.

AQ\$_AGENT_LIST_T Type

This type identifies the list of agents for which `DBMS_AQ.LISTEN` listens.



See Also:

["AQ\\$_AGENT Type"](#)

Syntax

```
TYPE SYS.AQ$_AGENT_LIST_T IS TABLE OF SYS.AQ$_AGENT
INDEX BY BINARY_INTEGER;
```

AQ\$_DESCRIPTOR Type

This type specifies the Oracle Database Advanced Queuing descriptor received by the AQ PL/SQL callbacks upon notification.



See Also:

"MESSAGE_PROPERTIES_T Type"

Syntax

```
TYPE SYS.AQ$_DESCRIPTOR IS OBJECT (
    queue_name      VARCHAR2(261),
    consumer_name   VARCHAR2(512),
    msg_id          RAW(16),
    msg_prop        MSG_PROP_T,
    gen_desc        AQ$_NTFN_DESCRIPTOR,
    msgid_array     SYS.AQ$_NTFN_MSGID_ARRAY,
    ntfnRecdInGrp  NUMBER);
```

Attributes

Table 309-2 AQ\$_DESCRIPTOR Attributes

Attribute	Description
queue_name	Name of the queue in which the message was enqueued which resulted in the notification
consumer_name	Name of the consumer for the multiconsumer queue
msg_id	Identification number of the message
msg_prop	Message properties specified by the MSG_PROP_T type
gen_desc	Indicates the timeout specifications
msgid_array	Group notification message ID list
ntfnRecdInGrp	Notifications received in group

AQ\$_NTFN_DESCRIPTOR Type

This type is for storing a generic notification descriptor regarding PL/SQL notification flags.

Syntax

```
TYPE SYS.AQ$_NTFN_DESCRIPTOR IS OBJECT( ntfn_flags  NUMBER)
```

Attributes

Table 309-3 AQ\$_NTFN_DESCRIPTOR Attributes

Attribute	Description
ntfn_flags	Set to 1 if the notifications are already removed after a stipulated timeout. Set 2 to denote grouping. Default is 0.

AQ\$_NTFN_MSGID_ARRAY Type

This type is for storing grouping notification data for AQ namespace, value 2^{30} which is the max varray size.

Syntax

```
TYPE SYS.AQ$_NTFN_MSGID_ARRAY AS VARRAY(1073741824) OF RAW(16);
```

AQ\$_POST_INFO Type

This type specifies anonymous subscriptions to which you want to post messages.

Syntax

```
TYPE SYS.AQ$_POST_INFO IS OBJECT (
    name          VARCHAR2(512),
    namespace     NUMBER,
    payload       RAW(32767));
```

Attributes

Table 309-4 AQ\$_POST_INFO Attributes

Attribute	Description
name	Name of the anonymous subscription to which you want to post
namespace	To receive notifications from other applications through <code>DBMS_AQ.POST</code> or <code>OCISubscriptionPost()</code> , the namespace must be <code>DBMS_AQ.NAMESPACE_ANONYMOUS</code>
payload	The payload to be posted to the anonymous subscription

AQ\$_POST_INFO_LIST Type

This type identifies the list of anonymous subscriptions to which you want to post messages.



See Also:

[AQ\\$_POST_INFO Type](#)

Syntax

```
TYPE SYS.AQ$_POST_INFO_LIST AS VARRAY(1024) OF SYS.AQ$_POST_INFO;
```

AQ\$_PURGE_OPTIONS_T Type

This type specifies the options available for purging a queue table.

**See Also:**

[PURGE_QUEUE_TABLE Procedure.](#)

Syntax

```
TYPE AQ$_PURGE_OPTIONS_T IS RECORD (
    block          BOOLEAN          DEFAULT FALSE
    delivery_mode  PLS_INTEGER     DEFAULT PERSISTENT);
```

Table 309-5 AQ\$_PURGE_OPTIONS_T Type Attributes

Attribute	Description
block	<p>TRUE/FALSE.</p> <ul style="list-style-type: none"> If <code>block</code> is <code>TRUE</code>, then an exclusive lock on all the queues in the queue table is held while purging the queue table. This will cause concurrent enqueueers and dequeuers to block while the queue table is purged. The purge call always succeeds if <code>block</code> is <code>TRUE</code>. The default for <code>block</code> is <code>FALSE</code>. This will not block enqueueers and dequeuers, but it can cause the purge to fail with an error during high concurrency times.
delivery_mode	<p>Kind of messages to purge, either <code>DBMS_AQ.BUFFERED</code> or <code>DBMS_AQ.PERSISTENT</code></p>

AQ\$_RECIPIENT_LIST_T Type

Identifies the list of agents that receive the message. This type can be used only when the queue is enabled for multiple dequeues.

