

Oracle® Database

アドバンスト・キューイング・ユーザーズ・ガイド 19c

F16967-02(原本部品番号:E96511-02)

2019年9月

タイトルおよび著作権情報

Oracle Databaseアドバンスド・キューイング・ユーザズ・ガイド 19c

F16967-02

Copyright © 1996, 2019, Oracle and/or its affiliates. All rights reserved.

原著者: Maitreyee Chaliha

原協力著者: Denis Raphaely, Neerja Bhatt, Charles Hall

原協力者: Alan Downing, Padmanabha Bhat, Longxing Deng, John Leinaweaver, Stella Kister, Qiang Liu, Anil Madan, Abhishek Saxena, James Wilson

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、このソフトウェアまたはハードウェアを安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したことにより起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除く

て、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

- [例一覧](#)
- [図一覧](#)
- [表一覧](#)
- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [対象読者](#)
 - [ドキュメントのアクセシビリティについて](#)
 - [関連ドキュメント](#)
 - [表記規則](#)
- [このリリースの『Oracle Databaseアドバンスド・キューイング・ユーザズ・ガイド』の変更内容](#)
 - [Oracle Database Advanced Queuingリリース19c, バージョン19.2の変更内容](#)
 - [新機能](#)
 - [サポート対象外機能](#)
 - [Oracle Database Advanced Queuing 12cリリース2 \(12.2\)の変更内容](#)
 - [新機能](#)
 - [Oracle Database Advanced Queuing 12cリリース1 \(12.1.0.2\)の変更内容](#)
 - [新機能](#)
 - [Oracle Database Advanced Queuing 12cリリース1 \(12.1\)の変更内容](#)
 - [新機能](#)
- [1 Oracle Database Advanced Queuingの概要](#)
 - [キューイングとは](#)
 - [Oracle Database Advanced QueuingによるOracle Databaseの活用](#)
 - [統合アプリケーション環境でのOracle Database Advanced Queuing](#)
 - [Oracle Database Advanced Queuingのクライアント/サーバー通信](#)
 - [マルチ・コンシューマによる同一メッセージのデキュー](#)
 - [Oracle Database Advanced Queuingを使用したワークフローの実装](#)
 - [Oracle Database Advanced Queuingを使用したパブリッシュ・サブスクライブの実装](#)
 - [バッファ済メッセージ](#)
 - [非同期通知](#)
 - [登録のビュー](#)
 - [イベント・ベースの通知](#)
 - [通知の時間別のグループ化](#)
 - [エンキュー機能](#)
 - [デキュー機能](#)
 - [伝播機能](#)
 - [メッセージ・フォーマットの変換](#)
 - [その他のOracle Database Advanced Queuing機能](#)
 - [Oracle Database Advanced Queuingへのインタフェース](#)
- [2 Oracle Database Advanced Queuingの基本コンポーネント](#)
 - [オブジェクト名](#)
 - [タイプ名](#)
 - [AQIエージェント型](#)

- [AQ受信者リスト型](#)
- [AQエージェント・リスト型](#)
- [AQサブスクライバ・リスト型](#)
- [AQ登録情報リスト型](#)
- [AQ転送情報リスト型](#)
- [AQ登録情報型](#)
- [AQ通知記述子型](#)
- [AQメッセージ・プロパティ型](#)
- [AQ転送情報型](#)
- [AQ\\$_NTFN_MSGID_ARRAY型](#)
- [AQ管理インタフェースの列挙定数](#)
- [AQ操作インタフェースの列挙定数](#)
- [AQバックグラウンド・プロセス](#)
 - [キュー・モニター・プロセス](#)
 - [ジョブ・キュー・プロセス](#)
 - [AQバックグラウンド・アーキテクチャ](#)
- [3 Oracle Database Advanced Queuing: プログラム・インタフェース](#)
 - [Oracle Database Advanced Queuingへのアクセスに必要なプログラム・インタフェース](#)
 - [PL/SQLを使用したOracle Database Advanced Queuingへのアクセス](#)
 - [OCIおよびThin JDBCドライバを使用したOracle Database Advanced Queuingへのアクセス](#)
 - [Oracle Database Advanced Queuingのプログラム・インタフェース別の比較](#)
 - [Oracle Database Advanced Queuing管理インタフェース](#)
 - [Oracle Database Advanced Queuing操作インタフェース](#)
 - [OCCIを使用したOracle Database Advanced Queuingへのアクセス](#)
 - [Oracle Java Message Service \(Oracle JMS\)を使用したOracle Database Advanced Queuingへのアクセス](#)
 - [Oracle Database Advanced Queuing XMLサブプレットを使用したOracle Database Advanced Queuingへのアクセス](#)
- [4 Oracle Database Advanced Queuingの管理](#)
 - [Oracle Database Advanced Queuing互換性パラメータ](#)
 - [キューのセキュリティとアクセス制御](#)
 - [Oracle Database Advanced Queuingのセキュリティ](#)
 - [管理者ロール](#)
 - [ユーザー・ロール](#)
 - [Oracle Database Advanced Queuingオブジェクト型へのアクセス](#)
 - [キューのセキュリティ](#)
 - [キュー権限とアクセス制御](#)
 - [OCIアプリケーションとキュー・アクセス](#)
 - [伝播に必要なセキュリティ](#)
 - [Oracle RACの非シャード・バッファ済メッセージに必要なセキュリティ](#)
 - [キュー表のエクスポート/インポート](#)
 - [キュー表データのエクスポ](#)
 - [キュー表データのインポート](#)
 - [データ・ポンプ・エクスポートとデータ・ポンプ・インポート](#)
 - [Oracle Enterprise Managerでの非シャード・キューのサポート](#)

- [Oracle Database Advanced QueuingとXAの併用](#)
- [キュー管理の制限事項](#)
 - [サブスクライバ](#)
 - [キュー表または関連IOTに対するDMLの未サポート](#)
 - [REFペイロード属性を持つオブジェクト・キューからの伝播](#)
 - [メッセージ・ペイロード\(実際に通信される情報\)内のコレクション型](#)
 - [キュー表およびキューにおけるシノニム](#)
 - [オブジェクト型におけるシノニム](#)
 - [表領域のPoint-in-Timeリカバリ](#)
 - [仮想プライベート・データベース](#)
- [伝播の管理](#)
 - [伝播に必要なEXECUTE権限](#)
 - [オブジェクト・キューからの伝播](#)
 - [伝播の最適化](#)
 - [伝播中の障害対策](#)
- [5 Oracle Database Advanced Queuingのパフォーマンスおよび拡張性](#)
 - [シャード・キュー](#)
 - [シャード・キューおよびメッセージ・キャッシュ](#)
 - [シャード・キューおよびエンキュー/デキュー・メッセージ](#)
 - [シャード・キューおよびネイティブJMSサポート](#)
 - [シャード・キューおよびパーティション化](#)
 - [シャード・キューおよびOracle Real Application Clusters \(Oracle RAC\)](#)
 - [シャード・キューの制限事項](#)
 - [シャード・キューのチューニング](#)
 - [ユーザー・シャーディング](#)
 - [非シャード・キュー](#)
 - [永続メッセージの基本的なチューニングのヒント](#)
 - [メモリー要件](#)
 - [格納パラメータの使用方法](#)
 - [I/O構成](#)
 - [単一の非シャード・キュー表でのエンキュー・プロセスとデキュー・プロセスの同時実行](#)
 - [単一の非シャード・キュー表でのエンキュー・プロセスとデキュー・プロセスのシリアル実行](#)
 - [キュー表の索引の作成](#)
 - [非シャード・キューに関するその他のヒント](#)
 - [伝播のチューニングのヒント](#)
 - [バッファ済メッセージのチューニング](#)
 - [非シャード・キューの永続メッセージのパフォーマンスの概要](#)
 - [非シャード・キューとOracle Real Application Clusters](#)
 - [共有サーバー環境でのOracle Database Advanced Queuing](#)
 - [パフォーマンス・ビュー](#)
- [6 Oracle Database Advanced Queuingへのインターネット・アクセス](#)
 - [インターネット経由のOracle Database Advanced Queuing操作の概要](#)
 - [Oracle Database Advanced Queuingのインターネット操作アーキテクチャ](#)
 - [インターネット・メッセージ・ペイロード](#)
 - [POSTリクエストを送信するユーザーを認証するWebサーバーの構成](#)

- [HTTPを使用したクライアント・リクエスト](#)
 - [HTTPを使用したOracle Database Advanced Queuingサブプレットのレスポンス](#)
 - [HTTPとHTTPSを使用したOracle Database Advanced Queuing伝播](#)
- [Oracle Database Advanced Queuing XMLサブプレットのデプロイ](#)
- [Internet Data Access Presentation\(IDAP\)](#)
 - [SOAPメッセージの構造](#)
 - [SOAPエンベロープ](#)
 - [SOAPヘッダー](#)
 - [SOAP本体](#)
 - [SOAPメソッドの起動](#)
 - [HTTPヘッダー](#)
 - [メソッド起動本体](#)
 - [メソッド・リクエストの結果](#)
- [リクエストおよびレスポンスのIDAPドキュメント](#)
 - [IDAPクライアントのエンキュー・リクエスト](#)
 - [IDAPクライアントのデキュー・リクエスト](#)
 - [IDAPクライアントの登録リクエスト](#)
 - [IDAPクライアントのトランザクション・コミット・リクエスト](#)
 - [IDAPクライアントのトランザクション・ロールバック・リクエスト](#)
 - [エンキュー・リクエストへのIDAPサーバー・レスポンス](#)
 - [デキュー・リクエストへのIDAPサーバー・レスポンス](#)
 - [登録リクエストへのIDAPサーバー・レスポンス](#)
 - [IDAPコミット・レスポンス](#)
 - [IDAPロールバック・レスポンス](#)
 - [IDAP通知](#)
 - [エラー発生時のIDAPレスポンス](#)
- [電子メールによるメッセージの通知](#)
- [7 Oracle Database Advanced Queuingのトラブルシューティング](#)
 - [Oracle Database Advanced Queuingの伝播に関する問題のデバッグ](#)
 - [Oracle Database Advanced Queuingのエラー・メッセージ](#)
- [8 Oracle Database Advanced Queuing管理インタフェース](#)
 - [非シャード・キュー表の管理](#)
 - [キュー表の作成](#)
 - [キュー表の変更](#)
 - [キュー表の削除](#)
 - [キュー表のページ](#)
 - [キュー表の移行](#)
 - [非シャード・キューの管理](#)
 - [キューの作成](#)
 - [キューの変更](#)
 - [キューの開始](#)
 - [キューの停止](#)
 - [キューの削除](#)
 - [シャード・キューの管理](#)
 - [シャード・キューの作成](#)

- [シャード・キューの削除](#)
 - [シャード・キューの変更](#)
 - [キューのパラメータの設定](#)
 - [キューのパラメータの設定解除](#)
 - [キューのパラメータの取得](#)
 - [例外キューの作成](#)
- [変換の管理](#)
 - [変換の作成](#)
 - [変換の変更](#)
 - [変換の削除](#)
- [権限の付与および取消し](#)
 - [Oracle Database Advanced Queuingシステム権限の付与](#)
 - [Oracle Database Advanced Queuingシステム権限の取消し](#)
 - [キュー権限の付与](#)
 - [キュー権限の取消し](#)
- [サブスクライバの管理](#)
 - [サブスクライバの追加](#)
 - [サブスクライバの変更](#)
 - [サブスクライバの削除](#)
- [伝播の管理](#)
 - [キューの伝播のスケジューリング](#)
 - [伝播キュー・タイプの検証](#)
 - [伝播スケジュールの変更](#)
 - [伝播スケジュールの有効化](#)
 - [伝播スケジュールの無効化](#)
 - [キューの伝播スケジュールの解除](#)
- [Oracle Database Advanced Queuingエージェントの管理](#)
 - [Oracle Database Advanced Queuingエージェントの作成](#)
 - [Oracle Database Advanced Queuingエージェントの変更](#)
 - [Oracle Database Advanced Queuingエージェントの削除](#)
 - [データベース・アクセスの許可](#)
 - [データベース・アクセスの禁止](#)
- [LDAPサーバーへの別名の追加](#)
- [LDAPサーバーからの別名の削除](#)
- [9 Oracle Database Advanced QueuingおよびMessaging Gatewayのビュー](#)
 - [DBA_QUEUE_TABLES: データベース内の全キュー表](#)
 - [USER_QUEUE_TABLES: ユーザー・スキーマのキュー表](#)
 - [ALL_QUEUE_TABLES: 現行ユーザーがアクセス可能なキュー表](#)
 - [DBA_QUEUES: データベース内の全キュー](#)
 - [USER_QUEUES: ユーザー・スキーマのキュー](#)
 - [ALL_QUEUES: ユーザーがなんらかの権限を持っているキュー](#)
 - [DBA_QUEUE_SCHEDULES: すべての伝播スケジュール](#)
 - [USER_QUEUE_SCHEDULES: ユーザー・スキーマの伝播スケジュール](#)
 - [QUEUE_PRIVILEGES: ユーザーがキュー権限を持っているキュー](#)
 - [AQ\\$<Queue_Table_Name>: キュー表内のメッセージ](#)

- [AQ\\$<Queue_Table_Name_S>: キューのサブスクライバ](#)
- [AQ\\$<Queue_Table_Name_R>: キューのサブスクライバおよびそのルール](#)
- [AQ\\$Queue_Name_R: マルチ・コンシューマ・キューのサブスクライバおよびそのルール](#)
- [AQ\\$Queue_Name_S: マルチ・コンシューマ・キューのサブスクライバおよびそのルール](#)
- [DBA_QUEUE_SUBSCRIBERS: データベース内の全キューのサブスクライバ](#)
- [USER_QUEUE_SUBSCRIBERS: ユーザー・スキーマのキューのサブスクライバ](#)
- [ALL_QUEUE_SUBSCRIBERS: ユーザーがキュー権限を持っているキューのサブスクライバ](#)
- [DBA_TRANSFORMATIONS: すべての変換](#)
- [DBA_ATTRIBUTE_TRANSFORMATIONS: すべての変換ファンクション](#)
- [USER_TRANSFORMATIONS: ユーザー変換](#)
- [USER_ATTRIBUTE_TRANSFORMATIONS: ユーザー変換ファンクション](#)
- [DBA_SUBSCR_REGISTRATIONS: すべてのサブスクリプション登録](#)
- [USER_SUBSCR_REGISTRATIONS: ユーザー・サブスクリプション登録](#)
- [AQ_INTERNET_USERS: インターネット・アクセスが登録されたOracle Database Advanced Queuingエージェント](#)
- [V\\$AQ: データベース内の状態ごとのメッセージ数](#)
- [V\\$BUFFERED_QUEUES: インスタンスにあるすべてのバッファ済キュー](#)
- [V\\$BUFFERED_SUBSCRIBERS: インスタンスにあるすべてのバッファ済キューのサブスクライバ](#)
- [V\\$BUFFERED_PUBLISHERS: インスタンスにあるすべてのバッファ済パブリッシャ](#)
- [V\\$PERSISTENT_QUEUES: インスタンスにあるすべてのアクティブな永続キュー](#)
- [V\\$PERSISTENT_QMN_CACHE: 永続キューのバックグラウンド・タスクのパフォーマンス統計](#)
- [V\\$PERSISTENT_SUBSCRIBERS: インスタンスにある永続キューのすべてのアクティブなサブスクライバ](#)
- [V\\$PERSISTENT_PUBLISHERS: インスタンスにある永続キューのすべてのアクティブなパブリッシャ](#)
- [V\\$PROPAGATION_SENDER: 送信\(ソース\)側のバッファ・キュー伝播スケジュール](#)
- [V\\$PROPAGATION_RECEIVER: 受信\(宛先\)側のバッファ・キュー伝播スケジュール](#)
- [V\\$SUBSCR_REGISTRATION_STATS: 通知の診断性](#)
- [V\\$METRICGROUP: メトリック・グループに関する情報](#)
- [V\\$AQ_MESSAGE_CACHE_STAT: シャード・キューのメモリー管理](#)
- [V\\$AQ_SHARDED_SUBSCRIBER_STAT: シャード・キュー・サブスクライバの統計](#)
- [V\\$AQ_MESSAGE_CACHE_ADVICE: シミュレートされたメトリック](#)
- [V\\$AQ_REMOTE_DEQUEUE_AFFINITY: デキュー・アフィニティ・インスタンス・リスト](#)
- [V\\$AQ_BACKGROUND_COORDINATOR: AQのマスター・バックグラウンド・コーディネータ・プロセス \(AQPC\)のパフォーマンス統計](#)
- [V\\$AQ_JOB_COORDINATOR: 1コーディネータ当たりのパフォーマンス統計](#)
- [V\\$AQ_SERVER_POOL: 全サーバーのパフォーマンス統計](#)
- [V\\$AQ_CROSS_INSTANCE_JOBS: クロス・プロセス・ジョブの説明](#)
- [V\\$AQ_IPC_ACTIVE_MSGS](#)
- [V\\$AQ_IPC_MSG_STATS](#)
- [V\\$AQ_IPC_PENDING_MSGS](#)
- [V\\$AQ_NONDUR_REGISTRATIONS: 非永続登録](#)
- [V\\$AQ_NOTIFICATION_CLIENTS: セキュアなOCIクライアント接続](#)
- [V\\$AQ_SUBSCRIBER_LOAD: 永続サブスクライバ](#)
- [V\\$AQ_NONDUR_SUBSCRIBER: 非永続サブスクライバ](#)
- [V\\$AQ_NONDUR_SUBSCRIBER_LWM: 非永続サブスクライバのLWM](#)
- [V\\$AQ_MESSAGE_CACHE: パフォーマンス統計](#)

- [MGW_GATEWAY: 構成およびステータスに関する情報](#)
- [MGW_AGENT_OPTIONS: サプリメンタル・オプションおよびプロパティ](#)
- [MGW_LINKS: メッセージ・システム・リンクの名前および型](#)
- [MGW_MQSERIES_LINKS: WebSphere MQメッセージ・システム・リンク](#)
- [MGW_TIBRV_LINKS: TIB/Rendezvousメッセージ・システム・リンク](#)
- [MGW_FOREIGN_QUEUES: 外部キュー](#)
- [MGW_JOBS: メッセージ・ゲートウェイ伝播ジョブ](#)
- [MGW_SUBSCRIBERS: サブスクライバに関する情報](#)
- [MGW_SCHEDULES: スケジュールに関する情報](#)
- [10 PL/SQLを使用したOracle Database Advanced Queuingの操作](#)
 - [保護キューの使用方法](#)
 - [メッセージのエンキュー](#)
 - [メッセージの配列のエンキュー](#)
 - [1つ以上のキューのリスニング](#)
 - [メッセージのデキュー](#)
 - [メッセージの配列のデキュー](#)
 - [通知の登録](#)
 - [通知の登録解除](#)
 - [サブスクライバの通知の転送](#)
 - [エージェントのLDAPサーバーへの追加](#)
 - [エージェントのLDAPサーバーからの削除](#)
- [11 Oracle Java Message Serviceの概要](#)
 - [JMSおよびOracle JMSの一般的な機能](#)
 - [JMSコネクションおよびセッション](#)
 - [ConnectionFactoryオブジェクト](#)
 - [AQjmsFactoryを使用したConnectionFactoryオブジェクトの取得](#)
 - [JNDIを使用したConnectionFactoryオブジェクトの検索](#)
 - [JMSコネクション](#)
 - [JMSセッション](#)
 - [JMS宛先](#)
 - [JMSセッションを使用したDestinationオブジェクトの取得](#)
 - [JNDIを使用したDestinationオブジェクトの検索](#)
 - [JMS宛先メソッド](#)
 - [JMSでのシステム・レベルのアクセス制御](#)
 - [JMSでの宛先レベルのアクセス制御](#)
 - [JMSでの保存およびメッセージ履歴](#)
 - [JMSでのOracle Real Application Clustersのサポート](#)
 - [JMSでの統計ビューのサポート](#)
 - [JMSでの構造化ペイロード/メッセージの型](#)
 - [JMSメッセージ・ヘッダー](#)
 - [JMSメッセージ・プロパティ](#)
 - [JMSメッセージ本体](#)
 - [StreamMessage](#)
 - [BytesMessage](#)
 - [MapMessage](#)