**See Also:**

["AQ\\$_AGENT Type"](#)

Syntax

```
TYPE SYS.AQ$_RECIPIENT_LIST_T IS TABLE OF SYS.AQ$_AGENT
    INDEX BY BINARY_INTEGER;
```

AQ\$_REG_INFO Type

This type identifies a producer or a consumer of a message.

Syntax

```
TYPE SYS.AQ$_REG_INFO IS OBJECT (
    name                VARCHAR2(512),
    namespace            NUMBER,
    callback             VARCHAR2(4000),
    context              RAW(2000) DEFAULT NULL,
    anyctx               ANYDATA,
    ctxtype              NUMBER,
    payloadcbk           VARCHAR2(4000),
    timeout              NUMBER,
    ntfn_grouping_class  NUMBER,
    ntfn_grouping_value  NUMBER DEFAULT 600,
    ntfn_grouping_type   NUMBER,
    ntfn_grouping_start_time TIMESTAMP(6) WITH TIME ZONE,
    ntfn_grouping_repeat_count NUMBER);
```

Attributes

Table 309-6 AQ\$_REG_INFO Type Attributes

Attribute	Description
name	Specifies the name of the subscription. The subscription name is of the form <i>schema.queue</i> if the registration is for a single consumer queue or <i>schema.queue:consumer_name</i> if the registration is for a multiconsumer queues.
namespace	Specifies the namespace of the subscription. To receive notification from Oracle Database Advanced Queuing queues, the namespace must be <code>DBMS_AQ.NAMESPACE_AQ</code> . To receive notifications from other applications through <code>DBMS_AQ.POST</code> or <code>OCISubscriptionPost()</code> , the namespace must be <code>DBMS_AQ.NAMESPACE_ANONYMOUS</code> .
callback	Specifies the action to be performed on message notification. For HTTP notifications, use <code>http://www.company.com:8080</code> . For e-mail notifications, use <code>mailto://xyz@company.com</code> . For raw message payload for the <code>PLSQLCALLBACK</code> procedure, use <code>plsql://schema.procedure?PR=0</code> . For user-defined type message payload converted to XML for the <code>PLSQLCALLBACK</code> procedure, use <code>plsql://schema.procedure?PR=1</code> .
context	Specifies the context that is to be passed to the callback function
anyctx	Specifies the <code>ANYDATA</code> type context that is passed to the callback function
ctxtype	Specifies the context type. Valid values are 0 (RAW) or 1 (ANYDATA)

Table 309-6 (Cont.) AQ\$_REG_INFO Type Attributes

Attribute	Description
payloadcbk	Specifies payload for the callback function
timeout	Specifies the time in seconds
ntfn_grouping_class	Currently, only the following flag can be set to specify criterion for grouping. The default value will be 0. If ntfn_grouping_class is 0, all other notification grouping attributes must be 0. <ul style="list-style-type: none"> NTFN_GROUPING_CLASS_TIME - Notifications grouped by time, that is, the user specifies a time value and a single notification gets published at the end of that time.
ntfn_grouping_value	Time-period of grouping notifications specified in seconds, meaning the time after which grouping notification would be sent periodically until ntfn_grouping_repeat_count is exhausted.
ntfn_grouping_type	<ul style="list-style-type: none"> NTFN_GROUPING_TYPE_SUMMARY - Summary of all notifications that occurred in the time interval. (Default) NTFN_GROUPING_TYPE_LAST - Last notification that occurred in the interval.
ntfn_grouping_start_time	Notification grouping start time. Notification grouping can start from a user-specified time that should a valid timestamp with time zone. If ntfn_grouping_start_time is not specified when using grouping, the default is to current timestamp with time zone
ntfn_grouping_repeat_count	Grouping notifications will be sent as many times as specified by the notification grouping repeat count and after that revert to regular notifications. The ntfn_grouping_repeat_count, if not specified, will default to <ul style="list-style-type: none"> NTFN_GROUPING_FOREVER - Keep sending grouping notifications forever.

Usage Notes

You can use the following notification mechanisms:

- OCI callback
- e-mail callback
- PL/SQL callback

[Table 309-7](#) shows the actions performed for nonpersistent queues for different notification mechanisms when RAW presentation is specified. [Table 309-8](#) shows the actions performed when XML presentation is specified.

Table 309-7 Actions Performed for Nonpersistent Queues When RAW Presentation Specified

Queue Payload Type	OCI Callback	E-mail	PL/SQL Callback
RAW	OCI callback receives the RAW data in the payload.	Not supported	PL/SQL callback receives the RAW data in the payload.
Oracle object type	Not supported	Not supported	Not supported

Table 309-8 Actions Performed for Nonpersistent Queues When XML Presentation Specified

Queue Payload Type	OCI Callback	E-mail	PL/SQL Callback
RAW	OCI callback receives the XML data in the payload.	XML data is formatted as a SOAP message and e-mailed to the registered e-mail address.	PL/SQL callback receives the XML data in the payload.
Oracle object type	OCI callback receives the XML data in the payload.	XML data is formatted as a SOAP message and e-mailed to the registered e-mail address.	PL/SQL callback receives the XML data in the payload.

AQ\$_REG_INFO_LIST Type

Identifies the list of registrations to a queue.



See Also:

"AQ\$_REG_INFO Type"

Syntax

```
TYPE SYS.AQ$_REG_INFO_LIST AS VARRAY(1024) OF SYS.AQ$_REG_INFO;
```

AQ\$_SUBSCRIBER_LIST_T Type

This type identifies the list of subscribers that subscribe to a queue.



See Also:

"AQ\$_AGENT Type"

Syntax

```
TYPE SYS.AQ$_SUBSCRIBER_LIST_T IS TABLE OF SYS.AQ$_AGENT
INDEX BY BINARY_INTEGER;
```


DEQUEUE_OPTIONS_T Type

DEQUEUE_OPTIONS_T specifies the options available for the dequeue operation.

Syntax

```
TYPE DEQUEUE_OPTIONS_T IS RECORD (
    consumer_name    VARCHAR2(30)    DEFAULT NULL,
    dequeue_mode     BINARY_INTEGER DEFAULT REMOVE,
    navigation       BINARY_INTEGER DEFAULT NEXT_MESSAGE,
    visibility       BINARY_INTEGER DEFAULT ON_COMMIT,
    wait             BINARY_INTEGER DEFAULT FOREVER,
    msgid            RAW(16)         DEFAULT NULL,
    correlation       VARCHAR2(128)   DEFAULT NULL,
    deq_condition     VARCHAR2(4000)  DEFAULT NULL,
    signature        aq$_sig_prop    DEFAULT NULL,
    transformation    VARCHAR2(61)    DEFAULT NULL,
    delivery_mode     PLS_INTEGER     DEFAULT PERSISTENT);
```

Attributes

Table 309-9 DEQUEUE_OPTIONS_T Attributes

Attribute	Description
consumer_name	<p>Name of the consumer. Only those messages matching the consumer name are accessed. If a queue is not set up for multiple consumers, then this field should be set to NULL.</p> <p>For secure queues, consumer_name must be a valid AQ agent name, mapped to the database user performing the dequeue operation, through dbms_aqadm.enable_db_access procedure call.</p>
dequeue_mode	<p>Specifies the locking behavior associated with the dequeue. Possible settings are:</p> <p>BROWSE: Read the message without acquiring any lock on the message. This specification is equivalent to a select statement.</p> <p>LOCKED: Read and obtain a write lock on the message. The lock lasts for the duration of the transaction. This setting is equivalent to a select for update statement.</p> <p>REMOVE: Read the message and delete it. This setting is the default. The message can be retained in the queue table based on the retention properties.</p> <p>REMOVE_NODATA: Mark the message as updated or deleted. The message can be retained in the queue table based on the retention properties.</p>