- [TextMessage](#)
 - [ObjectMessage](#)
 - [AdtMessage](#)
- [異なるメッセージ型でのメッセージ・プロパティの使用](#)
- [Oracle JMSを使用したバッファ済メッセージ](#)
- [JMSのバッファ済メッセージ](#)
- [JMS Point-to-Pointモデル機能](#)
- [JMSパブリッシュ・サブスクライブ・モデル機能](#)
 - [JMSパブリッシュ・サブスクライブの概要](#)
 - [DurableSubscriber](#)
 - [RemoteSubscriber](#)
 - [TopicPublisher](#)
 - [受信者リスト](#)
 - [TopicReceiver](#)
 - [TopicBrowser](#)
 - [JMSパブリッシュ・サブスクライバ操作の設定](#)
- [JMSメッセージ・プロデューサの機能](#)
 - [メッセージの優先順位および順序付け](#)
 - [メッセージ遅延の指定](#)
 - [メッセージ期限切れの指定](#)
 - [メッセージのグループ化](#)
- [JMSメッセージ・コンシューマ機能](#)
 - [メッセージの受信](#)
 - [受信におけるメッセージのナビゲーション](#)
 - [メッセージのブラウズ](#)
 - [取出しを伴わないメッセージの削除](#)
 - [遅延間隔をおいた後の再試行](#)
 - [MessageListenerを使用したメッセージの非同期受信](#)
 - [例外キュー](#)
- [JMS伝播](#)
 - [RemoteSubscriber](#)
 - [伝播スケジュール](#)
 - [拡張伝播スケジュール機能](#)
 - [伝播中の例外処理](#)
- [JMS AQのメッセージ変換](#)
- [JMSストリーミング](#)
 - [エンキューでのJMSストリーミング](#)
 - [デキューでのJMSストリーミング](#)
- [Java EEの準拠](#)
- [12 Oracle Java Message Serviceの基本操作](#)
 - [DBMS_AQINに対するEXECUTE権限](#)
 - [ConnectionFactoryの登録](#)
 - [データベースを介した登録: JDBCコネクション・パラメータの使用](#)
 - [データベースを介した登録: JDBC URLの使用](#)
 - [LDAPを介した登録: JDBCコネクション・パラメータの使用](#)

- [LDAPを介した登録: JDBC URLの使用](#)
- [キュー/トピックConnectionFactoryの登録解除](#)
 - [データベースを介した登録解除](#)
 - [LDAPを介した登録解除](#)
- [QueueConnectionFactoryまたはTopicConnectionFactoryの取得](#)
 - [QueueConnectionFactoryの取得: JDBC URLの使用](#)
 - [QueueConnectionFactoryの取得: JDBCコネクション・パラメータの使用](#)
 - [TopicConnectionFactoryの取得: JDBC URLの使用](#)
 - [TopicConnectionFactoryの取得: JDBCコネクション・パラメータの使用](#)
 - [LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの取得](#)
- [LDAP内のキューまたはトピックの取得](#)
- [非シャード・キュー表の作成](#)
- [キューの作成](#)
 - [Point-to-Pointキューの作成](#)
 - [パブリッシュ・サブスクライブ・トピックの作成](#)
 - [Point-to-Pointキューおよびパブリッシュ・サブスクライブ・トピックのためのシャード・キューの作成](#)
- [非シャード・キュー表の取得](#)
- [権限の付与および取消し](#)
 - [Oracle Database Advanced Queuingシステム権限の付与](#)
 - [Oracle Database Advanced Queuingシステム権限の取消し](#)
 - [パブリッシュ・サブスクライブ・トピック権限の付与](#)
 - [パブリッシュ・サブスクライブ・トピック権限の取消し](#)
 - [Point-to-Pointキュー権限の付与](#)
 - [Point-to-Pointキュー権限の取消し](#)
- [宛先の管理](#)
 - [宛先の開始](#)
 - [宛先の停止](#)
 - [宛先の変更](#)
 - [宛先の削除](#)
- [伝播スケジュール](#)
 - [伝播のスケジューリング](#)
 - [伝播スケジュールの有効化](#)
 - [伝播スケジュールの変更](#)
 - [伝播スケジュールの無効化](#)
 - [伝播スケジュールの解除](#)
- [13 Oracle Java Message ServiceのPoint-to-Point](#)
 - [ユーザー名/パスワードが設定されたConnectionの作成](#)
 - [コネクションの確立: デフォルトのConnectionFactoryパラメータの使用](#)
 - [ユーザー名/パスワードが設定されたQueueConnectionの作成](#)
 - [QueueConnectionの確立: オープンしているJDBCコネクションの使用](#)
 - [QueueConnectionの確立: デフォルトのConnectionFactoryパラメータの使用](#)
 - [QueueConnectionの確立: オープンしているOracleOCIConnectionPoolの使用](#)
 - [セッションの作成](#)
 - [QueueSessionの作成](#)
 - [QueueSenderの作成](#)

- [メッセージの送信: デフォルトの送信オプションのQueueSenderの使用](#)
- [メッセージの送信: 送信オプションを指定したQueueSenderの使用](#)
- [QueueBrowserの作成: 標準JMS型メッセージ](#)
- [QueueBrowserの作成: 標準JMS型メッセージ、メッセージのロック](#)
- [QueueBrowserの作成: Oracleオブジェクト型メッセージ](#)
- [QueueBrowserの作成: Oracleオブジェクト型メッセージ、メッセージのロック](#)
- [QueueReceiverの作成: 標準JMS型メッセージ](#)
- [QueueReceiverの作成: Oracleオブジェクト型メッセージ](#)
- [14 Oracle Java Message Serviceのパブリッシュ/サブスクライブ](#)
 - [ユーザー名/パスワードが設定されたConnectionの作成](#)
 - [コネクションの確立: デフォルトのConnectionFactoryパラメータの使用](#)
 - [ユーザー名/パスワードが設定されたTopicConnectionの作成](#)
 - [TopicConnectionの確立: オープンしているJDBCコネクションの使用](#)
 - [TopicConnectionの確立: オープンしているOracleOCIConnectionPoolの使用](#)
 - [セッションの作成](#)
 - [TopicSessionの作成](#)
 - [TopicPublisherの作成](#)
 - [メッセージのパブリッシュ: 最小限の指定](#)
 - [トピック指定によるメッセージのパブリッシュ](#)
 - [メッセージのパブリッシュ: 配信モード、優先順位およびTimeToLiveの指定](#)
 - [メッセージのパブリッシュ: 受信者リストの指定](#)
 - [JMSトピックに対するDurableSubscriberの作成: セレクタの指定なし](#)
 - [JMSトピックに対するDurableSubscriberの作成: セレクタの指定あり](#)
 - [Oracleオブジェクト型トピックに対するDurableSubscriberの作成: セレクタの指定なし](#)
 - [Oracleオブジェクト型トピックに対するDurableSubscriberの作成: セレクタの指定あり](#)
 - [トピック・サブスクライバ作成時の変換の指定](#)
 - [リモート・サブスクライバの作成: JMSメッセージ](#)
 - [リモート・サブスクライバの作成: Oracleオブジェクト型メッセージ](#)
 - [リモート・サブスクライバ作成時の変換の指定](#)
 - [永続サブスクリプションのサブスクライブの解除: ローカル・サブスクライバ](#)
 - [永続サブスクリプションのサブスクライブの解除: リモート・サブスクライバ](#)
 - [TopicReceiverの作成: 標準JMS型メッセージ・トピック](#)
 - [TopicReceiverの作成: Oracleオブジェクト型メッセージ・トピック](#)
 - [TopicBrowserの作成: 標準JMSメッセージ](#)
 - [TopicBrowserの作成: 標準JMSメッセージ、メッセージのロック](#)
 - [TopicBrowserの作成: Oracleオブジェクト型メッセージ](#)
 - [TopicBrowserの作成: Oracleオブジェクト型メッセージ、メッセージのロック](#)
 - [TopicBrowserを使用したメッセージのブラウズ](#)
- [15 Oracle Java Message Serviceの共有インタフェース](#)
 - [Oracle Database Advanced Queuing JMS操作インタフェース: 共有インタフェース](#)
 - [JMSコネクションの起動](#)
 - [JMSコネクションの取得](#)
 - [セッションにおけるすべての操作のコミット](#)
 - [セッションにおけるすべての操作のロールバック](#)
 - [セッションからのJDBCコネクションの取得](#)

- [JMSコネクションからのOracleOCIConnectionPoolの取得](#)
- [BytesMessageの作成](#)
- [MapMessageの作成](#)
- [StreamMessageの作成](#)
- [ObjectMessageの作成](#)
- [TextMessageの作成](#)
- [JMSメッセージの作成](#)
- [AdtMessageの作成](#)
- [JMS相関識別子の設定](#)
- [JMSメッセージ・プロパティの指定](#)
 - [Booleanメッセージ・プロパティの設定](#)
 - [Stringメッセージ・プロパティの設定](#)
 - [Integerメッセージ・プロパティの設定](#)
 - [Doubleメッセージ・プロパティの設定](#)
 - [Floatメッセージ・プロパティの設定](#)
 - [Byteメッセージ・プロパティの設定](#)
 - [Longメッセージ・プロパティの設定](#)
 - [Shortメッセージ・プロパティの設定](#)
 - [Objectメッセージ・プロパティの設定](#)
- [MessageProducerが送信するすべてのメッセージに対するデフォルトのTimeToLiveの設定](#)
- [MessageProducerが送信するすべてのメッセージに対するデフォルトの優先順位の設定](#)
- [AQjmsエージェントの作成](#)
- [メッセージの同期受信](#)
 - [タイムアウト指定でのメッセージ・コンシューマの使用](#)
 - [待機なしでのメッセージ・コンシューマの使用](#)
 - [変換による宛先からのメッセージの受信](#)
- [メッセージの受信に対するナビゲーション・モードの指定](#)
- [メッセージの非同期受信](#)
 - [メッセージ・コンシューマに対するメッセージ・リスナーの指定](#)
 - [セッションに対するメッセージ・リスナーの指定](#)
- [メッセージIDの取得](#)
 - [相関識別子の取得](#)
 - [メッセージ識別子の取得](#)
- [JMSメッセージ・プロパティの取得](#)
 - [Booleanメッセージ・プロパティの取得](#)
 - [Stringメッセージ・プロパティの取得](#)
 - [Integerメッセージ・プロパティの取得](#)
 - [Doubleメッセージ・プロパティの取得](#)
 - [Floatメッセージ・プロパティの取得](#)
 - [Byteメッセージ・プロパティの取得](#)
 - [Longメッセージ・プロパティの取得](#)
 - [Shortメッセージ・プロパティの取得](#)
 - [Objectメッセージ・プロパティの取得](#)
- [クローズおよびシャットダウン](#)
 - [MessageProducerのクローズ](#)

- [メッセージ・コンシューマのクローズ](#)
 - [JMSコネクションの停止](#)
 - [JMSセッションのクローズ](#)
 - [JMSコネクションのクローズ](#)
- [トラブルシューティング](#)
 - [JMSエラー・コードの取得](#)
 - [JMSエラー番号の取得](#)
 - [JMS例外にリンクされた例外の取得](#)
 - [JMS例外のスタック・トレースの出力](#)
 - [例外リスナーの設定](#)
 - [例外リスナーの取得](#)
- [16 Oracle Java Message Service型の例](#)
 - [Oracle Database Advanced QueuingのJMS型の設定方法の例](#)
 - [JMS BytesMessageの例](#)
 - [JMS StreamMessageの例](#)
 - [JMS MapMessageの例](#)
 - [その他のOracle Database Advanced Queuing JMSの例](#)
- [17 Oracle Messaging Gatewayの概要](#)
 - [Oracle Messaging Gatewayの概要](#)
 - [Oracle Messaging Gatewayの機能](#)
 - [Oracle Messaging Gatewayのアーキテクチャ](#)
 - [管理パッケージDBMS_MGWADM](#)
 - [Oracle Messaging Gatewayのエージェント](#)
 - [Oracle Database](#)
 - [Oracle以外のメッセージ・システム](#)
 - [伝播処理の概要](#)
 - [Oracle Database AQのバッファ済メッセージとメッセージ・ゲートウェイ](#)
- [18 Oracle Messaging Gatewayスタート・ガイド](#)
 - [Oracle Messaging Gatewayの前提条件](#)
 - [Oracle Messaging Gatewayのロードおよび設定](#)
 - [データベースへのデータベース・オブジェクトのロード](#)
 - [外部プロシージャのlistener.oraの変更](#)
 - [外部プロシージャのtnsnames.oraの変更](#)
 - [mgw.ora初期化ファイルの設定](#)
 - [Oracle Messaging Gateway管理者ユーザーの作成](#)
 - [Oracle Messaging Gatewayのエージェント・ユーザーの作成](#)
 - [Oracle Messaging Gateway接続情報の構成](#)
 - [Oracle RAC環境におけるOracle Messaging Gatewayの構成](#)
 - [メッセージ・ゲートウェイ・エージェント接続に使用する接続情報の構成](#)
 - [メッセージ・ゲートウェイ・エージェントに対するOracle RACインスタンスの設定](#)
 - [Oracle以外のメッセージ・システムの設定](#)
 - [TIB/Rendezvousの設定](#)
 - [WebSphere MQ Base JavaまたはJMSの設定](#)
 - [Oracle Messaging Gateway設定の検証](#)
 - [Oracle Messaging Gatewayのアンロード](#)

- [mgw.ora初期化ファイルについて](#)
 - [mgw.oraの初期化パラメータ](#)
 - [mgw.oraの環境変数](#)
 - [mgw.ora Javaプロパティ](#)
 - [mgw.oraコメント行](#)
- [19 Oracle Messaging Gatewayの操作](#)
 - [Oracle Messaging Gatewayのエージェントの構成](#)
 - [メッセージ・ゲートウェイ・エージェントの作成](#)
 - [メッセージ・ゲートウェイ・エージェントの削除](#)
 - [データベース接続の設定](#)
 - [リソース制限の設定](#)
 - [Oracle Messaging Gatewayのエージェントの起動と停止](#)
 - [Oracle Messaging Gatewayのエージェントの起動](#)
 - [Oracle Messaging Gatewayのエージェントの停止](#)
 - [Oracle Messaging Gatewayのエージェントのスケジューラジョブ](#)
 - [Oracle RAC上でのOracle Messaging Gatewayのエージェントの実行](#)
 - [メッセージ・システム・リンクの構成](#)
 - [WebSphere MQ Base Javaリンクの作成](#)
 - [WebSphere MQ JMSリンクの作成](#)
 - [SSLを使用するためのWebSphere MQリンクの作成](#)
 - [TIB/Rendezvousリンクの作成](#)
 - [メッセージ・システム・リンクの変更](#)
 - [メッセージ・システム・リンクの削除](#)
 - [メッセージ・システム・リンクのビュー](#)
 - [Oracle以外のメッセージ・システムのキューの構成](#)
 - [Oracle以外のキューの登録](#)
 - [WebSphere MQ Base Javaキューの登録](#)
 - [WebSphere MQ JMSキューまたはトピックの登録](#)
 - [TIB/Rendezvousサブジェクトの登録](#)
 - [Oracle以外のキューの登録解除](#)
 - [Oracle以外の登録済キューのビュー](#)
 - [Oracle Messaging Gatewayの伝播ジョブの構成](#)
 - [伝播ジョブの概要](#)
 - [Oracle Messaging Gatewayの伝播ジョブの作成](#)
 - [伝播ジョブの有効化および無効化](#)
 - [伝播ジョブのリセット](#)
 - [伝播ジョブの変更](#)
 - [伝播ジョブの削除](#)
 - [伝播ジョブ、サブスクライバおよびスケジュール](#)
 - [伝播ジョブ、サブスクライバおよびスケジュールのインタフェースの互換性](#)
 - [伝播ジョブ、サブスクライバ、スケジュールのビュー](#)
 - [伝播ソースとしてのシングル・コンシューマ・キュー](#)
 - [構成プロパティ](#)
 - [WebSphere MQシステム・プロパティ](#)
 - [TIB/Rendezvousシステム・プロパティ](#)

- [オプションのリンクの構成プロパティ](#)
 - [オプションの外部キューの構成プロパティ](#)
 - [オプションのジョブの構成プロパティ](#)
- [20 Oracle Messaging Gatewayのメッセージの変換](#)
 - [Oracle Messaging GatewayのJMS以外のメッセージの変換](#)
 - [JMS以外のメッセージ変換プロセスの概要](#)
 - [Oracle Messaging Gateway標準型](#)
 - [メッセージ・ヘッダーの変換](#)
 - [メッセージ変換を使用した任意のペイロード型の処理](#)
 - [論理変更レコードの処理](#)
 - [WebSphere MQに対するメッセージ変換](#)
 - [WebSphere MQメッセージ・ヘッダーのマッピング](#)
 - [WebSphere MQ発信伝播](#)
 - [WebSphere MQ着信伝播](#)
 - [TIB/Rendezvousに対するメッセージ変換](#)
 - [TIB/Rendezvousに対するAQメッセージ・プロパティ・マッピング](#)
 - [TIB/Rendezvous発信伝播](#)
 - [TIB/Rendezvous着信伝播](#)
 - [JMSメッセージ](#)
 - [JMS発信伝播](#)
 - [JMS着信伝播](#)
- [21 Oracle Messaging Gatewayの監視](#)
 - [Oracle Messaging Gatewayのログ・ファイル](#)
 - [Oracle Messaging Gatewayのログ・ファイル・サンプル](#)
 - [Oracle Messaging Gatewayのログ・ファイルにおける例外メッセージの解釈](#)
 - [Oracle Messaging Gatewayのエージェント・ステータスの監視](#)
 - [MGW_GATEWAYビュー](#)
 - [Oracle Messaging Gatewayのリカバリ不能なエラー・メッセージ](#)
 - [その他のOracle Messaging Gatewayのエラー状態](#)
 - [Oracle Messaging Gatewayの伝播の監視](#)
 - [Oracle Messaging Gatewayのエージェントのエラー・メッセージ](#)
- [A 非永続キュー](#)
 - [非永続キューの作成](#)
 - [非永続キューの管理](#)
 - [非永続キューの互換性](#)
 - [非永続キュー通知](#)
 - [非永続キューの制限](#)
- [B Oracle JMSおよびOracle AQ XMLサープレットのエラー・メッセージ](#)
 - [Oracle JMSのエラー・メッセージ](#)
 - [Oracle AQ XMLサープレットのエラー・メッセージ](#)
- [用語集](#)
- [索引](#)

例一覧

- [4-1 VARRAYを含むオブジェクトの作成](#)
- [8-1 AQ管理ユーザーの設定](#)
- [8-2 AQ管理のサンプル・タイプの設定](#)
- [8-3 オブジェクト型メッセージのキュー表の作成](#)
- [8-4 RAW型メッセージのキュー表の作成](#)
- [8-5 LOB型メッセージのキュー表の作成](#)
- [8-6 XMLTypeメッセージのキュー表の作成](#)
- [8-7 グループ化されたメッセージのキュー表の作成](#)
- [8-8 優先メッセージとマルチ・コンシューマのキュー表の作成](#)
- [8-9 コミット時間順序付けを使用したキュー表の作成](#)
- [8-10 マルチ・コンシューマの8.1互換キュー表の作成](#)
- [8-11 指定した表領域に対するキュー表の作成](#)
- [8-12 空きリストまたは空きリスト・グループがあるキュー表の作成](#)
- [8-13 プライマリおよびセカンダリ・インスタンスの変更によるキュー表の変更](#)
- [8-14 コメントの変更によるキュー表の変更](#)
- [8-15 キュー表の削除](#)
- [8-16 forceオプションを使用したキューの削除](#)
- [8-17 キュー表のすべてのメッセージのパージ](#)
- [8-18 指定したキューのすべてのメッセージのパージ](#)
- [8-19 指定したキューのすべてのPROCESSEDメッセージのパージ](#)
- [8-20 指定したキューと指定したコンシューマのすべてのメッセージのパージ](#)
- [8-21 指定した送信者からのすべてのメッセージの削除](#)
- [8-22 8.1互換から10.0互換へのキュー表のアップグレード](#)
- [8-23 オブジェクト型メッセージのキューの作成](#)
- [8-24 RAW型メッセージのキューの作成](#)
- [8-25 LOB型メッセージのキューの作成](#)
- [8-26 グループ化されたメッセージのキューの作成](#)
- [8-27 優先メッセージのキューの作成](#)
- [8-28 優先メッセージとマルチ・コンシューマのキューの作成](#)
- [8-29 伝播のデモ用のキューの作成](#)
- [8-30 マルチ・コンシューマの8.1形式キューの作成](#)
- [8-31 保存時間の変更によるキューの変更](#)
- [8-32 インキューおよびデキューの両方を有効にしたキューの開始](#)
- [8-33 デキューのみのキューの開始](#)
- [8-34 キューの停止](#)
- [8-35 標準キューの削除](#)
- [8-36 変換の作成](#)
- [8-37 AQシステム権限の付与](#)
- [8-38 AQシステム権限の取消し](#)
- [8-39 キュー権限の付与](#)
- [8-40 デキュー権限の取消し](#)
- [8-41 データベース・リンクの指定されたキューでのサブスクライバの追加](#)

- [8-42 データベース・リンクへのサブスクライバとしてのシングル・コンシューマ・キューの追加](#)
- [8-43 ルールを使用したサブスクライバの追加](#)
- [8-44 サブスクライバの追加と変換の指定](#)
- [8-45 マルチ・コンシューマ・キューからシングル・コンシューマ・キューへの伝播](#)
- [8-46 サブスクライバ・ルールの変更](#)
- [8-47 サブスクライバの削除](#)
- [8-48 同じデータベース内のキューへの伝播のスケジューリング](#)
- [8-49 他のデータベース内のキューへの伝播のスケジューリング](#)
- [8-50 キューからキューへの伝播のスケジューリング](#)
- [8-51 キュー・タイプの検証](#)
- [8-52 同じデータベース内のキューへの伝播スケジュールの変更](#)
- [8-53 他のデータベース内のキューへの伝播スケジュールの変更](#)
- [8-54 同じデータベース内のキューへの伝播の有効化](#)
- [8-55 他のデータベース内のキューへの伝播の有効化](#)
- [8-56 同じデータベース内のキューへの伝播の無効化](#)
- [8-57 他のデータベース内のキューへの伝播の無効化](#)
- [8-58 同じデータベース内のキューへの伝播スケジュールの解除](#)
- [8-59 他のデータベース内のキューへの伝播スケジュールの解除](#)
- [10-1 キュー名とペイロードを指定したメッセージのエンキュー](#)
- [10-2 優先順位を指定したメッセージのエンキュー](#)
- [10-3 LOB型メッセージのエンキュー・プロシージャの作成](#)
- [10-4 LOB型メッセージのエンキュー](#)
- [10-5 複数メッセージのエンキュー](#)
- [10-6 サブスクライバREDおよびGREENの追加](#)
- [10-7 マルチ・コンシューマ・キューへの複数メッセージのエンキュー](#)
- [10-8 グループ化されたメッセージのエンキュー](#)
- [10-9 遅延と期限切れを指定したメッセージのエンキュー](#)
- [10-10 変換を指定したメッセージのエンキュー](#)
- [10-11 メッセージの配列のエンキュー](#)
- [10-12 シングル・コンシューマ・キューのリスニング\(タイムアウト0\(ゼロ\)\)](#)
- [10-13 オブジェクト型メッセージのデキュー](#)
- [10-14 LOB型メッセージのデキュー・プロシージャの作成](#)
- [10-15 LOB型メッセージのデキュー](#)
- [10-16 グループ化されたメッセージのデキュー](#)
- [10-17 マルチ・コンシューマ・キューからのREDのメッセージのデキュー](#)
- [10-18 参照モードによるデキューと指定のメッセージの削除](#)
- [10-19 ロック・モードによるデキューと指定のメッセージの削除](#)
- [10-20 メッセージの配列のデキュー](#)
- [10-21 通知の登録](#)
- [10-22 オブジェクト型メッセージの転送](#)
- [12-1 データベースを介した登録: JDBCコネクション・パラメータの使用](#)
- [12-2 データベースを介した登録: JDBC URLの使用](#)
- [12-3 LDAPを介した登録: JDBCコネクション・パラメータの使用](#)
- [12-4 LDAPを介した登録: JDBC URLの使用](#)
- [12-5 データベースを介した登録解除](#)

- [12-6 LDAPを介した登録解除](#)
- [12-7 QueueConnectionFactoryの取得: JDBC URLの使用](#)
- [12-8 QueueConnectionFactoryの取得: JDBCコネクション・パラメータの使用](#)
- [12-9 TopicConnectionFactoryの取得: JDBC URLの使用](#)
- [12-10 TopicConnectionFactoryの取得: JDBCコネクション・パラメータの使用](#)
- [12-11 LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの取得](#)
- [12-12 LDAP内のキューまたはトピックの取得](#)
- [12-13 キュー表の作成](#)
- [12-14 Point-to-Pointキューの作成](#)
- [12-15 パブリッシュ・サブスクライブ・トピックの作成](#)
- [12-16 メッセージの最大試行回数と最大遅延の指定](#)
- [12-17 キュー表の取得](#)
- [12-18 Oracle Database Advanced Queuingシステム権限の付与](#)
- [12-19 Oracle Database Advanced Queuingシステム権限の取消し](#)
- [12-20 パブリッシュ・サブスクライブ・トピック権限の付与](#)
- [12-21 パブリッシュ・サブスクライブ・トピック権限の取消し](#)
- [12-22 Point-to-Pointキュー権限の付与](#)
- [12-23 Point-to-Pointキュー権限の取消し](#)
- [12-24 宛先の開始](#)
- [12-25 宛先の停止](#)
- [12-26 宛先の変更](#)
- [12-27 宛先の削除](#)
- [12-28 伝播のスケジューリング](#)
- [12-29 伝播スケジュールの有効化](#)
- [12-30 伝播スケジュールの変更](#)
- [12-31 伝播スケジュールの無効化](#)
- [12-32 伝播スケジュールの解除](#)
- [13-1 ユーザー名/パスワードが設定されたQueueConnectionの作成](#)
- [13-2 QueueConnectionの確立: オープンしているJDBCコネクションの使用](#)
- [13-3 データベース内のJavaプロシージャからのQueueConnectionの作成](#)
- [13-4 QueueConnectionの確立: オープンしているOracleOCIConnectionPoolの使用](#)
- [13-5 トランザクションQueueSessionの作成](#)
- [13-6 任意のキューにメッセージを送信するセNDERの作成](#)
- [13-7 特定のキューにメッセージを送信するセNDERの作成](#)
- [13-8 メッセージの送信: 送信オプションを指定したQueueSenderの使用1](#)
- [13-9 メッセージの送信: 送信オプションを指定したQueueSenderの使用2](#)
- [13-10 QueueBrowserの作成: セレクトタの指定なし](#)
- [13-11 QueueBrowserの作成: セレクトタの指定あり](#)
- [13-12 QueueBrowserの作成: セレクトタの指定なし、メッセージをロック](#)
- [13-13 QueueBrowserの作成: セレクトタの指定あり、メッセージをロック](#)
- [13-14 QueueBrowserの作成: ADTMessages](#)
- [13-15 AdtMessageキューに対するQueueBrowserの作成、メッセージをロック](#)
- [13-16 QueueReceiverの作成: セレクトタの指定なし](#)
- [13-17 QueueReceiverの作成: セレクトタの指定あり](#)
- [13-18 QueueReceiverの作成: AdtMessageメッセージ](#)

- [14-1 ユーザー名/パスワードが設定されたTopicConnectionの作成](#)
- [14-2 TopicConnectionの確立: オープンしているJDBCコネクションの使用](#)
- [14-3 TopicConnectionの確立: 新規JDBCコネクションの使用](#)
- [14-4 TopicConnectionの確立: オープンしているOracleOCIConnectionPoolの使用](#)
- [14-5 TopicSessionの作成](#)
- [14-6 トピック指定なしのパブリッシュ](#)
- [14-7 相関と遅延の指定によるパブリッシュ](#)
- [14-8 トピック指定によるパブリッシュ](#)
- [14-9 優先順位とTimeToLiveの指定によるパブリッシュ](#)
- [14-10 トピック・サブスクライバをオーバーライドする受信者リストの指定によるパブリッシュ](#)
- [14-11 JMSトピックに対する永続サブスクライバの作成: セレクタの指定なし](#)
- [14-12 JMSトピックに対する永続サブスクライバの作成: セレクタの指定あり](#)
- [14-13 Oracleオブジェクト型トピックに対する永続サブスクライバの作成: セレクタの指定なし](#)
- [14-14 Oracleオブジェクト型トピックに対する永続サブスクライバの作成: セレクタの指定あり](#)
- [14-15 変換による宛先へのメッセージの送信](#)
- [14-16 トピック・サブスクライバ作成時の変換の指定](#)
- [14-17 リモート・サブスクライバの作成: JMS型メッセージ・トピック](#)
- [14-18 リモート・サブスクライバの作成: Oracleオブジェクト型メッセージ・トピック](#)
- [14-19 リモート・サブスクライバ作成時の変換の指定](#)
- [14-20 永続サブスクリプションのサブスクライブの解除: ローカル・サブスクライバ](#)
- [14-21 永続サブスクリプションのサブスクライブの解除: リモート・サブスクライバ](#)
- [14-22 TopicReceiverの作成: 標準JMS型メッセージ](#)
- [14-23 TopicReceiverの作成: Oracleオブジェクト型メッセージ](#)
- [14-24 TopicBrowserの作成: セレクタの指定なし](#)
- [14-25 TopicBrowserの作成: セレクタの指定あり](#)
- [14-26 TopicBrowserの作成: セレクタの指定なし、ブラウザ中にメッセージをロック](#)
- [14-27 TopicBrowserの作成: セレクタの指定あり、メッセージをロック](#)
- [14-28 TopicBrowserの作成: AdtMessageメッセージ](#)
- [14-29 TopicBrowserの作成: AdtMessageメッセージ、メッセージのロック](#)
- [14-30 TopicBrowserの作成: セレクタの指定あり](#)
- [15-1 JMSセッションからの基礎となるJDBCコネクションの取得](#)
- [15-2 JMSコネクションからの基礎となるOracleOCIConnectionPoolの取得](#)
- [15-3 MessageProducerが送信するすべてのメッセージに対するデフォルトのTimeToLiveの設定](#)
- [15-4 QueueSenderが送信するすべてのメッセージに対するデフォルトの優先順位の設定](#)
- [15-5 TopicPublisherが送信するすべてのメッセージに対するデフォルトの優先順位の設定](#)
- [15-6 タイムアウト指定でのメッセージ・コンシューマの使用](#)
- [15-7 JMS: メッセージが届くまでブロック](#)
- [15-8 JMS: メッセージを非ブロック](#)
- [15-9 JMS: 変換による宛先からのメッセージの受信](#)
- [15-10 メッセージの受信に対するナビゲーション・モードの指定](#)
- [15-11 メッセージ・コンシューマに対するメッセージ・リスナーの指定](#)
- [15-12 Objectメッセージ・プロパティの取得](#)
- [15-13 コネクション用の例外リスナーの指定](#)
- [15-14 ExceptionListenerとMessageListenerの組合せ使用](#)
- [15-15 コネクション用の例外リスナーの取得](#)

- [16-1 JMS型の例を実行する環境の設定](#)
- [16-2 例の設定](#)
- [16-3 BytesMessageの移入とエンキュー](#)
- [16-4 JMS BytesMessageデータのデキューおよび取出し](#)
- [16-5 JMS StreamMessageの移入とエンキュー](#)
- [16-6 JMS StreamMessageからのデータのデキューおよび取出し](#)
- [16-7 JMS MapMessageの移入とエンキュー](#)
- [16-8 JMS MapMessageからのデータのデキューおよび取出し](#)
- [16-9 大きなTextMessageのエンキュー](#)
- [16-10 大きなBytesMessageのエンキュー](#)
- [18-1 リスナーの静的サービス情報の追加](#)
- [18-2 MGW_AGENTの構成](#)
- [18-3 メッセージ・ゲートウェイ管理者ユーザーの作成](#)
- [18-4 メッセージ・ゲートウェイ・エージェント・ユーザーの作成](#)
- [18-5 メッセージ・ゲートウェイ接続情報の構成](#)
- [18-6 Javaプロパティの設定](#)
- [19-1 メッセージ・ゲートウェイ・エージェントの作成](#)
- [19-2 メッセージ・ゲートウェイ・エージェントの削除](#)
- [19-3 データベース接続情報の設定](#)
- [19-4 リソース制限の設定](#)
- [19-5 メッセージ・ゲートウェイ・エージェントの起動](#)
- [19-6 メッセージ・ゲートウェイ・エージェントの停止](#)
- [19-7 WebSphere MQ Base Javaリンクの構成](#)
- [19-8 WebSphere MQ JMSリンクの構成](#)
- [19-9 SSLのためのWebSphere MQ Base Javaリンクの構成](#)
- [19-10 TIB/Rendezvousリンクの構成](#)
- [19-11 WebSphere MQリンクの変更](#)
- [19-12 メッセージ・ゲートウェイのリンクの削除](#)
- [19-13 すべてのメッセージ・ゲートウェイのリンクのリスト](#)
- [19-14 メッセージ・システム・リンクの構成情報のチェック](#)
- [19-15 WebSphere MQ Base Javaキューの登録](#)
- [19-16 Oracle以外のキューの登録解除](#)
- [19-17 登録済キューの確認](#)
- [19-18 メッセージ・ゲートウェイ伝播ジョブの作成](#)
- [19-19 メッセージ・ゲートウェイ伝播ジョブの有効化](#)
- [19-20 メッセージ・ゲートウェイ伝播ジョブの無効化](#)
- [19-21 伝播ジョブのリセット](#)
- [19-22 例外キューの追加による伝播ジョブの変更](#)
- [19-23 ポーリンク間隔変更による伝播ジョブの変更](#)
- [19-24 伝播ジョブの削除](#)
- [20-1 変換関クションのシグネチャ](#)
- [20-2 変換の作成](#)
- [20-3 変換の登録](#)
- [20-4 発信LCRの変換](#)
- [20-5 着信LCRの変換](#)

- [21-1 メッセージ・ゲートウェイのサンプル・ログ・ファイル](#)
- [21-2 例外メッセージ・サンプル](#)

図一覧

- [1-1 Oracle Database Advanced Queuingを使用した統合アプリケーション環境](#)
- [1-2 Oracle Database Advanced Queuingを使用したクライアント/サーバー通信](#)
- [1-3 マルチ・コンシューマ・キューを使用した通信](#)
- [1-4 メッセージの明示的および暗黙的な受信者](#)
- [1-5 Oracle Database Advanced Queuingを使用したワークフローの実装](#)
- [1-6 Point-to-Pointメッセージ](#)
- [1-7 パブリッシュ・サブスクライブ・モード](#)
- [1-8 Oracle Database Advanced Queuingを使用したパブリッシュ・サブスクライブの実装](#)
- [1-9 Oracle Database Advanced Queuingでのメッセージの伝播](#)
- [1-10 アプリケーション統合における変換](#)
- [1-11 HTTPを使用してOracle Database Advanced Queuing操作を実行するためのアーキテクチャ](#)
- [6-1 HTTPを使用してOracle Database Advanced Queuing操作を実行するためのアーキテクチャ](#)
- [6-2 HTTPでのOracle Database Advanced Queuing伝播](#)
- [11-1 LDAPサーバー内のOracle Database Advanced Queuingエントリの構造](#)
- [17-1 メッセージ・ゲートウェイのアーキテクチャ](#)
- [20-1 JMS以外のメッセージの変換](#)
- [20-2 Oracle Database Advanced Queuingのメッセージ変換](#)
- [20-3 SYS.MGW_BASIC_MSG_T使用によるWebSphere MQのメッセージ変換](#)
- [20-4 TIB/Rendezvousに対するメッセージ変換](#)
- [20-5 JMSメッセージの伝播](#)

表一覧

- [2-1 AQ\\$_REG_INFO型の属性](#)
- [2-2 AQ\\$_DESCRIPTORの属性](#)
- [2-3 Oracle Database Advanced Queuing管理インタフェースの列挙定数](#)
- [2-4 Oracle Database Advanced Queuing操作インタフェースの列挙定数](#)
- [3-1 Oracle Database Advanced Queuingのプログラム・インタフェース](#)
- [3-2 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 管理インタフェース](#)
- [3-3 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—コネクション、セッション、メッセージの作成での利用方法](#)
- [3-4 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—シングル・コンシューマ・キューへのメッセージのエンキュー\(Point-to-Pointモデルでの利用方法\)](#)
- [3-5 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—マルチ・コンシューマ・キュー/トピックへのメッセージのパブリッシュ\(パブリッシュ・サブスクライブ・モデルでの利用方法\)](#)
- [3-6 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—マルチ・コンシューマ・キュー/トピックのメッセージに対するサブスクライブ\(パブリッシュ・サブスクライブ・モデルでの利用方法\)](#)
- [3-7 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—キューのメッセージのブラウズでの利用方法](#)
- [3-8 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—キュー/トピックからのメッセージの受信での利用方法](#)
- [3-9 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—キュー/トピックからメッセージを非同期受信するための登録での利用方法](#)
- [4-1 操作および必要な権限](#)
- [8-1 CREATE_SHARDED_QUEUEプロシージャのパラメータ](#)
- [8-2 ALTER_SHARDED_QUEUEプロシージャのパラメータ](#)
- [8-3 SET_QUEUE_PARAMETERプロシージャのパラメータ](#)
- [8-4 UNSET_QUEUE_PARAMETERプロシージャのパラメータ](#)
- [8-5 GET_QUEUE_PARAMETERプロシージャのパラメータ](#)
- [8-6 CREATE_EXCEPTION_QUEUEプロシージャのパラメータ](#)
- [9-1 AQ\\$<Queue_Table_Name>ビュー](#)
- [9-2 AQ\\$<Queue_Table_Name_S>ビュー](#)
- [9-3 AQ\\$<Queue_Table_Name_R>ビュー](#)
- [9-4 AQ\\$Queue_Name_R: マルチ・コンシューマ・キューのサブスクライバおよびそのルール](#)
- [9-5 AQ\\$Queue_Name_S: マルチ・コンシューマ・キューのサブスクライバおよびそのルール](#)
- [9-6 AQ\\$INTERNET_USERSビュー](#)
- [9-7 V\\$AQ_MESSAGE_CACHE_STATビュー](#)
- [9-8 V\\$AQ_SHARDED_SUBSCRIBER_STATビュー](#)
- [9-9 V\\$AQ_MESSAGE_CACHE_ADVICEビュー](#)
- [9-10 V\\$AQ_REMOTE_DEQUEUE_AFFINITYビュー](#)
- [9-11 MGW_GATEWAYビューのプロパティ](#)
- [9-12 MGW_AGENT_OPTIONSビュー](#)
- [9-13 MGW_LINKSビューのプロパティ](#)
- [9-14 MGW_MQSERIES_LINKSビューのプロパティ](#)