Table 309-9 (Cont.) DEQUEUE_OPTIONS_T Attributes

Attribute	Description
navigation	<p>Specifies the position of the message that will be retrieved. First, the position is determined. Second, the search criterion is applied. Finally, the message is retrieved. Possible settings are:</p> <p>NEXT_MESSAGE: Retrieve the next message that is available and matches the search criteria. If the previous message belongs to a message group, then AQ retrieves the next available message that matches the search criteria and belongs to the message group. This setting is the default.</p> <p>NEXT_TRANSACTION: Skip the remainder of the current transaction group (if any) and retrieve the first message of the next transaction group. This setting can only be used if message grouping is enabled for the current queue.</p> <p>FIRST_MESSAGE: Retrieves the first message which is available and matches the search criteria. This setting resets the position to the beginning of the queue.</p> <p>FIRST_MESSAGE_MULTI_GROUP: indicates that a call to <code>DBMS_AQ.DEQUEUE_ARRAY</code> will reset the position to the beginning of the queue and dequeue messages (possibly across different transaction groups) that are available and match the search criteria, until reaching the <code>ARRAY_SIZE</code> limit. Refer to the <code>TRANSACTION_GROUP</code> attribute for the message to distinguish between transaction groups.</p> <p>NEXT_MESSAGE_MULTI_GROUP: indicates that a call to <code>DBMS_AQ.DEQUEUE_ARRAY</code> will dequeue the next set of messages (possibly across different transaction groups) that are available and match the search criteria, until reaching the <code>ARRAY_SIZE</code> limit. Refer to the <code>TRANSACTION_GROUP</code> attribute for the message to distinguish between transaction groups.</p>
visibility	<p>Specifies whether the new message is dequeued as part of the current transaction. The visibility parameter is ignored when using the <code>BROWSE</code> dequeue mode. Possible settings are:</p> <p>ON_COMMIT: The dequeue will be part of the current transaction. This setting is the default.</p> <p>IMMEDIATE: The dequeue operation is not part of the current transaction, but an autonomous transaction which commits at the end of the operation.</p>
wait	<p>Specifies the wait time if there is currently no message available which matches the search criteria. Possible settings are:</p> <p>FOREVER: Wait forever. This setting is the default.</p> <p>NO_WAIT: Do not wait.</p> <p>NUMBER: Wait time in seconds.</p>
msgid	Specifies the message identifier of the message to be dequeued.
correlation	Specifies the correlation identifier of the message to be dequeued. Special pattern matching characters, such as the percent sign (%) and the underscore (_) can be used. If more than one message satisfies the pattern, then the order of dequeuing is undetermined.

Table 309-9 (Cont.) DEQUEUE_OPTIONS_T Attributes

Attribute	Description
deq_condition	<p>A conditional expression based on the message properties, the message data properties, and PL/SQL functions.</p> <p>A <code>deq_condition</code> is specified as a Boolean expression using syntax similar to the <code>WHERE</code> clause of a SQL query. This Boolean expression can include conditions on message properties, user data properties (object payloads only), and PL/SQL or SQL functions (as specified in the <code>WHERE</code> clause of a SQL query). Message properties include <code>priority</code>, <code>corrid</code> and other columns in the queue table.</p> <p>To specify dequeue conditions on a message payload (object payload), use attributes of the object type in clauses. You must prefix each attribute with <code>tab.user_data</code> as a qualifier to indicate the specific column of the queue table that stores the payload. The <code>deq_condition</code> parameter cannot exceed 4000 characters. If more than one message satisfies the dequeue condition, then the order of dequeuing is undetermined.</p>
signature	Currently not implemented
transformation	Specifies a transformation that will be applied after dequeuing the message. The source type of the transformation must match the type of the queue.
delivery_mode	The dequeuer specifies the delivery mode of the messages it wishes to dequeue in the dequeue options. It can be <code>BUFFERED</code> or <code>PERSISTENT</code> or <code>PERSISTENT_OR_BUFFERED</code> . The message properties of the dequeued message indicate the delivery mode of the dequeued message. Array dequeue is only supported for buffered messages with an array size of '1'.

ENQUEUE_OPTIONS_T Type

`ENQUEUE_OPTIONS_T` Type specifies the options available for the enqueue operation.

Syntax

```
TYPE SYS.ENQUEUE_OPTIONS_T IS RECORD (
    visibility      BINARY_INTEGER  DEFAULT ON_COMMIT,
    relative_msgid  RAW(16)         DEFAULT NULL,
    sequence_deviation BINARY_INTEGER DEFAULT NULL,
    transformation  VARCHAR2(61)    DEFAULT NULL,
    delivery_mode   PLS_INTEGER     NOT NULL DEFAULT PERSISTENT);
```

Attributes

Table 309-10 ENQUEUE_OPTIONS_T Attributes

Attribute	Description
visibility	<p>Specifies the transactional behavior of the enqueue request. Possible settings are:</p> <p><code>ON_COMMIT</code>: The enqueue is part of the current transaction. The operation is complete when the transaction commits. This setting is the default.</p> <p><code>IMMEDIATE</code>: The enqueue operation is not part of the current transaction, but an autonomous transaction which commits at the end of the operation. This is the only value allowed when enqueueing to a non-persistent queue.</p>