- [9-15 MGW_TIBRV_LINKSビューのプロパティ](#)
- [9-16 MGW_FOREIGN_QUEUESビューのプロパティ](#)
- [9-17 MGW_JOBSビュー](#)
- [9-18 MGW_SUBSCRIBERSビューのプロパティ](#)
- [9-19 MGW_SCHEDULESビューのプロパティ](#)
- [11-1 StreamMessageの変換](#)
- [11-2 MapMessageの変換](#)
- [11-3 Oracle Database AQとOracle JMSのバッファ済メッセージ定数](#)
- [18-1 SID_DESCパラメータ](#)
- [19-1 メッセージ・ゲートウェイ伝播ジョブのサブプログラム](#)
- [19-2 WebSphere MQリンク・プロパティ](#)
- [19-3 WebSphere MQ Base Javaのオプションの構成プロパティ](#)
- [19-4 WebSphere MQ JMSのオプションの構成プロパティ](#)
- [19-5 TIB/Rendezvousリンク・プロパティ](#)
- [19-6 TIB/Rendezvousのオプション・プロパティ](#)
- [20-1 WebSphere MQのヘッダー値に対するMGWの名前](#)
- [20-2 伝播のデフォルト優先順位のマッピング](#)
- [20-3 TIB/Rendezvousデータ型のマッピング](#)
- [20-4 Oracle Database Advanced Queuingメッセージ・プロパティに対するTIB/RendezvousおよびMGWの名前](#)
- [20-5 Oracle JMSメッセージの変換](#)
- [A-1 RAW表現が指定された場合に実行される非永続キューのアクション](#)
- [A-2 XML表現が指定された場合に実行される非永続キューのアクション](#)

はじめに

このマニュアルでは、Oracle Database Advanced Queuing (AQ)を使用したアプリケーションの開発および統合に関する機能について説明します。この情報は、特に指定されないかぎり、すべてのプラットフォームで実行するOracle Databaseサーバーの各バージョンに適用されます。

ここでは、次の項目について説明します。

- [対象読者](#)
- [ドキュメントのアクセシビリティについて](#)
- [関連ドキュメント](#)
- [表記規則](#)

対象読者

このガイドは、Oracle Database Advanced Queuingを使用するアプリケーションの開発者を対象としています。

このガイドを使用するには、アプリケーションの開発言語とオブジェクト・リレーショナル・データベース管理の概念に関する知識が必要です。

ドキュメントのアクセシビリティについて

Oracleのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートを購入したオラクル社のお客様は、My Oracle Supportを介して電子的なサポートにアクセスできます。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- [『Oracle Database開発ガイド』](#)
- [『Oracle Database PL/SQL言語リファレンス』](#)
- [『Oracle Database Advanced Queuing Java APIリファレンス』](#)
- [『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)
- [『Oracle XML DB開発者ガイド』](#)

このガイドでは、Oracle Databaseのインストール時に「基本インストール」オプションを選択した場合にデフォルトでインストールされる、サンプル・スキーマを例として多用しています。それらのスキーマの作成方法および使用方法については、[『Oracle Databaseサンプル・スキーマ』](#)を参照してください。

表記規則

このマニュアルでは次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック体	イタリックは、ユーザーが特定の値を指定するプレースホルダ変数を示します。
固定幅	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

このリリースの『Oracle Databaseアドバンスト・キューイング・ユーザーズ・ガイド』の変更内容

ここでは、次の内容について説明します。

- [Oracle Database Advanced Queuingリリース19c, バージョン19.2の変更内容](#)
- [Oracle Database Advanced Queuing 12cリリース 2 \(12.2\)の変更内容](#)
- [Oracle Database Advanced Queuing 12cリリース1 \(12.1.0.2\)の変更内容](#)
- [Oracle Database Advanced Queuing 12cリリース1 \(12.1\)の変更内容](#)

Oracle Database Advanced Queuingリリース19c, バージョン19.2の変更内容

Oracle Databaseリリース19c, バージョン19.2の『Oracle Databaseアドバンスト・キューイング・ユーザーズ・ガイド』における変更点は次のとおりです。

新機能

このリリースの新機能は次のとおりです。

- 拡張キーベースのメッセージング

AQシャード・キューのパフォーマンスはAQクラシック(非シャード)キューと比較して大幅に向上しています。これは、データベースの特定のインスタンスが各シャードを所有するように、キューがシャード化されているためです。デフォルトでは、エンキュー操作とデキュー操作に関して、シャード化はユーザーに対して完全に透過的です。AQでは、適切なシャードにメッセージを内部的に配置して、最大のパフォーマンスおよびJMS仕様で求められるセッション・レベルの順序付けを実現しています。セッション・レベルの順序付けでは、同じセッションでメッセージがエンキューされ、優先度と配信モードが同じである場合、2つのメッセージがエンキュー順序の逆順でデキューされることはありません。

場合によっては、ユーザー・アプリケーションでシャード化を制御する場合があります。メッセージがエンキューされるシャードをユーザー・アプリケーションで選択できます。アプリケーション・ロジックをサポートするために、シャード・キュー内のメッセージをシャードする方法を必要に応じてユーザーが決定できます。AQシャード・キューのパフォーマンスおよび順序付けの利点は、シャード化をユーザーが制御する場合でもそのまま維持されます。アプリケーションでは、次のことを制御できます。

- キューのシャード数
- キーベースのエンキュー: エンキュー・セッションでエンキューするときにメッセージとともにキーを指定して、メッセージがエンキューされるキューのシャードを選択できます。AQサーバーにより、キーのすべてのメッセージが同じシャードにエンキューされます。1つのシャードは、異なるキーのメッセージを処理できます。
- スティック・デキュー: シャードは、シングル・コンシューマ・キューまたはJMSキューのためのアクティブなデキュー・セッションを1つのみ持つことができます。同様に、シャードは、マルチ・コンシューマ・キューまたはJMSトピックのためのデキュー・セッションをサブスクライバごとに1つのみ持つことができます。そのデキュー・セッションは、セッションの存続期間中はキューのそのシャードに固定されたままになります。そのような機能は、JMSリスナーにも使用できます。

関連項目:

[ユーザー・シャーディング](#)

サポート対象外機能

次の機能は、このリリースではサポートされなくなりました。

- Oracle Streamsのサポート終了

Oracle Database 19c以降では、Oracle Streams機能はサポートされなくなりました。Oracle GoldenGateを使用して、Oracle Streamsのすべてのレプリケーション機能を置き換えます。

Oracle Database Advanced Queuing 12cリリース2 (12.2)の変更内容

Oracle Database 12cリリース2 (12.2)の『Oracle Databaseアドバンスド・キューイング・ユーザーズ・ガイド』における変更点は次のとおりです。

新機能

このリリースの新機能は次のとおりです。

- シャード・キューのJMSペイロードおよびJMS以外(ADTまたはRAW)のペイロードに対するPL/SQLエンキューおよびデキューのサポート

Oracle Database 12cリリース2 (12.2)は、シャード・キューでJMS、ADTおよびRAWペイロードに対してエンキューおよびデキュー操作を実行するように、PL/SQL APIを拡張およびサポートします。PL/SQL配列APIもシャード・キューをサポートします。既存のJMS以外の多くのアプリケーションで、ほとんど変更することなくシャード・キューを使用できるようになりました。

Oracle Database 12cリリース2 (12.2)以降では、シャード・キューを使用するJMSユーザーは、PL/SQL通知を使用してPL/SQLプロシージャを登録し、正常なエンキューでAQサーバーによって自動的に呼び出されるようにできます。PL/SQL通知を使用すると、サーバーでメッセージを自動的にデキューおよび処理できるため、クライアントがキューでメッセージをポーリングする必要がなくなります。

詳細は、[「シャード・キューの管理」](#)を参照してください。

- シャード・キューの診断性および管理性

12cリリース2 (12.2)以降では、AQシャード・キューはSTREAMS_POOLのメモリー割当ておよびスループットを最適化するためのメッセージ・キャッシュのアドバイザ、ビューおよび自動管理を提供しています。

詳細は、[シャード・キューのチューニング](#)を参照してください。

- Oracle Database Advanced Queuingのより長い識別子

12cリリース2 (12.2)以降では、AQキュー名の最大長が122バイトに増加しています。サブスクリバ名および受信者名の最大長が128文字に増加しています。AQルール・エンジンの場合、ルール名およびルール・セット名の最大長が128バイトになりました。

Oracle Database Advanced Queuing 12cリリース1 (12.1.0.2)の変更内容

Oracle Database 12c リリース1 (12.1.0.2)の『Oracle Databaseアドバンスド・キューイング・ユーザーズ・ガイド』における変更点は次のとおりです。

新機能

このリリースの新機能は次のとおりです。

- JMSストリーミング

Oracle Database 12c リリース1 (12.1.0.2)では、Advanced Queuingは、大量のメッセージ・データまたはペイロードを送受信するアプリケーション用に、AQjmsBytesMessageおよびAQjmsStreamMessageを介した共有キューのエンキューおよびデキューでのJMSストリーミングを導入しました。

詳細は、[「JMSストリーミング」](#)を参照してください。

Oracle Database Advanced Queuing 12cリリース1 (12.1)の変更内容

Oracle Database 12c リリース1 (12.1)の『Oracle Databaseアドバンスド・キューイング・ユーザーズ・ガイド』における変更点は次のとおりです。

新機能

このリリースの新機能は次のとおりです。

- JMSシャード・キュー

Oracle Database 12cリリース1 (12.1)のAdvanced Queuingは、高いパフォーマンスと可用性を持つシャードJMSキューを導入しています。シャード・キューとは、システムで管理されているパーティション化を使用して、独立した複数の物理キューに分割される1つの論理キューです。[シャード・キュー](#)により、異なるキュー・シャードにある2つのメッセージ間の順序付けがベスト・エフォートで実行されるため、エンキューおよびデキューのスループットが、特にOracle RACインスタンスで向上します。各シャードは、セッション内のエンキュー時刻に基づいて順序付けされます。シャード・キューは、エンキュー元およびデキュー元がそれら同士で競合しないように、表のパーティションを自動的に管理します。また、シャード・キューは、インメモリーのメッセージ・キャッシュを使用して、パフォーマンスを最適化し、AQ-JMSのエンキューおよびデキューのディスクおよびCPUのオーバーヘッドを削減します。シャード・キューは、Oracle RACインスタンス全体で使用されるキュー、エンキュー率またはデキュー率の高いキュー、またはサブスクリバの多いキューについて推奨されるJMSキューです。

12.2では、JMS以上をサポートするようにシャード・キューが拡張されました。詳細は、[「シャード・キュー」](#)を参照してください。

- 結果キャッシュ拡張

Oracle Database 12cリリース1 (12.1)では、ルール・エンジンに、共通で使用される多数のルールのパフォーマンスを向上させる[結果キャッシュ](#)が導入されました。結果キャッシュでは、属性が同一の式が以前に評価されている場合は、評価フェーズが省略されます。ルールの結果が確定的でない可能性がある場合や、評価されないルールがある場合、または属性に非スカラー・データ型が含まれる場合などには、すべてのルールがキャッシュされるわけではありません。アドバンスド・キューでは、サブスクリプションやそのデキュー・セッションの存続期間が長い場合に、キャッシュが非常に便利で

す。

- LONG VARCHARのサポート

Oracle Database 12cリリース1 (12.1)のOracle Database Advanced Queuingでは、LONG VARCHARデータ型がサポートされています。

- 3層バックグラウンド・アーキテクチャ

Oracle Database 12cリリース1 (12.1)には、新しい3層設計のAQバックグラウンド・プロセス・アーキテクチャが導入されています。

詳細は、[「AQバックグラウンド・アーキテクチャ」](#)を参照してください。

- Data Guardデータベースのローリング・アップグレードのサポート

Oracle Database Advanced Queuingを使用するデータベースを、Data Guardデータベースのローリング・アップグレードを使用して、新しいOracleデータベースのリリースおよびパッチ・セットにアップグレードできるようになりました (一時ロジカル・スタンバイ・データベースのみ)。ローリング・アップグレードは、Oracle Database 12cリリース1 (12.1)からサポートされるようになりました。

Data Guardデータベース・ローリング・アップグレードにより、ローリング方式で新しいデータベース・リリースまたはパッチ・セットへのアップグレードを行うことができ計画停止時間が短縮されます。このようなアップグレードでのデータベースの合計停止時間は、Data Guardスイッチオーバーを実行するために必要な短い時間に制限されます。

次のパッケージを使用すると、ロジカル・スタンバイを使用したローリング・アップグレードがサポートされます。

- DBMS_AQ
- DBMS_AQJMS
- DBMS_AQADM(次のプロシージャを除く):
 - SCHECULE_PROPAGATION
 - UNSCHEDULE_PROPAGATION
 - ALTER_PROPAGATION_SCHEDULE
 - ENABLE_PROPAGATION_SCHEDULE
 - DISABLE_PROPAGATION_SCHEDULE

関連項目:

- Oracle Database AQパッケージの詳細は、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください

1 Oracle Database Advanced Queuingの概要

アドバンスド・キューイング(AQ)は、堅牢で豊富な機能を持つメッセージ・キューイング・システムであり、Oracle Databaseに統合されています。次のトピックでは、Oracle Database Advanced Queuing (AQ)、および統合環境での複雑な情報処理の要件について説明します。

- [キューイングとは](#)
- [Oracle Database Advanced QueuingによるOracle Databaseの活用](#)
- [統合アプリケーション環境でのOracle Database Advanced Queuing](#)
- [バッファ済メッセージ](#)
- [非同期通知](#)
- [エンキュー機能](#)
- [デキュー機能](#)
- [伝播機能](#)
- [メッセージ・フォーマットの変換](#)
- [その他のOracle Database Advanced Queuing機能](#)
- [Oracle Database Advanced Queuingへのインタフェース](#)

キューイングとは

アドバンスド・キューイングでは、ユーザー・メッセージがキューと呼ばれる抽象的な記憶単位に保存されます。Webベースのビジネス・アプリケーションが相互に通信する場合、プロデューサ・アプリケーションがメッセージをエンキューし、コンシューマ・アプリケーションがメッセージをデキューします。

最も基本的なキューイングでは、1つのプロデューサが、1つのキューに1つ以上のメッセージをエンキューします。各メッセージは、1つのコンシューマによって1回のみデキューされ、処理されます。メッセージは、コンシューマによってデキューされるか、または期限が切れるまで、[キュー](#)内に保持されます。[プロデューサ](#)は、メッセージが使用可能になるまでの遅延および期限切れを指定できます。同様に、[デキュー](#)時に使用可能なメッセージがない場合は、[コンシューマ](#)は待機できます。エージェント・プログラムまたはアプリケーションは、プロデューサおよびコンシューマの両方で動作できます。

プロデューサは、メッセージをどのような順序でも[エンキュー](#)できます。メッセージは、エンキューされた順序でデキューする必要はありません。メッセージは、デキューされなくてもエンキューできます。

やや複雑になると、多数のプロデューサがメッセージをキューにエンキューし、そのすべてのメッセージが1つのコンシューマによって処理されます。あるいは、多数のプロデューサがメッセージをエンキューし、個々のメッセージが型および相関識別子に応じて異なるコンシューマで処理されます。

エンキューされたメッセージは、別のキューで再生成されるときに伝播されると考えられています。別のキューは、同じデータベースまたはリモート・データベースに存在します。

アプリケーションでは一般に、様々なフォーマットのデータが使用されます。[変換](#)は、1つのデータ型から別のデータ型へのマッピングを定義します。変換は、ソース・データ型を入力として取得し、ターゲット・データ型のオブジェクトを戻すSQLファンクションによって表現されます。メッセージの変換は、メッセージがエンキューされる時、デキューされる時、またはリモート・[サブスクリバ](#)に伝播される時に行われるように設定できます。

Oracle Database Advanced QueuingによるOracle Databaseの活用

Oracle Database Advanced Queuingは、データベース統合型のメッセージ・キューイング機能を提供します。この機能では、メッセージを永続的に格納し、異なるコンピュータおよびデータベースにあるキュー間でメッセージを伝播し、Oracle Net ServicesおよびHTTP(S)を使用してメッセージを送信できるように、Oracle Databaseの機能を利用します。

Oracle Database Advanced Queuingはデータベース表内に実装されるため、高可用性、拡張性および信頼性という操作上のすべてのメリットが、キュー・データに適用されます。Oracle Database Advanced Queuingでは、リカバリ、再起動、セキュリティなどの標準のデータベース機能がサポートされます。Oracle Enterprise Managerなどのデータベース開発ツールおよびデータベース管理ツールを使用して、キューを監視できます。他のデータベース表と同様に、キュー表もインポートおよびエクスポートできます。同様に、アドバンスド・キューはOracle Data Guardによってサポートされており、ステートレスな中間層を使用する場合にメッセージを保持するために重要であることがある高可用性が提供されます。エンキューおよびデキューは、データベース内にあることにより、分散トランザクションを必要とせずにデータベース・トランザクションに組み入れることができます。(分散トランザクションの例としては、Oracle XAがあります。)

標準のSQLを使用してメッセージを問い合わせることができます。ユーザーはSQLを使用して、メッセージのプロパティ、履歴およびペイロードにアクセスできます。SQLアクセスの場合は、メッセージの監視と追跡も実行できます。索引機能などの使用可能なすべてのSQLテクノロジーを使用してメッセージへのアクセスを最適化できます。

注意:



Oracle Database Advanced Queuing は、キュー表に対するデータ操作言語(DML)操作、対応する索引構成表(IOT)、またはシャード・キューによって使用される関連するシステム・パーティション化表(ある場合)をサポートしていません。キュー表を修正するためにサポートされている唯一の手段は、提供されている API を使用することです。キュー表や IOT は、DML 操作を行うと一貫性がなくなり、実質的に破損する可能性があります。

システム・レベルのアクセス制御

Oracle Database Advanced Queuingは、あらゆるキューイング操作に対するシステム・レベルのアクセス制御に対応しているため、アプリケーション開発者またはDBAが、ユーザーをキュー管理者として指定できます。キュー管理者は、データベースのどのキューに対しても、Oracle Database Advanced Queuingインタフェース(管理および操作)を起動できます。これによって、データベース上のキュー全体に対するすべての管理スクリプトを1つのスキーマで管理できるため、管理作業が容易になります。

キュー・レベルのアクセス制御

Oracle Database Advanced Queuingでは、エンキュー操作およびデキュー操作に対するキュー・レベルのアクセス制御がサポートされています。この機能により、アプリケーション開発者は、あるスキーマで作成されたキューを、別のスキーマで実行されているアプリケーションから保護できます。アプリケーション開発者は、キュー・スキーマ外で実行されているアプリケーションに最低限のアクセス権のみを付与することが可能です。

パフォーマンス

サービスに対するリクエストとサービスの供給を分離することによって、効率を向上させ、複雑なスケジューリングを有効化する必要があります。Oracle Database Advanced Queuingは、次の基準で測定した場合、高いパフォーマンスを示します。

- 1秒間にエンキューおよびデキューされるメッセージ数

- メッセージ・ウェアハウスに対する複雑な問合せの評価にかかる時間
- 障害後にメッセージ処理のリカバリおよび再起動にかかる時間

スケーラビリティ

キューイング・システムはスケーラブルである必要があります。Oracle Database Advanced Queuingは、アプリケーションを使用するプログラム数が増加しても、メッセージ数が増加しても、またメッセージ・ウェアハウスのサイズが増加しても、高いパフォーマンスを実現します。

セキュリティのための永続性

ネットワーク、コンピュータおよびアプリケーションに障害が発生したときに、遅延実行を正常に動作させるには、サービスに対するリクエストで構成されるメッセージを永続的に格納し、確実に1回のみ処理する必要があります。Oracle Database Advanced Queuingは、次の場合に要件を満たすことができます。