Table 309-10 (Cont.) ENQUEUE_OPTIONS_T Attributes

Attribute	Description
relative_msgid	Specifies the message identifier of the message which is referenced in the sequence deviation operation. This field is valid only if BEFORE is specified in sequence_deviation . This parameter is ignored if sequence deviation is not specified.
sequence_deviation	Specifies whether the message being enqueued should be dequeued before other messages already in the queue. Possible settings are: BEFORE : The message is enqueued ahead of the message specified by relative_msgid . TOP : The message is enqueued ahead of any other messages.
transformation	Specifies a transformation that will be applied before enqueueing the message. The return type of the transformation function must match the type of the queue.
delivery_mode	The enqueuer specifies the delivery mode of the messages it wishes to enqueue in the enqueue options. It can be BUFFERED or PERSISTENT . The message properties of the enqueued message indicate the delivery mode of the enqueued message. Array enqueue is only supported for buffered messages with an array size of '1'.

QUEUE_PROPS_T Type

This type specifies the Oracle Database Advanced Queuing descriptor received by the AQ PL/SQL callbacks upon notification.

Syntax

```

TYPE QUEUE_PROPS_T IS RECORD (
    retry_delay      NUMBER          DEFAULT 0,
    retention_time   NUMBER          DEFAULT 0,
    sort_list        VARCHAR2(30)    DEFAULT NULL,
    cache_hint       BINARY_INTEGER DEFAULT AUTO,
    retention_type   BINARY_INTEGER DEFAULT DBMS_AQ.DEQUEUE_TIME
);

```

Attributes

Table 309-11 QUEUE_PROPS_T Attributes

Attribute	Description
retry_delay	Specifies the number of seconds after which this message is scheduled for processing again after an application rollback
retention_time	Specifies the number of seconds a message is retained in the queue table after being dequeued from the queue
sort_list	Sharded queues can be sorted by ENQ_TIME or PRIORITY , ENQ_TIME .
cache_hint	Specifies a hint to sharded queue whether to cache messages or not. User can specify following: <ul style="list-style-type: none"> UNCACHED: Message cache will not cache messages CACHED: Message cache will cache messages and give preference to those queues with CACHED hint. AUTO: Message cache does best effort based on available memory

Table 309-11 (Cont.) QUEUE_PROPS_T Attributes

Attribute	Description
retention_type	Specifies the type of retention. DBMS_AQ.DEQUEUE_TIME (default): The retention time starts after a subshard is dequeued by all the subscribers. If a new subscriber seeks back before or to this subshard, then the retention clock is reset.

SEEK_INPUT_T Type

This type specifies the seek input per shard.

Syntax

```
TYPE SEEK_INPUT_T IS RECORD(
    shard          binary_integer DEFAULT DBMS_AQ.ALL_SHARDS,
    priority       binary_integer DEFAULT DBMS_AQ.ALL_PRIORITIES,
    seek_msgid     raw(16)        DEFAULT NULL,
    seek_time      TIMESTAMP WITH TIME ZONE DEFAULT NULL
);
```

Attributes

Table 309-12 SEEK_INPUT_T Attributes

Attribute	Description
shard	Shard identifier on which seek operation needs to be performed. Possible values are DBMS_AQ.ALL_SHARDS or a specific shard on which seek is to be performed If value is DBMS_AQ.ALL_SHARDS, then seek is performed on requested shards, otherwise it is performed on the specific shard mentioned in this field.
priority	Seek is performed on all priorities of the shard individually. If value is DBMS_AQ.ALL_PRIORITIES, then seek is performed on all priorities, otherwise it is performed on the specific priority mentioned in this field
seek_msgid	Input message id when seek_type is DBMS_AQ.MESSAGE.
seek_time	Specifies the input time when seek_type is DBMS_AQ.TIME.

SEEK_OUTPUT_T Type

This type specifies the seek output of a seek call.