- アプリケーションに、外部クライアントまたは内部プログラムから同時に到着する、複数の未処理メッセージを処理するためのリソースがない場合。
- データベース間の通信リンクが、常に使用可能なわけではなく、他の用途に確保されている場合。システム容量が不足しているためにメッセージをすぐに処理できない場合、アプリケーションは、処理可能になるまでそのメッセージを保存する必要があります。
- 外部クライアントまたは内部プログラムが、処理済メッセージを受信する準備ができていない場合。

スケジューリングのための永続性

キューイング・システムは、優先順位を処理する必要があります。この優先順位は、次のように変化する可能性があります。

- 後から到着したメッセージは、先に到着したメッセージより高い優先順位になります。
- メッセージは、後のメッセージを待ってからアクションを実行できます。
- 同一のメッセージに異なるプロセスからアクセスできます。
- 特定のキューにあるメッセージがより重要になり、遅延や他のキューのメッセージからの介入を減らして処理する必要性が生じる場合もあります。
- ある受信者へのメッセージ送信が、他の受信者への送信より優先順位が高くなる場合があります。

メタデータのアクセスおよび分析のための永続性

メッセージのメタデータがペイロード・データと同様に重要になる場合があるため、キューイング・システムはメッセージのメタデータを保存する必要があります。たとえば、メッセージの受信時刻またはディスパッチ時刻が、ビジネスや正当な理由から重要になることがあります。Oracle Database Advanced Queuingの永続性機能を使用すると、最大需要の周期の分析、または注文を受信してから処理を完了するまでのタイムラグの評価を行うことができます。

オブジェクト型のサポート

Oracle Database Advanced Queuingでは、キューのタイプが抽象データ型(ADT)の場合のエンキュー、デキュー、および伝播操作をサポートしています。また、タイプが基本ADTの継承タイプの場合もエンキューおよびデキュー操作をサポートしています。型が基本ADTから継承されている2つのキュー間での伝播はサポートされていません。

Oracle Database Advanced Queuingは、ANYDATAキューもサポートし、このキューを使用すると、アプリケーションは1つのキュー内で様々なメッセージ型をエンキューできます。Oracle Database Advanced Queuingは、12cリリース1 (12.1)以降のLONG VARCHARデータ型をサポートしています。

ユーザー定義型メッセージをエンキュー、伝播またはデキューする場合は、これらのメッセージで使用される各型が、メッセージをキューにエンキューできる各データベースに存在する必要があります。環境によっては有向ネットワークを使用して、中間データ

ベースを介してメッセージをルーティングしてから宛先に配信する場合があります。そのような環境の場合は、対応する型のメッセージが特定の間データベースにエンキューまたはデキューされない場合でも、各中間データベースにその型が存在する必要があります。

さらに、各型に対して次の要件が満たされる必要があります。

- 型名が各データベースで同じであること。
- 型が各データベースの同じスキーマにあること。
- 型の形状が各データベースで完全に一致していること。
- どのデータベースにおいても、型に継承または型進化を使用できないこと。
- 型に、VARRAY、ネストした表、LOB、ROWIDまたはUROWIDが含まれないこと。

オブジェクト識別子は各データベースで一致する必要はありません。

構造化ペイロードとXMLTypeペイロード

ユーザーは、オブジェクト型を使用してメッセージ・ペイロードを構造化および管理できます。一般にリレーショナル・データベース・システムは、メッセージ・システムより豊富な型指定のシステムを備えています。Oracle Databaseはオブジェクト・リレーショナル・データベース・システムであるため、従来のリレーショナル型とユーザー定義型をサポートします。強力な型指定を持つ内容(外部の型指定システムによって定義されたフォーマットを持つ内容など)によって、多くの高性能な機能が使用できます。これらの機能には、次のものがあります。

- 内容ベースのルーティング

Oracle Database Advanced Queuingで内容を調査し、その内容に基づいてメッセージを別のキューに自動的にルーティングできます。

- 内容ベースのサブスクリプション

パブリッシュ・サブスクライブ・システムがメッセージ・システム上に組み込まれているため、内容に基づいてサブスクリプションを作成できます。

- 問合せ

メッセージの内容を問い合わせることができるため、メッセージ・ウェアハウスが可能です。

新しい不透明型、XMLTypeを使用するキューを作成できます。これらのキューは、XML文書であるメッセージを転送および格納するために使用できます。XMLTypeを使用すると、次の操作を実行できます。

- 任意の型のメッセージをキューに格納する。
- 複数の型のペイロードをキューに格納する。
- `ExistsNode()` 演算子を使用してXMLType列を問い合わせる。
- サブスクライブ・ルールまたはデキュー条件にその演算子を指定する。

Oracle Internet Directoryとの統合

キューに関するシステム・イベント、ユーザー・イベントおよび通知をOracle Internet Directoryに登録できます。システム・イベントは、データベースの起動、データベースの停止およびシステム・エラー・イベントです。ユーザー・イベントには、ユーザー・ログインとユーザー・ログオフ、DDL文(CREATE、DROP、ALTER)およびDML文トリガーがあります。キューに関する通知には、OCI通知、PL/SQL通知および電子メールによる通知があります。

Oracle Database Advanced Queuingエージェントの別名をOracle Internet Directoryに作成することもできます。これらの別名は、Oracle Database Advanced Queuing操作(エンキュー、デキューおよび通知)の実行中に指定できま

す。これは、内部エージェント名を非公開にする必要があるときに有効です。



注意:

シャード・キューは、12c リリース 2 (12.2.0.1)の OCI および Thick ドライバをサポートしていません。

Oracle Real Application Clusters(Oracle RAC)のサポート

Oracle Real Application Clusters (Oracle RAC)を使用すると、異なるキューを別々のインスタンスによって管理できるようにすることでOracle Database Advanced Queuingパフォーマンスを改善できます。このためには、キューを格納するキュー表に様々なインスタンス・アフィニティ(作業環境)を指定します。これによって、様々なキューに対する操作(エンキューおよびデキュー)を平行で行うことができるようになります。シャード・キューは、複数のOracle RACインスタンスにエンキュー元またはデキュー元のあるアプリケーションに使用することをお勧めします。詳細は、[「シャード・キューおよびOracle Real Application Clusters \(Oracle RAC\)」](#)を参照してください。

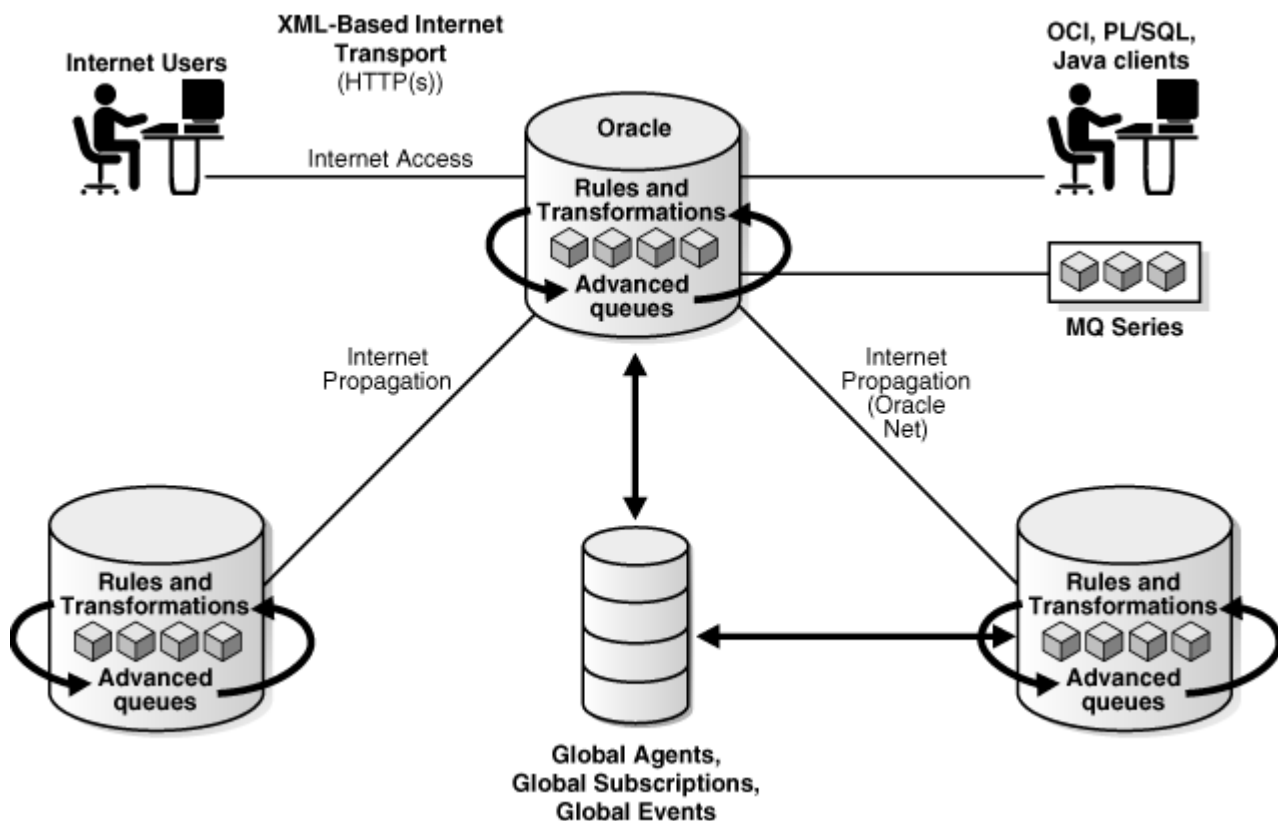
互換性をOracle8iリリース8.1.5以上に設定すると、アプリケーションはキュー表に対してインスタンス・アフィニティを指定できません。Oracle Database Advanced QueuingをOracle RACと複数のインスタンスで使用する場合、この情報を使用してインスタンス間のキュー表をパーティション化して、伝播用のみでなくキュー・モニターのスケジューリングも行うことができます。キュー表は、ユーザーが指定したインスタンスのキュー・モニターによって監視されます。キュー表の所有が終了すると、セカンダリ・インスタンスまたはいずれかの使用可能なインスタンスがキュー表の所有権を引き継ぎます。

インスタンス・アフィニティが指定されないと、キュー表は、使用可能なインスタンス間で任意にパーティション化されることとなります。これにより、キュー表にアクセスするアプリケーションとそれを監視するキュー・モニターの間で、pingが発生する可能性があります。インスタンス・アフィニティを指定すると、これを回避できますが、そのアプリケーションが他のインスタンスから、キュー表とそのキューにアクセスできなくなることはありません。

統合アプリケーション環境でのOracle Database Advanced Queuing

Oracle Database Advanced Queuingを使用すると、アプリケーションの統合に必要なメッセージ管理と通信が可能になります。統合環境では、メッセージはOracle Databaseサーバーとアプリケーション間およびOracle Databaseサーバーとユーザー間を移動します。これを示したのが[図1-1](#)です。

図1-1 Oracle Database Advanced Queuingを使用した統合アプリケーション環境



メッセージは、Oracle Net Servicesを使用して、クライアントとOracle Databaseサーバー間または2つのOracle Databaseサーバー間で交換されます。Oracle Net Servicesはまた、あるOracle Databaseキューから別のOracle Databaseキューへのメッセージの伝播も行います。さらに、[図1-1](#)に示すように、Oracle Database Advanced Queuingの操作は、HTTP(S)を使用してインターネット経由で行うこともできます。この場合、クライアント、ユーザーまたはインターネット・アプリケーションは構造化XMLメッセージを生成します。インターネット経由の[伝播](#)では、Oracle Databaseサーバーも構造化XMLを使用して通信します。

アプリケーションの統合には、異機種間のメッセージ・システムの統合も含まれます。Oracle Database Advanced Queuingは、メッセージ・ゲートウェイを介して、IBM Websphere MQなどのOracle Database以外の既存のメッセージ・システムと透過的に統合されるため、既存のWebsphere MQベースのアプリケーションもOracle Database Advanced Queuing環境に統合できます。

トピック:

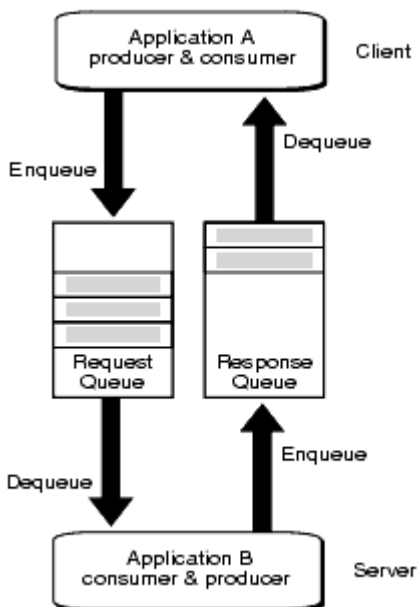
- [Oracle Database Advanced Queuingのクライアント/サーバー通信](#)
- [マルチ・コンシューマによる同一メッセージのデキュー](#)
- [Oracle Database Advanced Queuingを使用したワークフローの実装](#)
- [Oracle Database Advanced Queuingを使用したパブリッシュ・サブスクライブの実装](#)

Oracle Database Advanced Queuingのクライアント/サーバー通信

Oracle Database Advanced Queuingは、クライアント/サーバー・アプリケーションが通常実行される同期形式の代替として非同期を提供しています。この図は、Oracle Database Advanced Queuingを使用したクライアント/サーバー通信の例を示しています。

この例では、アプリケーションB(サーバー)は、リクエスト・キューまたはレスポンス・キューを使用して、アプリケーションA(クライアント)にサービスを提供しています。

図1-2 Oracle Database Advanced Queuingを使用したクライアント/サーバー通信



アプリケーションAは、リクエストをリクエスト・キューにエンキューします。別のトランザクションで、アプリケーションBは、リクエストをデキューして処理します。アプリケーションBは、その結果をレスポンス・キューにエンキューし、アプリケーションAは引き続き別のトランザクションで、エンキューされた結果をデキューします。

クライアントはサーバーとの接続の確立を待つ必要はなく、サーバーは独自の処理タイミングでメッセージをデキューします。サーバーによるメッセージ処理が終了したとき、クライアントは結果を受け取るまで待つ必要はありません。このような二重遅延処理により、クライアントとサーバーはどちらも自由に処理を実行できます。

マルチ・コンシューマによる同一メッセージのデキュー

メッセージは、一度に1つのキューにしか入れることができません。複数のコンシューマに送信するために、プロデューサから同じメッセージを複数のキューに挿入する必要がある場合は、非常に大量のキューを管理する必要があります。複数のコンシューマが同じメッセージをデキューできるように、Oracle Database Advanced Queuingにはキューのサブスクライバとメッセージの受信者が用意されています。

[サブスクライバ](#)・リストおよび[受信者](#)リストを可能にするために、複数のコンシューマ・オプションを使用して作成された[キュー表](#)にキューを常駐させる必要があります。各メッセージは、所定のすべてのコンシューマによって処理されるまで、キュー内にとどまります。

キューのサブスクライバ

複数のコンシューマ(アプリケーションまたはその他のキュー)は、サブスクライバとして1つのキューに関連付けることができます。これにより、キューにエンキューされたすべてのメッセージを、それぞれのキューのサブスクライバが使用できるようになります。キューに対応するサブスクライバは、メッセージまたはメッセージ・プロデューサを変更することなく、動的に変更できます。

シングル・コンシューマ・キューまたは例外キューには、サブスクリプションを追加できません。あるキューにサブスクライバとして追加されたコンシューマは、サブスクライバが追加された後にエンキューされたメッセージのみデキューできます。2つのサブスクライバが名前、アドレスおよびプロトコルに同じ値を持つことはできません。この3つの属性のうち少なくとも1つは、2つのサブスクライバに対して異なる値にする必要があります。

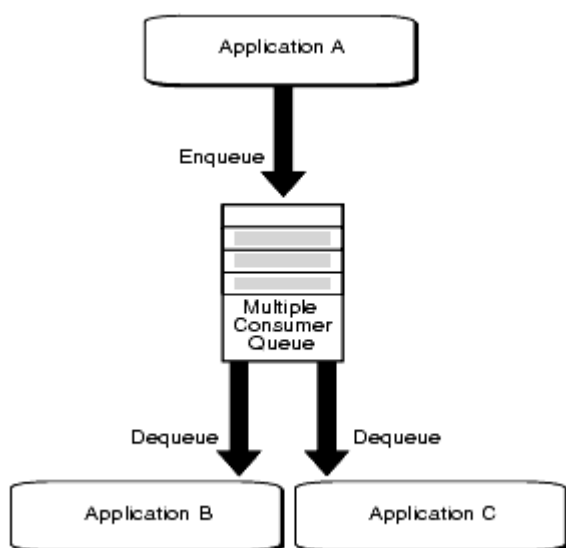
サブスクライバ間には優先順位がないため、どのサブスクライバがどのメッセージをどのような順番でデキューするかはわかりません。つまり、サブスクライバによるデキューの順序は予想できません。

サブスクライバをルールベースにすることもできます。SQL問合せのWHERE句の構文と同様、ルールはメッセージ・プロパティまたはメッセージ内容を示す属性によって表現されます。このようなサブスクライバ・ルールが、着信メッセージに対して評価され、一致したルールを使用してメッセージ受信者が判断されます。

[図1-3](#)では、アプリケーションBとアプリケーションCのそれぞれがアプリケーションAで生成されたメッセージを必要としているため、ア

アプリケーションBとアプリケーションCにはキューのサブスクライバとして、マルチ・コンシューマ・キューが特別に構成されています。これにより、それぞれのアプリケーションで、キューに格納されたあらゆるメッセージを受信できるようになります。

図1-3 マルチ・コンシューマ・キューを使用した通信



メッセージの受信者

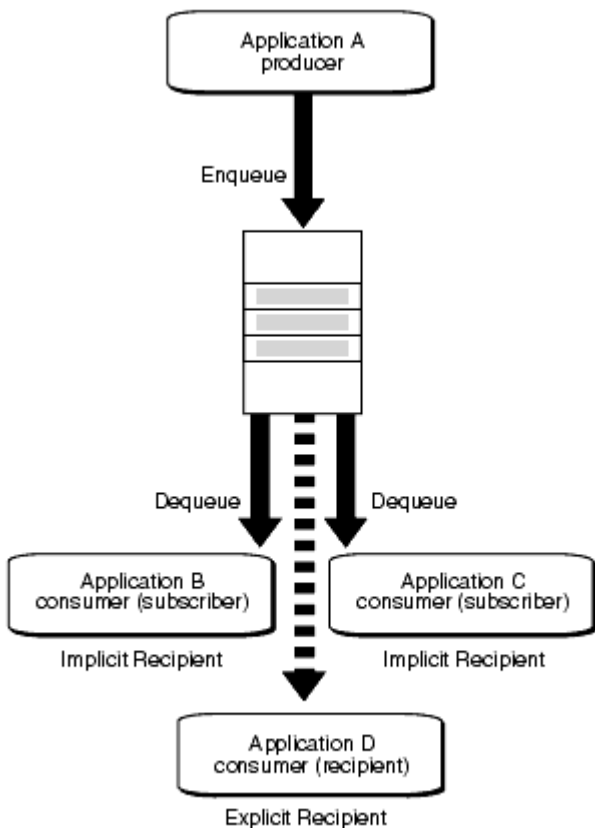
メッセージ・プロデューサは、メッセージが非シャード・キューにエンキューされた時点で受信者リストを送ることができます。これにより、キュー内の各メッセージと一連の受信者を一意に対応させることができます。メッセージに対応付けられた受信者リストは、キューに対応付けられたサブスクライバ・リストが存在する場合には、それをオーバーライドします。受信者がサブスクライバ・リストに指定されている必要はありません。ただし、受信者はサブスクライバの中から選択できます。