Syntax

```
TYPE SEEK_OUTPUT_T IS RECORD(
    shard          binary_integer,
    priority       binary_integer,
    seeked_from    raw(16),
    seeked_to      raw(16));
```

Attributes

Table 309-13 SEEK_OUTPUT_T Attributes

Attribute	Description
shard	Specifies the shard identifier on which the seek operation is performed for the subscriber.
priority	Specifies the priority of the shard, which is applicable only for priority queues, else value is null.
seeked_from	Secifies the message id of dequeue position before seek. NULL value signifies seeking from end of shard.
seeked_to	Specifies the message id of dequeue position after seek. NULL value signifies seeking to end of shard.

SYS.MSG_PROP_T Type

This type is used in PL/SQL notification, as one field in `aq$_descriptor`, to pass message properties of an AQ message to the PL/SQL notification client callback.

Syntax

```
CREATE or replace TYPE sys.msg_prop_t AS OBJECT (
    priority          NUMBER,
    delay             NUMBER,
    expiration        NUMBER,
    correlation        VARCHAR2(128),
    attempts          NUMBER,
    exception_queue    VARCHAR2(51),
    enqueue_time      DATE,
    state             NUMBER,
    sender_id         aq$_agent,
    original_msgid     RAW(16),
    delivery_mode      NUMBER);
```

Parameters

Table 309-14 SYS.MSG_PROP_T Type Attributes

Parameter	Description
priority	Specifies the priority of the message. A smaller number indicates higher priority. The priority can be any number, including negative numbers.
delay	<p>Specifies the delay of the enqueued message. The delay represents the number of seconds after which a message is available for dequeuing. Dequeuing by <code>msgid</code> overrides the delay specification. A message enqueued with <code>delay</code> set is in the <code>WAITING</code> state, and when the delay expires, the message goes to the <code>READY</code> state. <code>DELAY</code> processing requires the queue monitor to be started. However the queue monitor is started automatically by the system if needed. Delay is set by the producer who enqueues the message.</p> <p>The possible settings follow:</p> <p><code>NO_DELAY</code>: The message is available for immediate dequeuing</p> <p>number: The number of seconds to delay the message</p>

Table 309-14 (Cont.) SYS.MSG_PROP_T Type Attributes

Parameter	Description
expiration	<p>Specifies the expiration of the message. It determines, in seconds, the duration the message is available for dequeuing. This parameter is an offset from the time the message is ready for dequeue. Expiration processing requires the queue monitor to be running. However the queue monitor is started automatically by the system if needed.</p> <p>The possible settings follow:</p> <p>NEVER: The message does not expire</p> <p>number: The number of seconds message remains in <code>READY</code> state. If the message is not dequeued before it expires, then it is moved to the exception queue in the <code>EXPIRED</code> state.</p>
correlation	Returns the identifier supplied by the producer of the message at enqueue time.
attempts	Returns the number of attempts that have been made to dequeue the message. This parameter cannot be set at enqueue time.
exception_queue	<p>Specifies the name of the queue into which the message is moved if it cannot be processed successfully.</p> <p>Messages are moved automatically into the exception queue.</p> <p>Messages are moved into the exception queue in the following cases:</p> <ul style="list-style-type: none"> RETRY_COUNT, the number of unsuccessful dequeue attempts, has exceeded the specification for the MAX_RETRIES parameter in the DBMS_AQADM.CREATE_QUEUE procedure during queue creation. <p>For multiconsumer queues, the message becomes eligible to be moved to the exception queue even if failed dequeue attempts exceeds the MAX_RETRIES parameter for only one of the consumers. But the message will not be moved until either all other consumers have successfully consumed the message or failed more than MAX_RETRIES. You can view MAX_RETRIES for a queue in the ALL_QUEUES data dictionary view.</p> <p>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.</p> <ul style="list-style-type: none"> A message was not dequeued before the expiration time elapsed. Message propagation to the specified destination queue failed with one of the following errors: <ul style="list-style-type: none"> * There were no recipients for the multiconsumer destination queue. * Recipients were specified for a single-consumer destination queue. * Destination queue was an exception queue * There was an error when applying transformation. <p>The default is the exception queue associated with the queue table. If the exception queue specified does not exist at the time of the move, then the message is moved to the default exception queue associated with the queue table, and a warning is logged in the alert log. If the default exception queue is specified, then the parameter returns a NULL value at dequeue time.</p>
enqueue_time	Specifies the time the message was enqueued. This value is determined by the system and cannot be set by the user at enqueue time.