受信者は名前のみで指定できますが、その場合、その受信者が、メッセージをエンキューされたキューからデキューする必要があります。名前およびアドレスで指定し、プロトコル値を0にできます。アドレスは、同じデータベース内の別のキューの名前か、Oracle Databaseの別のインストール(データベース・リンクで識別される)の名前にする必要があります。この場合は、メッセージは指定されたキューに伝播され、指定された名前の子シューマによってデキューされます。受信者の名前がNULLの場合、メッセージはアドレスに指定されたキューに伝播され、アドレスに指定されたキューのサブスクライバによってデキューされます。プロトコル・フィールドの値が0(ゼロ)でない場合、名前およびアドレスのフィールドはシステムにより無視され、メッセージは特定のコンシューマによってデキューされます。

キューへのサブスクライブは雑誌の購読と似ており、各サブスクライバは特定のキューに格納されたすべてのメッセージをデキューすることが可能で、これは雑誌の購読者がすべての記事にアクセスできることと類似しています。ただし、受信者は手紙を受け取る場合と似ており、各受信者が特定のメッセージの指定されたターゲットです。

図1-4に、Oracle Database Advanced Queuingが両方の種類のコンシューマに対応する方法を示します。アプリケーションAがメッセージをエンキューします。アプリケーションBおよびアプリケーションCはサブスクライバです。しかし、メッセージには、明示的にアプリケーションDのような受信者宛に送信されているものがあり、アプリケーションDはキューのサブスクライバである場合とそうでない場合があります。特定のメッセージのこのような受信者のリストは、そのメッセージのエンキュー・コールで指定されています。これは、そのキューのサブスクライバのリストよりも優先されます。

図1-4 メッセージの明示的および暗黙的な受信者



注意:



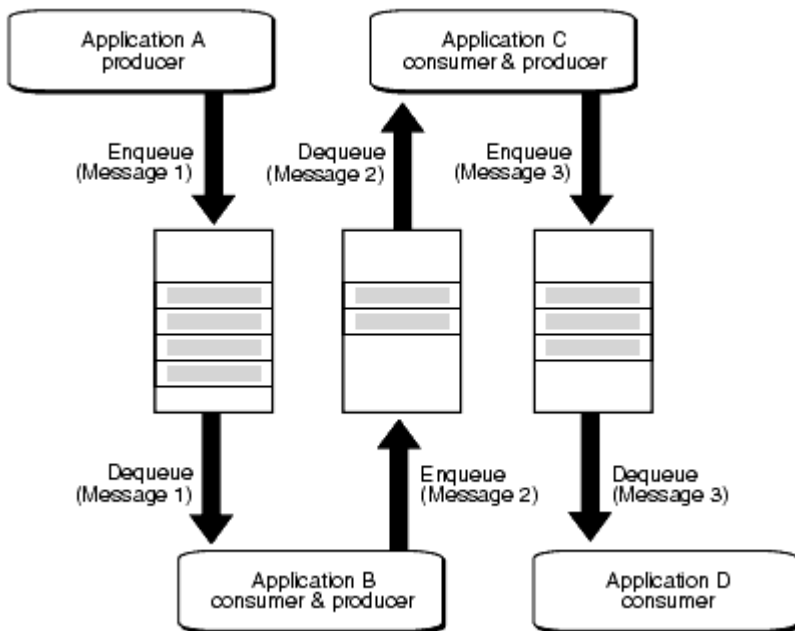
複数のプロデューサによって、異なる受信者に宛てられたメッセージが同時にエンキューされる場合があります。

Oracle Database Advanced Queuingを使用したワークフローの実装

Oracle Database Advanced Queuingを使用すると、ワークフロー(チェーン・アプリケーション・トランザクションとも呼ばれます)を実装できます。図は、Oracle Database Advanced Queuingを使用してワークフローを実装する方法を例示します。

1. [ワークフロー](#)は、アプリケーションAがメッセージ1をエンキューすることによって開始されます。
2. アプリケーションBはそのメッセージをデキューして、必要なアクティビティをすべて実行し、メッセージ2をエンキューします。
3. アプリケーションCはメッセージ2をデキューして、メッセージ3を生成します。
4. アプリケーションDはワークフローの最後のステップとして、メッセージ3をデキューします。

図1-5 Oracle Database Advanced Queuingを使用したワークフローの実装



注意:



メッセージ 1、2 および 3 の内容は同一の場合も異なる場合もあります。異なるメッセージの場合でも、あるメッセージに前のメッセージの内容が一部含まれることがあります。

キューは、ビジネス・プロセスの各段階での情報の流れをバッファするために使用されます。メッセージの遅延間隔および期限切れを指定することで、各アプリケーションに実行枠を設定できます。

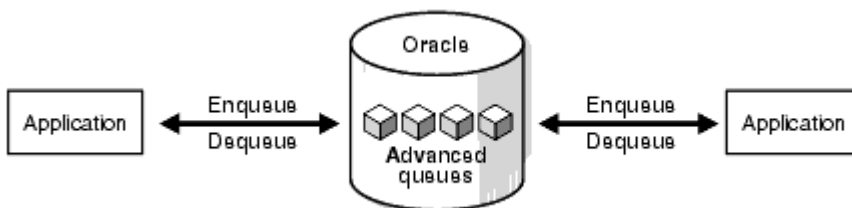
ワークフローの観点から、メッセージ・フローのボリュームとタイミングに関する知識は、ペイロード・データの価値に加えて1つの業務資産となります。Oracle Database Advanced Queuingでは、この知識を得るための手段として、メッセージを保存できるオプションを用意することで、過去の傾向を分析して今後の動向を予測できるようにしています。

Oracle Database Advanced Queuingを使用したパブリッシュ・サブスクライブの実装

Point-to-Pointメッセージは、特定のターゲットを対象としています。送信者および受信者は、メッセージ交換に使用する共通キューを決定します。各メッセージは、1人の受信者によってのみ処理されます。

[図1-6](#)は、各アプリケーションにシングル・コンシューマ・キューと呼ばれる独自のメッセージ・キューがあることを示しています。

図1-6 Point-to-Pointメッセージ

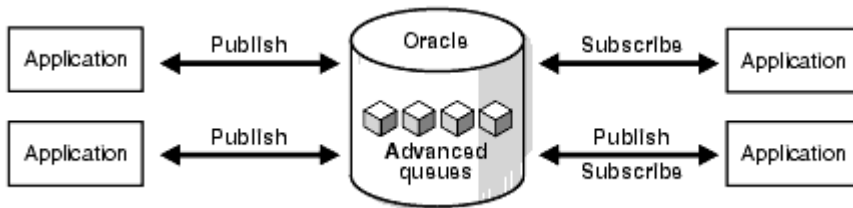


[パブリッシュ・サブスクライブ](#)・メッセージは、[図1-7](#)に示すように、複数の受信者によって使用される場合があります。パブリッシュ・サブスクライブ・メッセージ機能には、伝播範囲が広い[ブロードキャスト](#)と呼ばれるモードと、伝播範囲が狭い[マルチキャスト](#)と呼ばれるモードがあります。

ブロードキャストは、特定のプログラムの視聴者が誰かはっきりとわからないラジオ局に似ています。デキュー元は、マルチ・コン

シューマ・キューのサブスクライバです。これに対し、マルチキャストは、誰が購読者であるかがわかっている雑誌の発行元と似ています。マルチキャストはPoint-to-Multipointとも呼ばれます。これは単一のパブリッシャがメッセージを受信者と呼ばれる複数のレシーバに送信するため、この受信者は交換メカニズムとして機能するキューのサブスクライバである場合とそうでない場合があります。

図1-7 パブリッシュ・サブスクライブ・モード



パブリッシュ・サブスクライブとは、パブリッシャ・アプリケーションが匿名で(受信者を指定せずに)キューにメッセージをエンキューする状況を表しています。エンキューされたメッセージは、アプリケーションごとに指定されたルールに基づいてサブスクライバ・アプリケーションに配信されます。ルールは、メッセージ・プロパティ、メッセージ・データの内容、またはその両方に対して定義できます。

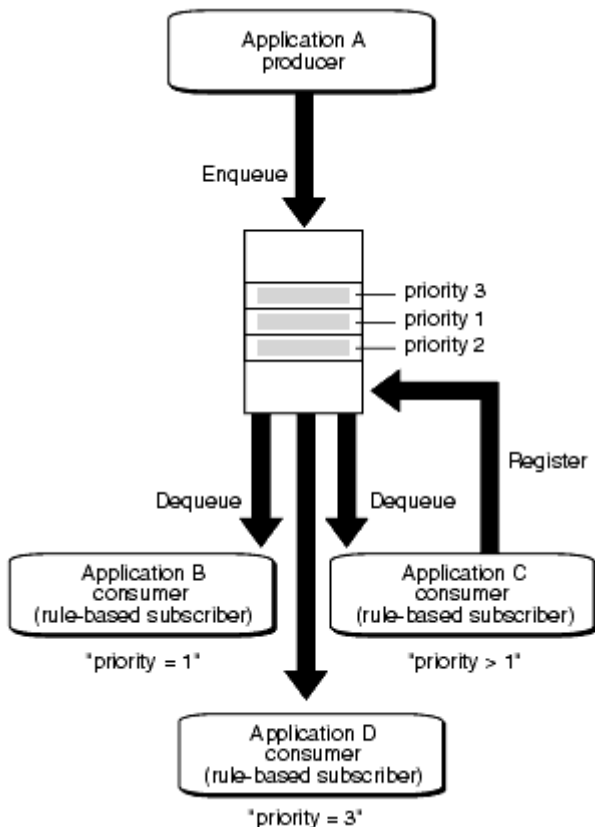
Oracle Database Advanced Queuingを使用した、パブリッシュ・サブスクライブ・モデルの通信を実装するには、次の手順に従います。

1. メッセージを保持するために1つまたは複数のキューを設定します。このキューは、関心がある領域またはサブジェクトを表しています。たとえば、あるキューは請求済注文を表すために使用される可能性があります。
2. ルールベースのサブスクライバを1組設定します。各サブスクライバは、受信を希望するメッセージの仕様(ルール)を指定できます。NULLルールは、サブスクライバがすべてのメッセージの受信を希望することを意味します。
3. パブリッシャ・アプリケーションが、エンキュー・コールをコールしてキューにメッセージをパブリッシュします。
4. サブスクライバ・アプリケーションは、DEQUEUEコールでメッセージを受信できます。これにより、サブスクリプションの基準に合うメッセージが取り出されます。
5. サブスクライバ・アプリケーションは、LISTENコールを使用して、様々なキューに対するサブスクリプションを複数のキューで監視することもできます。これは、サブスクライバ・アプリケーションが多数のキューにサブスクライブしていて、どのキューに届いたメッセージでも受信する場合には、非常にスケーラブルなソリューションです。
6. サブスクライバ・アプリケーションは、Oracle Call Interface(OCI)通知メカニズムを使用することもできます。これによって、プッシュ・モードのメッセージ配信が可能になります。サブスクライバ・アプリケーションは、メッセージの受信元となるキュー(およびサブスクライブ・エージェントとして指定されるサブスクリプション)を登録します。これによって、サブスクリプションに一致するメッセージが届いたときに、コールされるコールバックが登録されます。

図1-8は、Oracle Database Advanced Queuingを使用して、パブリッシャ・アプリケーションAと、サブスクライバ・アプリケーションB、C、Dとの間にパブリッシュ・サブスクライブ関係を実装する方法を示しています。

- アプリケーションBはルール「priority = 1」に基づいてサブスクライブします。
- アプリケーションCはルール「priority > 1」に基づいてサブスクライブします。
- アプリケーションDはルール「priority = 3」に基づいてサブスクライブします。

図1-8 Oracle Database Advanced Queuingを使用したパブリッシュ・サブスクライブの実装



アプリケーションAがそれぞれ優先順位1、2および3に基づいて3つのメッセージをエンキューすると、各メッセージは次のように配信されます。

- アプリケーションBは、1つのメッセージ(優先順位が1)を受信します。
- アプリケーションCは、2つのメッセージ(優先順位が2、3)を受信します。
- アプリケーションDは、1つのメッセージ(優先順位が3)を受信します。

バッファ済メッセージ

バッファ済メッセージは、この製品が常に提供してきた豊富な機能と大幅に速いキュー実装を組み合わせたものです。バッファ済メッセージは、Oracle Database Advanced Queuing永続メッセージの信頼性とトランザクション・サポートを必要としないアプリケーションに理想的です。

バッファ済メッセージが永続メッセージよりも高速なのは、メッセージが共有メモリーに常駐するためです。通常、ディスクに書き込まれるのは、バッファ済メッセージのメモリー使用量合計が使用可能な共有メモリーの上限に達した場合のみです。

注意:



メモリー内でバッファ済メッセージを格納するキューの部分は、バッファ済キューと呼ばれることもあります。

バッファ済メッセージでは、メッセージの保存はサポートされていません。

バッファ済メッセージを使用する場合は、次のいずれかを実行することをお勧めします。

- `streams_pool_size`パラメータの設定

このパラメータは、Oracle Database Advanced Queuingで使用可能な共有メモリーのサイズを制御します。指定しなかった場合は、共有プール・サイズの最大10%がデータベース・キャッシュからOracle Database Advanced

Queuingプールに割り当てられます。

シャード・キューでのバッファ済メッセージのメッセージ・キャッシュの設定の詳細は、[シャード・キューの手動チューニング](#)を参照してください。

- **SGA自動チューニングの有効化**

Oracle Database Advanced Queuingには、Oracle Database Advanced Queuingの使用量だけでなく、SGAを使用するその他のコンポーネントの使用量にも基づいて、適切な量のメモリーがSGAから自動的に割り当てられます。その他のコンポーネントには、バッファ・キャッシュ、ライブラリ・キャッシュなどがあります。streams_pool_sizeを指定した場合は、下限として使用されます。

トピック:

- [バッファ済メッセージのエンキュー](#)
- [バッファ済メッセージのデキュー](#)
- [バッファ済メッセージの伝播](#)
- [フロー制御](#)
- [Oracle Real Application Clusters\(Oracle RAC\)を持つバッファ済メッセージ](#)
- [バッファ済メッセージの制限](#)
- [エラー処理](#)

バッファ済メッセージのエンキュー

バッファ済メッセージと永続メッセージは、同じシングル・コンシューマ・キューまたはマルチ・コンシューマ・キュー、および同じ管理インタフェースと操作インタフェースを使用します。これらは、メッセージをOracle Database Advanced Queuingキューにエンキューするときにアプリケーションで設定される、配信モード・パラメータによって区別されます。

バッファ済メッセージ・エンキューでは、受信者リストがサポートされています。

バッファ・メッセージングは、8.1以上の互換性で作成されたすべてのキュー表でサポートされます。このリリースのバッファ済メッセージでは、グループ化トランザクション・キューおよび配列のエンキューはサポートされていません。配列エンキュー・プロシージャを使用してバッファ済メッセージをエンキューすることはできますが、配列サイズを1に設定する必要があります。

バッファ済メッセージは、AQ\$Queue_Table_Nameビューを使用して問い合わせることができます。バッファ済メッセージは、IN-MEMORYまたはSPILLED状態で表示されます。

バッファ済メッセージのキュー・タイプは、ADT、XML、ANYDATAまたはRAWです。LOB属性を持つADTタイプの場合、NULLのLOB属性を持つバッファ済メッセージのみエンキューできます。

永続メッセージに使用できる順序付けスキームはすべて、バッファ済メッセージにも使用できますが、使用できるのは各メッセージ・クラス内のみです。同じセッション内でエンキューされた永続メッセージとバッファ済メッセージ間の順序付けは、現在はサポートされていません。

バッファ済メッセージのエンキューおよびデキュー操作は、IMMEDIATE可視性モードで実行する必要があります。したがって、これらの操作は別のトランザクションの一部にはなりません。バッファ済メッセージをエンキューする場合、遅延を指定することはできません。

関連項目:

- [メッセージのエンキュー](#)

- [AQ\\$<Queue_Table_Name>: キュー表内のメッセージ](#)
- [エンキューにおけるメッセージの優先順位および順序付け](#)

バッファ済メッセージのデキュー

バッファ済メッセージでは、ルールベースのサブスクリプションがサポートされています。サブスクライバを追加する手順の拡張により、アプリケーションで永続メッセージのみ、バッファ済メッセージのみ、またはその両方を対象とするよう選択できます。

非シャード・キューの場合、バッファ済メッセージで配列のデキューはサポートされていませんが、配列サイズを1メッセージに設定すると、引き続き配列デキュー・プロシージャを使用できます。

アプリケーションのデキューでは、永続メッセージのみ、バッファ済メッセージのみ、または両方のデキューを選択できます。バッファ済メッセージをデキューする場合は、可視性をIMMEDIATEに設定する必要があります。次に示すすべてのデキュー・オプションがサポートされています。

- デキュー・モードBROWSE、LOCK、REMOVEおよびREMOVE_NO_DATA
- ナビゲーション・モードFIRST_MESSAGEおよびNEXT_MESSAGE
- 相関識別子
- デキューの条件
- メッセージ識別子

関連項目:

- [サブスクライバの追加](#)
- [デキュー・オプション](#)

バッファ済メッセージの伝播

バッファ済メッセージの伝播がサポートされています。1つの伝播スケジュールで永続メッセージとバッファ済メッセージの両方が処理されます。DBA_QUEUE_SCHEDULESビューには、統計およびエラー情報が表示されます。

Oracle Database AQは、バッファ済メッセージがリモート・サイトに伝播されるとそれらを削除します。メッセージが使用される前に受信サイトが失敗した場合、これらのメッセージは失われます。ソース・サイトはメッセージを再送信できません。メッセージの重複配信も可能です。

関連項目:

- [DBA_QUEUE_SCHEDULES: すべての伝播スケジュール](#)
- [Oracle Real Application Clusters \(Oracle RAC\)を持つバッファ済メッセージ](#)

フロー制御

Oracle Database Advanced Queuingでは、アプリケーションによって共有メモリーがメッセージでいっぱいになるのを防ぐために、フロー制御システムを実装しています。1人の送信者当たりの未完了メッセージ数がシステム定義のしきい値を超えると、エンキュー・コールがブロックしてタイムアウトし、エラー・メッセージが発行されます。メッセージの送信者は、エンキュー・オプションの *sender_id.name* によって識別されます。キューのフロー制御によって特定の送信者がブロックされても、他のメッセージ送信

者には影響がありません。解決するにはメッセージをデキューします。これによってフロー制御が解決され、その後、新しくメッセージをエンキューできます。

フロー制御のしきい値はメモリー・プレッシャーによって異なり、ストリーム・プールの使用量が大きくなった場合は、システム定義の制限に依存する場合があります。メッセージ送信者は、イベントStreams AQ: フロー制御のためにエンキューがブロックされましたでブロックされ、フロー制御が解決されなければ、エラーORA-25307を発行してタイムアウトになります。アプリケーションによってこのエラーを処理し、失敗したメッセージを再エンキューする必要があります。

フロー制御を使用しても、マルチ・コンシューマ・キューのコンシューマが低速の場合、メモリー内に格納されるメッセージの数が無制限に増える可能性があります。対応する1つ以上のサブスクライバが存在する場合、メモリーを解放するために古いメッセージがディスクに移され、プールから削除されます。これにより、低速のコンシューマがディスク・アクセスのコストを負担し、高速のサブスクライバは妨害されずに作業を続行できます。

Oracle Real Application Clusters(Oracle RAC)を持つバッファ済メッセージ

シャード・キューおよび非シャード・キューでは、Oracle RAC上のバッファ済メッセージの処理が異なります。

シャード・キューは、インスタンス間通信を実行しますが、複数のOracle RACインスタンスでの同じブロックへの同時書き込みを回避します。通常、デキューはメッセージがエンキューされているインスタンスのローカルにあるシャードで行われますが、特定の状況ではOracleは別のインスタンスでデキューするためにメッセージをインスタンス間で効率的に転送します。たとえば、シャード・キューで、1つのOracleインスタンス上に単一のエンキュー・セッションがあり、別のインスタンスに単一のデキュー・セッションがある場合、シャード・キューはOracle RACインスタンス間でメッセージを転送します。パフォーマンスを向上させるため、メッセージの転送はエンキュー・トランザクションに対して非同期で行われます。シャードにメッセージがないインスタンスに接続している場合、デキュー元でORA-25228が発生する場合があります。

非シャード・キューの場合、アプリケーションは、パスワード・ベースの認証を使用してデータベースに接続するかぎり、Oracle RACインスタンスからバッファ済メッセージをエンキューおよびデキューできます。バッファ済メッセージに必要な構造は、1つのOracle RACインスタンスに実装されます。バッファ済メッセージ構造が実装されるインスタンスは、キューを格納するキュー表のOWNER_INSTANCEです。その他のインスタンスで受信されたエンキューおよびデキュー・リクエストは、インターコネクトによってOWNER_INSTANCEに転送されます。listener.oraのREMOTE_LISTENERパラメータも設定して、バッファ済メッセージ・リクエストを正しいインスタンスに転送できるようにする必要があります。内部的には、Oracle RAC上のバッファ済キューはインスタンス間でdblinksを使用することがあります。Oracle RACのバッファ済キューにエンキューまたはデキューする定義者権限のパッケージでは、パッケージのユーザーにINHERIT_REMOTE_PRIVILEGESを付与する必要があります。

サービス名はOracle RACの各キューに関連付けられ、DBA_QUEUESおよびUSER_QUEUESビューに表示されます。このサービス名は、常にバッファ済メッセージに最も効率的にアクセスできるインスタンスを指し、インスタンス間のping操作を最小限に抑えます。OCIクライアントは、サービス名を使用してバッファ済メッセージ操作を実行します。

キューからキューへの伝播にバッファ済メッセージを使用する場合は、インスタンス・アフィニティを指定することをお勧めします。これにより、メッセージを宛先Oracle RACシステムに伝播するときに、透過的なフェイルオーバーが行われます。プライマリAQ Oracle RACインスタンスに障害が発生した場合に、データベース・リンクを再度指定する必要がありません。

関連項目:

- OWNER_INSTANCEの詳細は、[「ALL_QUEUE_TABLES: 現行ユーザーがアクセス可能なキュー表」](#)を参照してください
- REMOTE_LISTENERパラメータの設定の詳細は、[『Oracle Databaseリファレンス』](#)のREMOTE_LISTENERに関する項を参照してください。
- [「DBA_QUEUES: データベース内の全キュー」](#)または[「USER_QUEUES: ユーザー・スキーマのキュー」](#)を参照してく

ださい

- [Oracle Real Application Clusters \(Oracle RAC\)のサポート](#)

バッファ済メッセージの制限

次のOracle Database Advanced Queuingの機能は、シャード・キューのバッファ済メッセージでのみサポートされています。

- メッセージの遅延
- 配列のエンキュー
- 配列のデキュー
- PL/SQL通知

現在、バッファ済メッセージでは、次のOracle Database Advanced Queuingの機能はサポートされていません。

- メッセージの保存
- グループ化トランザクション
- メッセージのエクスポートおよびインポート
- メッセージ・ゲートウェイ
- OCI通知

エラー処理

バッファ済メッセージでは、再試行回数および再試行遅延はサポートされていません。メッセージの期限切れはサポートされています。バッファ済メッセージが有効期限をすぎてもキューに残っている場合、永続メッセージとして例外キューに移されます。

非同期通知

非同期通知によって、クライアントは、関心があるメッセージの通知を受信できます。

このクライアントは、これらの通知を使用して複数のサブスクリプションを監視できます。クライアントは、自分自身のサブスクリプションに関する通知を受け取るためにデータベースと接続している必要はありません。バッファ済メッセージでは、非同期通知がサポートされています。メッセージの配信モードは、通知記述子のメッセージ記述子で使用できます。

クライアントは、各メッセージごとに実行されるコールバック関数を指定します。非同期通知を使用して実行可能ファイルを起動することはできませんが、コールバック関数でストアド・プロシージャを起動することはできます。

クライアントは、PL/SQL、Java Message Service(JMS)またはOCIコールバック関数を使用してプロシージャで通知を受信するか、電子メールまたはHTTPのPOSTを介して通知を受信できます。クライアントは、通知の表現をRAWまたはXMLのいずれかに指定することもできます。

注意:



シャード・キューは PL/SQL 通知のみをサポートします。

JMSキューでは、通知の一部としてデキューが行われるため、明示的デキューの必要はありません。RAWキューでは、クライアントはペイロード配信を指定できますが、REMOVE_NO_DATAモードでメッセージをデキューする必要があります。その他のすべての永

続キューでは、通知にはメッセージ・プロパティのみが含まれ、クライアントは明示的にデキューしてメッセージを受信します。

RAWキューのペイロード配信

RAWキューの場合、Oracle Database Advanced Queuingクライアントは、通知とともにメッセージ・ペイロードが配信されるように指定できます。

関連項目:

[AQ登録情報型](#)

信頼できる通知

クライアントは永続メッセージ通知を指定できます。Oracle RACインスタンスが失敗した場合は、別のOracle RACノードによって通知が配信されます。スタンドアロン・インスタンスが失敗した場合は、インスタンスの再起動時に通知が配信されます。

注意:



通知の信頼性では、サーバーの障害のみを参照します。Oracle Database Advanced Queuing がその他の理由でクライアント通知を配信できない場合、通知はクライアント登録とともにページされます。

指定されたポート通知

非シャード・キューの場合、Oracle Database Advanced Queuingクライアントは、OCIサブスクリプション・ハンドル属性OCI_ATTR_SUBSCR_PORTNOを使用して、通知が配信されるポートを指定できます。これは、ファイアウォールの背後にあるコンピュータ上のクライアントに対して特に有効です。リスナー・スレッドのポートは、最初の登録の前に環境ハンドル内の属性を使用して指定できます。スレッドは、初めてOCI Subscription Registerがコールされたときに開始されます。クライアントが別の環境ハンドルを使用して別のポートで別のスレッドを開始しようとする、Oracle Database Advanced Queuingはエラーを戻します。

注意:



指定のポート通知および IP アドレス通知は、OCI クライアントにのみ適用されます。

関連項目:

[『Oracle Call Interfaceプログラマーズ・ガイド』のOCIのパブリッシュ・サブスクライブ登録機能に関する項](#)

IPv6コンプライアンスおよび指定IPのサポート

非シャード・キューの場合、Oracle Database AQはIPv6をサポートし、Oracle Database AQクライアントはOCIサブスクリプション・ハンドル属性OCI_ATTR_SUBSCR_IPADDRを使用して、通知が配信されるIPアドレスを指定できます。これは、複数のネットワーク・インタフェース・カードまたはIPアドレスがあるコンピュータ上のクライアントに対して特に有効です。リスナー・スレッドのIPアドレスは、最初の登録の前に環境ハンドル内の属性を使用して指定できます。スレッドは、初めて

OCISubscriptionRegisterがコールされたときに開始されます。クライアントが別の環境ハンドルを使用して別のIPアドレスで別のスレッドを開始しようとすると、Oracle Database AQはエラーを戻します。IPアドレスが指定されていない場合、Oracle Database AQはクライアントがあるコンピュータのすべてのIPアドレスに対して通知を配信します。

登録タイムアウト

Oracle Database Advanced Queuingの以前のリリースでは、通知の登録は、クライアントによって明示的に削除されるか、拡張クライアントの障害によりページされるまで保持されていました。Oracle Database Advanced Queuing 10gリリース2 (10.2)以降では、クライアントは一定期間登録した後、登録は自動的にページされます。

登録がページされると、Oracle Database Advanced Queuingはクライアントに通知を送信するため、クライアントはコールバックを起動して、必要なアクションを実行できます。

関連項目:

timeoutパラメータの詳細は、[「AQ登録情報型」](#)を参照してください。

通知のページ

クライアントは、最初の通知のみ受信するように登録することもできます。その後、登録は自動的にページされます。

通知のページが役立つ例としては、クライアントがエンキューの開始を待機している場合があります。この場合は、最初の通知のみが役に立ち、その後の通知では追加情報は提供されません。以前なら、このクライアントは、エンキューの開始後、登録を解除する必要がありましたが、現在は、登録が自動的に解除されるよう構成できます。

バッファ済メッセージの通知

クライアントは、バッファ済メッセージの通知を登録できます。登録リクエストは、バッファ済メッセージと永続メッセージの両方に適用されます。PL/SQLまたはOCI通知とともに配信されるメッセージ・プロパティは、メッセージがバッファ済か永続かどうかを指定します。

関連項目:

- PL/SQL通知の詳細は、[「通知の登録」](#)を参照してください。
- OCI通知の例については、付録C「OCIの例」を参照してください。ただし、この付録は、このマニュアルのHTMLバージョンにのみ含まれています。

信頼できる通知はサポートされていません。

登録のビュー

ディクショナリ・ビューDBA_SUBSCR_REGISTRATIONSおよびUSER_SUBSCR_REGISTRATIONSは、システム内の各種登録を示します。

診断ビューGV\$SUBSCR_REGISTRATION_STATSは、通知統計およびパフォーマンスの監視に使用されます。

イベント・ベースの通知

イベント・ベースの通知は、コーディネータ(EMNC)および下位プロセスのセットで処理されます。

イベント通知ロードはこのプロセス間に配置されます。このプロセスはシステム通知で平行に機能して、より大量な通知をより迅速なレスポンス時間で処理し、ステージング通知で使用する共有メモリーを削減する機能を提供します。

通知の時間別のグループ化

通知アプリケーションは、指定した時間間隔に発生した全イベントに関する1つの通知を受信するよう登録できます。通知クライアントは、通知の開始時間を指定できます。さらに、グループ化クラスとして時間を指定し、グループ化の値として時間間隔を指定する必要があります。

配信される通知の数を制限するために繰返し件数を使用できます。クライアントは2つのタイプのグループ化イベント(サマリーまたは最終)を受け取ることができます。「サマリー」通知は、サブスクリプションの全メッセージのメッセージ識別子のリストです。「最終」がグループ化タイプとして指定された場合、通知には、通知間隔の最後のメッセージの情報が含まれます。間隔内のメッセージ件数も送信されます。PLSQLおよびOCIの登録インタフェースでは、START_TIME、REPEAT_COUNT、GROUPING CLASS、GROUPING VALUE、GROUPING TYPEをAQ\$_REGISTRATION_INFOおよびOCIサブスクリプション・ハンドルで指定できます。

AQ通知を開始したクライアントによって受信された通知記述子は、メッセージ識別子のグループおよびグループ内の通知数に関する情報を提供します。

関連項目:

- [『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)
- [『Oracle Call Interfaceプログラマーズ・ガイド』](#)

エンキュー機能

このトピックでは、メッセージの配列のエンキュー、相関識別子、エンキューでのメッセージの優先順位と順序付け、メッセージのグループ化、送信元の識別、および時間指定とスケジュールなどのエンキュー機能について説明します。

次の機能は、メッセージのエンキューに適用されます。

- [メッセージの配列のエンキュー](#)
- [相関識別子](#)
- [エンキューにおけるメッセージの優先順位および順序付け](#)
- [メッセージのグループ化](#)
- [送信元の識別](#)
- [時間指定とスケジュールリング](#)

メッセージの配列のエンキュー

複数のメッセージをキューにエンキューすると、1つのメッセージを1回ずつ操作するのではなく、1つの配列の複数メッセージを同時に操作できます。これにより、エンキュー操作のパフォーマンスが向上します。メッセージの配列をキューにエンキューする場合、エンキュー・オプションは各メッセージで共有されますが、メッセージ・プロパティはメッセージごとに個別に指定できます。配列のエンキュー操作は、PL/SQLまたはOCIを使用して実行できます。

このリリースのバッファ済メッセージでは、配列のエンキューはサポートされていません。

相関識別子

各メッセージに識別子を割り当て、後で特定のメッセージを取得する方法を提供できます。

エンキューにおけるメッセージの優先順位および順序付け

エンキューされたメッセージの優先順位およびキュー内での正確な位置を指定できます。つまり、ユーザーはメッセージが使用される順序を次の3つの方法で指定できます。

- 各メッセージに優先順位を割り当てます。
- ソート順序では、どのプロパティを使用してキューのすべてのメッセージを順序付けるかを指定します。これは、キュー表の作成時に設定され、変更できません。優先順位、エンキュー時刻またはコミット時刻でメッセージをソートすることもできます。

関連項目:

- ソート順序の詳細は、[「キュー表の作成」](#)を参照してください。
- 順序逸脱では、メッセージの位置を他のメッセージとの関連で指定します。

注意:



10g リリース 2(10.2)では、順序逸脱機能は非推奨になりました。

複数のコンシューマが同じキューを操作している場合、各コンシューマはすぐに使用できる最初のメッセージを取得します。別のコンシューマが使用中のメッセージはスキップされます。

メッセージの優先順位による順序付けは、優先順位およびエンキュー時刻でソート順序を指定することによって実行されます。優先順位による順序付けを選択すると、各メッセージがエンキューされるときにエンキュー・エージェントによって優先順位が割り当てられます。デキューのときには、割り当てられた優先順位でデキューされます。2つのメッセージの優先度が同じ場合、デキューの順番はエンキュー時間で決定されます。先入れ先出し(FIFO)優先順位のキューも、エンキュー時刻および優先順位でメッセージのソート順序を指定することによって作成できます。

メッセージのグループ化

1つのキューに属するメッセージをグループ化すると、1人のユーザーが一度に消費することのみが可能なセットを形成できます。そのためには、メッセージのグループ化が可能なキュー表にキューを作成する必要があります。同じグループに属するメッセージはすべて同じトランザクション内に作成する必要があり、同じトランザクション内に作成されたメッセージはすべて同じグループに属します。

この機能によって、複雑なメッセージを、複数の単純なメッセージにセグメント化できます。たとえば、あるキュー宛てのメッセージに請求書情報が含まれている場合、そのメッセージは、最初にヘッダーのメッセージ、次に詳細情報のメッセージ、その次にフッターのメッセージというような順番で構成されるグループとして作成できます。

小さいオブジェクトにセグメント化できるイメージやビデオなどの複合ラージ・オブジェクトがメッセージ・ペイロードにある場合は、メッセージのグループ化が有効です。

グループ・メッセージ・プロパティ(優先順位、遅延、期限切れ)は、単にグループの最初のメッセージのプロパティによってのみ判断され、グループの他のメッセージのプロパティは無視されます。

グループ化メッセージのプロパティは、伝播されても保持されます。ただし、メッセージの伝播先キューも、トランザクション処理で

ループ化可能な状態にする必要があります。トランザクション処理でグループ化可能なキューからメッセージをデキューするときにメッセージ・グループのプロパティを保持する場合、他にも注意する必要がある制限があります。

送信元の識別

アプリケーションは、送信メッセージにユーザー定義の識別マークを付加できます。Oracle Database Advanced Queuingでは、メッセージがデキューされたキューを自動的に識別することもできます。これにより、アプリケーションで、伝播されたメッセージや同じデータベース内の文字列メッセージの追跡が可能になります。

時間の指定とスケジュール

メッセージをエンキューするときに、そのメッセージがいつまで使用可能かという期限切れを指定できます。デフォルトでは、期限切れはありません。期限切れになったメッセージは、例外キューに移されます。期限切れ処理を行うには、キュー・モニターを起動しておく必要があります。

デキュー機能

このトピックでは、同時デキュー、デキューの方法、デキュー・モード、メッセージの配列のデキュー、メッセージの状態、デキューでのメッセージのナビゲーション、メッセージの待機、遅延を伴う再試行、オプションのトランザクション保護、例外キューなどのデキュー機能について説明します。

次の機能は、メッセージのデキューに適用されます。

- [同時デキュー](#)
- [デキューの方法](#)
- [デキュー・モード](#)
- [メッセージの配列のデキュー](#)
- [メッセージの状態](#)
- [デキューにおけるメッセージのナビゲーション](#)
- [メッセージの待機](#)
- [遅延を伴う再試行](#)
- [トランザクション保護のオプション](#)
- [例外キュー](#)

同時デキュー

シングル・コンシューマ・キューからデキューするプロセスまたはマルチ・コンシューマ・キューから同じコンシューマでデキューするプロセスが複数ある場合、同時プロセスで処理中のメッセージは別のプロセスではスキップされます。このため、複数のプロセスが同一コンシューマの異なるメッセージを同時に処理できます。

デキューの方法

次のいずれかの方法で、メッセージをデキューできます。

- 相関識別子の指定
相関識別子は、ユーザー定義のメッセージ・プロパティです。同じ相関識別子を持つ複数のメッセージがキュー内に存在することが可能です。これにより、メッセージ間の順序(エンキュー順序)がデキュー・コールで保持されない場合があります。
- メッセージ識別子の指定

メッセージ識別子はシステムによって割り当てられる値(RAWデータ型)です。同じメッセージ識別子を持つメッセージがキュー内に複数存在することはありません。

- デキュー条件の指定

デキュー条件は、メッセージ・プロパティまたはメッセージ内容によって表現され、SQL問合せのWHERE句と同様の構文を持ちます。キュー内のメッセージは条件に対して評価され、指定された条件を満たすメッセージが返されます。デキュー条件を使用すると、メッセージのデキュー順序が不定になり、キューのソート順序は考慮されません。

- デフォルトのデキュー

デフォルトのデキューは、最初の使用可能なメッセージを取り出します。

注意:



相関識別子、メッセージ識別子またはデキュー条件を使用してメッセージをデキューすると、グループ化メッセージのプロパティが変更されます。

デキュー・モード

デキュー・リクエストは、メッセージをブラウズして削除するか、データを伴わないメッセージを削除できます。メッセージが参照されると、そのメッセージは引き続き処理できます。メッセージは、削除されるか、データを伴わずに削除されると、デキュー・リクエストには使用できなくなります。キュー・プロパティによっては、取り消されたメッセージがキュー表に保持されることもあります。キューに保存時間が指定されている場合、メッセージは削除された後もキュー表に保存されます。

ブラウズ・モードには3点のリスクがあります。1点目は、ブラウズ後にメッセージが再度デキューできる保証がないという点です。同時ユーザーからのデキュー・コールによってメッセージが削除される可能性があるためです。表示されたメッセージが同時ユーザーによってデキューされないようにするには、メッセージをロック・モードで表示する必要があります。

2点目は、待機時間を0(ゼロ)以外に指定しており、ナビゲート位置がキューの最後に到達した場合、ブラウズ・モードのデキュー位置は自動的にそのキューの先頭に変更されるという点です。NEXT_MESSAGEナビゲーション・オプションと0(ゼロ)以外の待機時間を指定してブラウズ・モードでデキュー・コールを繰り返すと、同じメッセージを何度もデキューすることになります。セッション内のキューの最初のデキュー・コールについては待機時間を0(ゼロ)以外に指定し、後続のデキュー・コールでは待機時間を0(ゼロ)にしてNEXT_MESSAGEナビゲーション・オプションを使用することをお勧めします。デキュー・コールが、キューの最後であるというエラー・メッセージを受信した場合、デキュー・コールによってデキュー位置を明示的にキューの最初に設定できます。これには、FIRST_MESSAGEナビゲーション・オプションを使用します。その後、キュー内のメッセージをまたブラウズできます。

さらに、キューのソート順序がENQ_TIME、PRIORITYまたはこの2つの組合せの場合、ブラウズからブラウズへ結果を繰返しできない場合があります。一貫したブラウズ結果が必要な場合は、[コミット時間キュー](#)を使用する必要があります。

関連項目:

- [キュー表の作成](#)

REMOVE_NODATAモードでメッセージがデキューされた場合、メッセージのペイロードは取り出されません。このモードは、ユーザーが先にBROWSEモードでデキューしてペイロードを調べてあるときに有効です。