Table 309-14 (Cont.) SYS.MSG_PROP_T Type Attributes

Parameter	Description
state	Specifies the state of the message at the time of the dequeue. This parameter cannot be set at enqueue time. The possible states follow: <ul style="list-style-type: none"> DBMS_AQ.READY: The message is ready to be processed. DBMS_AQ.WAITING: The message delay has not yet been reached. DBMS_AQ.PROCESSED: The message has been processed and is retained. DBMSAQ.EXPIRED: The message has been moved to the exception queue.
sender_id	The application-sender identification specified at enqueue time by the message producer. Sender id is of type aq\$_agent. Sender name is required for secure queues at enqueue time. This must be a valid AQ agent name, mapped to the database user performing the enqueue operation, through dbms_aqadm.enable_db_access procedure call. Sender address and protocol should not be specified. The Sender id in the message properties returned at dequeue time may have a sender address if the message was propagated from another queue. The value of the address is the source_queue, source database name if it was a remote database [format source_queue@source_database_name]
original_msgid	This parameter is used by Oracle Database Advanced Queuing for propagating messages.
delivery_mode	DBMS_AQ.BUFFERED or DBMS_AQ.PERSISTENT.

MESSAGE_PROPERTIES_T Type

This type is defined inside the DBMS_AQ package, and describes the information that AQ uses to convey the state of individual messages. These are set at enqueue time, and their values are returned at dequeue time.



See Also:

[AQ\\$_RECIPIENT_LIST_T Type](#)

Syntax

```

TYPE message_properties_t IS RECORD (
    priority          BINARY_INTEGER NOT NULL DEFAULT 1,
    delay             BINARY_INTEGER NOT NULL DEFAULT NO_DELAY,
    expiration        BINARY_INTEGER NOT NULL DEFAULT NEVER,
    correlation       VARCHAR2(128)   DEFAULT NULL,
    attempts          BINARY_INTEGER,
    recipient_list    AQ$_RECIPIENT_LIST_T,
    exception_queue   VARCHAR2(61)   DEFAULT NULL,
    enqueue_time      DATE,
    state             BINARY_INTEGER,
    sender_id         SYS.AQ$_AGENT   DEFAULT NULL,
    original_msgid    RAW(16)         DEFAULT NULL,
    signature         aq$_sig_prop    DEFAULT NULL,

```



```

transaction_group    VARCHAR2(30)    DEFAULT NULL,
user_property        SYS.ANYDATA    DEFAULT NULL
delivery_mode        PLS_INTEGER    NOT NULL DEFAULT DBMS_AQ.PERSISTENT);

```

Attributes

Table 309-15 MESSAGE_PROPERTIES_T Attributes

Attribute	Description
priority	Specifies the priority of the message. A smaller number indicates higher priority. The priority can be any number, including negative numbers.
delay	<p>Specifies the delay of the enqueued message. The delay represents the number of seconds after which a message is available for dequeuing. Dequeuing by <code>msgid</code> overrides the delay specification. A message enqueued with <code>delay</code> set is in the <code>WAITING</code> state, and when the delay expires, the message goes to the <code>READY</code> state. <code>DELAY</code> processing requires the queue monitor to be started. However the queue monitor is started automatically by the system if needed. Delay is set by the producer who enqueues the message.</p> <p>The possible settings follow:</p> <p><code>NO_DELAY</code>: The message is available for immediate dequeuing</p> <p>number: The number of seconds to delay the message</p>
expiration	<p>Specifies the expiration of the message. It determines, in seconds, the duration the message is available for dequeuing. This parameter is an offset from the time the message is ready for dequeue. Expiration processing requires the queue monitor to be running. However the queue monitor is started automatically by the system if needed.</p> <p>The possible settings follow:</p> <p><code>NEVER</code>: The message does not expire</p> <p>number: The number of seconds message remains in <code>READY</code> state. If the message is not dequeued before it expires, then it is moved to the exception queue in the <code>EXPIRED</code> state.</p>
correlation	Returns the identifier supplied by the producer of the message at enqueue time.
attempts	Returns the number of attempts that have been made to dequeue the message. This parameter cannot be set at enqueue time.
recipient_list	<p>This parameter is only valid for queues that allow multiple consumers. The default recipients are the queue subscribers. This parameter is not returned to a consumer at dequeue time.</p> <p>For type definition, see the "AQ\$AGENT Type".</p>

Table 309-15 (Cont.) MESSAGE_PROPERTIES_T Attributes

Attribute	Description
exception_queue	<p>Specifies the name of the queue into which the message is moved if it cannot be processed successfully.</p> <p>Messages are moved automatically into the exception queue. Messages are moved into the exception queue in the following cases:</p> <ul style="list-style-type: none"> RETRY_COUNT, the number of unsuccessful dequeue attempts, has exceeded the specification for the MAX_RETRIES parameter in the DBMS_AQADM.CREATE_QUEUE procedure during queue creation. <p>For multiconsumer queues, the message becomes eligible to be moved to the exception queue even if failed dequeue attempts exceeds the MAX_RETRIES parameter for only one of the consumers. But the message will not be moved until either all other consumers have successfully consumed the message or failed more than MAX_RETRIES. You can view MAX_RETRIES for a queue in the ALL_QUEUES data dictionary view.</p> <p>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.</p> <ul style="list-style-type: none"> A message was not dequeued before the expiration time elapsed. Message propagation to the specified destination queue failed with one of the following errors: <ul style="list-style-type: none"> * There were no recipients for the multiconsumer destination queue. * Recipients were specified for a single-consumer destination queue. * Destination queue was an exception queue * There was an error when applying transformation. <p>The default is the exception queue associated with the queue table. If the exception queue specified does not exist at the time of the move, then the message is moved to the default exception queue associated with the queue table, and a warning is logged in the alert log. If the default exception queue is specified, then the parameter returns a NULL value at dequeue time.</p>
enqueue_time	<p>The enqueue_time attribute specifies the time the message was enqueued. This value is always in Universal Coordinated Time (UTC), and is determined by the system and cannot be set by the user at enqueue time.</p>
state	<p>Specifies the state of the message at the time of the dequeue. This parameter cannot be set at enqueue time. The possible states follow:</p> <ul style="list-style-type: none"> DBMS_AQ.READY: The message is ready to be processed. DBMS_AQ.WAITING: The message delay has not yet been reached. DBMS_AQ.PROCESSED: The message has been processed and is retained. DBMSAQ.EXPIRED: The message has been moved to the exception queue.

Table 309-15 (Cont.) MESSAGE_PROPERTIES_T Attributes

Attribute	Description
sender_id	<p>The application-sender identification specified at enqueue time by the message producer. Sender id is of type aq\$_agent.</p> <p>Sender name is required for secure queues at enqueue time. This must be a valid AQ agent name, mapped to the database user performing the enqueue operation, through <code>dbms_aqadm.enable_db_access</code> procedure call.</p> <p>Sender address and protocol should not be specified.</p> <p>The Sender id in the message properties returned at dequeue time may have a sender address if the message was propagated from another queue. The value of the address is the <code>source_queue</code>, source database name if it was a remote database [format <code>source_queue@source_database_name</code>]</p>
original_msgid	This parameter is used by Oracle Database Advanced Queuing for propagating messages.
signature	Currently not implemented
transaction_group	Specifies the <code>transaction_group</code> for the dequeued message. Messages belonging to the same transaction group will have the same value for this attribute. This attribute is only set by the <code>DBMS_AQ.DEQUEUE_ARRAY</code> . This attribute cannot be used to set the transaction group of a message through <code>DBMS_AQ.ENQUEUE</code> or <code>DBMS_AQ.ENQUEUE_ARRAY</code> calls.
user_property	This optional attribute is used to store additional information about the payload.
delivery_mode	The message publisher specifies the delivery mode in the <code>message_properties</code> . This can be <code>DBMS_AQ.BUFFERED</code> or <code>DBMS_AQ.PERSISTENT</code> . Array enqueue is only supported for buffered messages with an array size of '1'.

MESSAGE_PROPERTIES_ARRAY_T Type

This type is used by `dbms_aq.enqueue_array` and `dbms_aq.dequeue_array` calls to hold the set of message properties.

Each element in the `payload_array` should have a corresponding element in the `MESSAGE_PROPERTIES_ARRAY_T` VARRAY.



See Also:

"MESSAGE_PROPERTIES_T Type"

Syntax

```
TYPE MESSAGE_PROPERTIES_ARRAY_T IS VARRAY (2147483647)
  OF MESSAGE_PROPERTIES_T;
```

MSGID_ARRAY_T Type

The `msgid_array_t` type is used in `dbms_aq.enqueue_array` and `dbms_aq.dequeue_array` calls to hold the set of message IDs that correspond to the enqueued or dequeued messages.

Syntax

```
TYPE MSGID_ARRAY_T IS TABLE OF RAW(16) INDEX BY BINARY_INTEGER
```