メッセージの配列のデキュー

複数のメッセージをキューからデキューすると、1つのメッセージを1回ずつ操作するのではなく、1つの配列の複数メッセージを同

時に操作できます。これにより、デキュー操作のパフォーマンスが向上します。トランザクション・キューからデキューする場合は、トランザクションに対するすべてのメッセージを1回のコールでデキューできるため、アプリケーションのプログラミングが容易になります。キューからメッセージの配列をデキューする場合、デキュー・オプションは各メッセージで共有されますが、メッセージ・プロパティはメッセージごとに個別に指定できます。配列のエンキュー操作とデキュー操作は、PL/SQLまたはOCIを使用して実行します。このリリースのバッファ済メッセージでは、配列のデキューはサポートされていません。

メッセージの状態

複数のプロセスまたはスレッドが、1つのキューから同じコンシューマ名を使用して同時にデキューできます。そのような場合、Oracle Database Advanced Queuingはキューの先頭にあるロックされていない最初のメッセージをコンシューマに提供します。デキュー中に特定のメッセージのメッセージ識別子が指定されないかぎり、コンシューマはREADY状態のメッセージをデキューできます。

メッセージがPROCESSED(処理済)とみなされるのは、所定のコンシューマ全員がメッセージのデキューに成功したときのみです。メッセージがEXPIRED(期限切れ)とみなされるのは、1つまたは複数のコンシューマがEXPIRATION時刻までにそのメッセージをデキューしなかったときです。期限切れになったメッセージは、例外キューに移されます。

マルチ・コンシューマ・キューから移された期限切れメッセージは、受信者を指定してデキューできません。ただし、デキュー・オプションのコンシューマ名にNULLを指定することによって、REMOVEモードで1回のみデキューできます。

注意:

マルチ・コンシューマ例外キューが、compatible パラメータを 8.0 に設定してキュー表に作成された場合は、期限切れメッセージはメッセージ識別子を指定する方法でのみデキューできます。



compatible が 8.0 に設定されているキュー表で作成されたキュー(このマニュアルでは 8.0 形式のキューと呼びます)は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になっています。したがって、新しいキューの作成には 8.1 以降の形式を使用し、既存の 8.0 形式のキューをなるべく早く移行することをお勧めします。

Oracle Database Advanced Queuingリリース8.1.6以上では、マルチ・コンシューマ・キューからメッセージを削除できるのはキュー・モニターのみです。これにより、デキュー元は、キュー表のメッセージをロックせずにデキュー操作を完了できます。キュー・モニターは、すべてのコンシューマが処理を完了したメッセージをマルチ・コンシューマ・キューから削除する作業を毎分約1回行うため、メッセージが完全に処理された後キューから物理的に削除されるまでの遅延があります。

デキューにおけるメッセージのナビゲーション

キューからメッセージを選択するには、いくつかのオプションがあります。FIRST_MESSAGEナビゲーション・オプションを使用して、最初のメッセージを選択できます。メッセージを選択してキュー内での位置を設定した後、NEXT_MESSAGEナビゲーション・オプションを使用して次のメッセージを選択することもできます。

FIRST_MESSAGEナビゲーション・オプションでは、キューに対してSELECTが実行されます。NEXT_MESSAGEオプションでは、FIRST_MESSAGEナビゲーションで実行されたSELECTの結果からフェッチが実行されます。このように、後続のデキューではSELECT全体を再度実行する必要がないため、パフォーマンスが最適化されます。

トランザクション処理のグループ化がキューで使用可能になっている場合、ナビゲーション・オプションの動作はわずかに異なります。FIRST_MESSAGEが要求された場合、デキュー位置はやはりキューの最初にリセットされます。しかし、NEXT_MESSAGEが要求されると、位置は同じトランザクション内の次のメッセージに設定されます。トランザクション処理のグループ化では、

NEXT_TRANSACTIONオプションも使用できます。これにより、デキュー位置が次のトランザクションの最初のメッセージに設定されず。

相関識別子またはメッセージ識別子を指定してデキューするか、トランザクションのメッセージの一部をデキューしてコミットする場合は、グループ化トランザクションは無効です。

NEXT_MESSAGEまたはNEXT_TRANSACTIONオプションの使用中にキューの最後に到達し、待機時間を0(ゼロ)以外に指定していた場合、ナビゲート位置は自動的にそのキューの先頭に変更されます。待機時間を0(ゼロ)に指定した場合、キューの最後に到達すると例外が発生することがあります。

メッセージの待機

Oracle Database Advanced Queuingでは、新しくエンキューされるメッセージまたはREADY状態になるメッセージのどちらかに、1つ以上のキューでアプリケーションを待機させることができます。DEQUEUE操作によって、1つのキューへのメッセージ到着を待機でき、LISTEN操作によって複数のキューへのメッセージ到着を待機できます。

注意:



アプリケーションは例外キューのブロック・デキューを実行して、EXPIRED メッセージを待機することもできます。

ブロック(待機)しているDEQUEUEコールが戻るとき、メッセージ・プロパティおよびメッセージ・ペイロードが戻されます。ブロックしているLISTENコールが戻るときは、メッセージが届いたキューの名前のみが戻ります。メッセージをデキューするためには、その後にDEQUEUE操作をする必要があります。

エージェント・リストの複数のエージェントに向けたメッセージがいくつかある場合、LISTENはメッセージの宛先になっている最初のエージェントとともに戻ります。あるエージェントが他のエージェントに対してメッセージの欠乏状態を引き起こさないように、アプリケーションはエージェント・リストのエージェント順序を変更できます。

アプリケーションは、必要に応じてOracle Database Advanced Queuingのメッセージ到着待機時間を示すタイムアウトを0(ゼロ)または任意の秒数に指定できます。デフォルトでは、そのキューにメッセージが到着するまで、待機することになっています。これにより、アプリケーションからメッセージをポーリングし続けるという負担がなくなり、新しいメッセージがエンキューされるか、DELAY時間がすぎてREADY状態になるまでブロックし続けるため、CPUおよびネットワーク・リソースの節約になります。

デキューによってブロックされたアプリケーションは、新しいメッセージにDELAYが指定されていない場合はエンキュー元が直接アクティブにし、DELAYまたはEXPIRATION時間が経過した場合はキュー・モニター・プロセスがアクティブにします。アプリケーションがリモート・キューでメッセージの到着を待機している場合、Oracle Database Advanced Queuingのプロパゲータにより、メッセージの伝播後に、ブロックされたデキュー元がアクティブになります。

遅延を伴う再試行

キューからメッセージをデキューするトランザクションが失敗した場合、そのメッセージを使用する試行に失敗したとみなされます。Oracle Database Advanced Queuingは、メッセージ使用の試行に失敗した回数をメッセージ履歴に記録します。アプリケーションは、キュー表ビューのRETRY_COUNT列を問い合せて、メッセージに対する試行の失敗回数を参照できます。さらに、Oracle Database Advanced Queuingでは、アプリケーションがキュー内のメッセージに対する再試行の最大回数を、キュー・レベルで指定できます。再試行の最大回数のデフォルト値は5です。メッセージ削除がこの数より多く失敗した場合、メッセージは例外キューに移動されるか、またはアプリケーションで使用できなくなります。

注意:



サーバー・プロセスがインスタンスで停止した(ALTER SYSTEM KILL SESSION または SHUTDOWN ABORT など)ためにデキュー・トランザクションが失敗した場合、RETRY_COUNT は増分されません。

条件が不適切な場合、メッセージを受信するトランザクションが終了する場合があります。Oracle Database Advanced Queuingでは、WAITING状態のときに、指定された再試行の遅延間隔で、不適切なメッセージを隠すことができます。再試行の遅延後、失敗したメッセージが再度デキュー可能になります。Oracle Database Advanced Queuingのタイム・マネージャは、再試行遅延プロパティを強制的に適用します。再試行の遅延のデフォルト値は0(ゼロ)です。

複数のセッションが1つのキューから同時にメッセージをデキューしている場合、RETRY_COUNT情報が常に正しく更新されるとはかぎりません。セッション1がメッセージをデキューしてトランザクションをロールバックすると、Oracle Database AQでは、このメッセージのRETRY_COUNT情報を更新する必要があることが認識されます。ただし、セッション1がロールバックを完了するまではRETRY_COUNTを増分できません。セッション1がロールバックを完了してからRETRY_COUNTを増分するまでの間に、セッション2が同じメッセージをデキューしようとする、セッション2によるデキューは成功します。セッション1がRETRY_COUNTを増分しようとする、メッセージがセッション2によりロックされていることが検出され、RETRY_COUNTは増分されません。インスタンスのUSER_DUMP_DESTINATIONに、次のメッセージを含むトレース・ファイルが生成されます。

```
Error on rollback: ORA-25263: no message in queue schema.qname with message ID ...
```

注意:

再試行の最大回数および再試行の遅延は、8.0 形式のマルチ・コンシューマ・キューでは使用できません。



compatible が 8.0 に設定されているキュー表で作成されたキュー(このマニュアルでは 8.0 形式のキューと呼びます)は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になっています。したがって、新しいキューの作成には 8.1 以降の形式を使用し、既存の 8.0 形式のキューをなるべく早く移行することをお勧めします。

トランザクション保護のオプション

エンキュー・リクエストおよびデキュー・リクエストでは、通常は複数のリクエストを含むトランザクションの一部として、必要な[トランザクション](#)処理が提供されます。ただし、特定のリクエストをそれ自身でトランザクションに指定して、そのリクエストの結果をすぐに他のトランザクションから参照できるようにできます。つまり、エンキュー文またはデキュー文が適用されたとき、またはそのトランザクションがコミットされた後、メッセージを外部から参照できるようにできます。

注意:



バッファ済メッセージでは、トランザクション保護はサポートされていません。

例外キュー

例外キューは、期限切れまたは処理できないメッセージのリポジトリになります。アプリケーションがメッセージを例外キューに直接エンキューすることはできません。また、マルチ・コンシューマの例外キューに、サブスクライバを対応付けることはできません。ただし、期限切れまたは処理できないメッセージを処理するアプリケーションは、REMOVE(削除)モードを使用する例外キューから1回のみデキューできます。デキュー中のコンシューマ名はNULLに指定する必要があります。メッセージ識別子を指定すると、メッ

ページも例外キューからデキューできます。

注意:

期限切れまたは処理できないバッファ済メッセージは、永続メッセージとして例外キューに移されます。

シングル・コンシューマ・キューまたは 8.0 形式のマルチ・コンシューマ・キューを指定したメッセージが例外キューに移された場合は、メッセージ識別子を指定してデキューできます。

`compatible` が 8.0 に設定されているキュー表で作成されたキュー(このマニュアルでは 8.0 形式のキューと呼びます)は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になっています。したがって、新しいキューの作成には 8.1 以降の形式を使用し、既存の 8.0 形式のキューをなるべく早く移行することをお勧めします。

メッセージが例外キューに移動した後、例外キューに移動する前にメッセージが常駐していたキューを識別する方法はありません。この情報が重要な場合、アプリケーションはこの情報をメッセージ自体に保存する必要があります。

例外キューは、エンキュー時に指定可能なメッセージ・プロパティです。例外キューが指定されていないと、デフォルトの例外キューが使用されます。デフォルトの例外キューは、キュー表が作成されるときに自動的に作成されます。

メッセージは、次の条件が成立するときに例外キューに移されます。

- 指定された期限内にデキューされなかった場合。

複数の受信者を指定したメッセージの場合、指定されていない指定された期限切れまでにそのメッセージをデキューできない受信者が1つでもあると、メッセージは例外キューに移されます。デフォルトの期限切れはなしで、そのメッセージは期限切れになりません。

- メッセージは正常にデキューされたものの、その処理でエラーが発生したために、メッセージをデキューしたアプリケーションがトランザクションをロールバックした場合。デキューされても、ロールバック回数が再試行制限の指定回数を超過したメッセージは例外キューに移されます。

複数の受信者を指定したメッセージの場合、受信者ごとにそれぞれの再試行回数が保持されます。すべての受信者の再試行回数が再試行制限の指定を超えたときのみ、そのメッセージは例外キューに移されます。

デフォルトの再試行回数は、シングル・コンシューマ・キューおよび8.1形式以上のマルチ・コンシューマ・キューの場合は5です。8.0形式のマルチ・コンシューマ・キュー(Oracle Database Advanced Queuing 10gリリース2 (10.2)では非推奨)では再試行制限はサポートされていません。

注意:

サーバー・プロセスがインスタンスで停止した(`ALTER SYSTEM KILL SESSION` または `SHUTDOWN ABORT` など)のためにデキュー・トランザクションが失敗した場合、`RETRY_COUNT` は増分されません。

- クライアントによって処理された文に含まれているデキューは成功したが、文自体は後で例外処理のために取り消された場合。

デキュー・プロセスに成功してPL/SQLプロセスに例外が発生した場合、Oracle Database Advanced

Queuingは、デキュー・プロセスにより戻されたメッセージの再試行回数を増分します。

- クライアント・プログラムはメッセージのデキューに成功したが、トランザクションをコミットする前に終了した場合。

伝播機能

メッセージは1つのキューから別のキューに伝播でき、これにより、アプリケーションは同じデータベースまたは同じキューに接続されていなくても相互に通信できます。宛先キューは同じデータベースでもリモート・データベースでも構いません。

伝播により、すべてのサブスクライバに1つのキューからメッセージをデキューするよう要求しなくても、多くの受信者にメッセージを展開できます。伝播を使用すると、異なるキューのメッセージを1つのキューに結合することもできます。これは、メッセージのコンポジットまたはファネリングと呼ばれます。

注意:

- マルチ・コンシューマ・キューからシングル・コンシューマ・キューにメッセージを伝播できます。シングル・コンシューマ・キューからマルチ・コンシューマ・キューへの伝播は不可能です。
- 非シャード・キューの場合は、マルチ・コンシューマ・キューからシングル・コンシューマ非シャード・キューにメッセージを伝播できます。シングル・コンシューマ・キューからマルチ・コンシューマ非シャード・キューへの伝播は不可能です。
- シャード・キューの場合は、シングル・コンシューマおよびマルチ・コンシューマのシャード・キュー間で伝播できます。
- シャード・キューおよび非シャード・キュー間では伝播できません。

伝播されたメッセージがリモート・キューからデキューされていなくても、ソース・キューでは、伝播された直後に処理済とマークされます。同様に、伝播されたメッセージがリモート・キューで期限切れになると、そのメッセージはローカル・キューの例外キューではなく、リモート・キューの例外キューに移されます。現状では、Oracle Database Advanced Queuingは、例外をソース・キューに伝播しません。

伝播を有効化するために、メッセージ伝播の送信元のキューに1つ以上のサブスクライバが定義され、そのキューからメッセージを伝播する宛先ごとに、スケジュールが定義されます。

Oracle Database Advanced Queuingによって、リモート・キューの型が、それが作成されたキャラクタ・セットのコンテキストで、ローカル・キューの型と構造的に同等であるかが自動的に確認されます。ソース・キューでエンキューされたメッセージが伝播され、自動的に宛先キューでデキューできるようになります。

メッセージが宛先キューに届くと、ソース・キューのスキーマ名に基づいたセッションによって、新しく届いたメッセージが宛先キューにエンキューされます。つまり、ソース・キューのスキーマに、宛先キューに対するエンキュー権限を付与する必要があります。

伝播は、Oracle Schedulerジョブとして実行されます。バックグラウンド・プロセス、JOB_QUEUE_PROCESSがジョブを実行します。伝播のスケジューリングは、継続的に終了せずに実行される専用のプロセス、または伝播するメッセージがある場合のみ実行されるイベント・ドリブンのプロセスです。

Oracle Database Advanced Queuingには、2種類の伝播があります。

- キューからdblinkへの伝播

- キューからキューへの伝播

キューからdblinkへの伝播では、ソース・キューから、dblinkで識別される宛先データベースのすべてのサブスクライブ・キューにメッセージまたはイベントが配信されます。

1つの伝播スケジュールを使用して、すべてのサブスクライブ・キューにメッセージが伝播されます。したがって、このスケジュールを変更すると、すべてのサブスクライブ・キューへのメッセージ配信に影響します。

キューからキューへの伝播では、ソース・キューから、dblinkで識別される特定の宛先キューにメッセージまたはイベントが配信されます。これにより、メッセージ配信の伝播スケジュール制御の自由度を高めることができます。

この新しい伝播モードは、宛先Oracle RACシステムに伝播する場合、透過的なフェイルオーバーもサポートします。キューからキューへの伝播では、Oracle RACでキューの所有者インスタンスが失敗した場合、データベース・リンクを再指定する必要はなくなります。

Oracle Database Advanced Queuingには、伝播されたメッセージおよびそのスケジュールに関する詳細な統計を取得する機能があります。この情報は、最高のパフォーマンスが得られるようにスケジュールを調整するために使用できます。

リモート・コンシューマ

マルチ・コンシューマ・キュー内のメッセージのコンシューマは、ローカルまたはリモートです。ローカル・コンシューマは、プロデューサーがそのメッセージをエンキューしたキューからデキューします。ローカル・コンシューマのエージェント説明には名前がありますが、アドレスまたはプロトコルはありません。

リモート・コンシューマは、メッセージがエンキューされたキューとは異なるキューからデキューします。リモート・コンシューマは、3つのカテゴリに分類されます。

- アドレスが同一データベース内のキューを参照しているもの。

この場合、コンシューマは同一データベースにある別のキューからメッセージをデキューします。アドレスの形式は [schema].queue_name です。スキーマが指定されていない場合は、現在のユーザーのスキーマが使用されます。

- アドレスが別のデータベース内のキューを参照しているもの。

この場合の別のデータベースは、データベース・リンクを使用して到達でき、プロトコルがNULLまたは0(ゼロ)である必要があります。アドレスの形式は、 [schema].queue_name@dblink になります。スキーマが指定されていない場合は、現在のユーザーのスキーマが使用されます。データベース・リンクにドメイン名が指定されていない場合は、DB_DOMAIN init.oraパラメータで指定されたデフォルトのドメインが使用されます。

- アドレスがサード・パーティのプロトコルによって到達できる宛先を参照しているもの。

サード・パーティ・ソフトウェアのドキュメントを参照してデータベース・リンクのアドレスおよびプロトコルの指定方法、および伝播のスケジュールリング方法を決定する必要があります。

リモート・サブスクライバへの伝播

Oracle Database Advanced Queuingでは、スケジュールの実行時に伝播スケジュールで指定されたデータベース・リンクを検証しますが、スケジュールの作成時には検証しません。そのため、関連付けられたデータベース・リンクの作成前に、キューからdblinkや、キューからキューへの伝播を作成することが可能です。また、データベース・リンクを削除しても、伝播スケジュールは無効になりません。

Oracle Database AQには、2種類の伝播があります。

A) キューからdblinkへの伝播 - (ソース)キューおよび(宛先)データベース・リンクによって指定されます。dblinkで指定された宛先のすべてのキューに対するソース・キューのメッセージは、この伝播で処理されます。

この使用例では、dblinkが同一のデータベースに接続しているため、1つのソース・キューでは複数の伝播を持つことはできません。

ん。したがって、(q1, dblink1)と(q1, dblink2)は、両方のdblinkが同一のデータベースに接続している場合、同時に存在できません。ただし、(q1, dblink1)と(q2, dblink1)、または(q1, dblink1)と(q2, dblink2)は、ソース・キューが異なるため共存できます。

B) キューからキューへの伝播 - (ソース)キュー、(宛先)dblinkおよび(宛先)キューによって指定されます。宛先dblinkで示されるキューに対するソース・キューのメッセージは、この伝播で処理されます。ここでは、(q1, dblink1, dq1)と(q1, dblink1, dq2)または(q1, dblink1, dq1)と(q1, dblink2, dq2)は、成功します。これは、ソース・キューが同一でdblinkが同一のデータベースに接続されている場合でも、宛先キューが異なるためです。

この使用例では、dblink1およびdblink2が同一のデータベースを指している場合、(q1, dblink1, q2)と(q1, dblink2, q2)のように異なるdblinkを使用しても、1つのソース・キューと宛先キューの間では複数の伝播を持つことはできません。

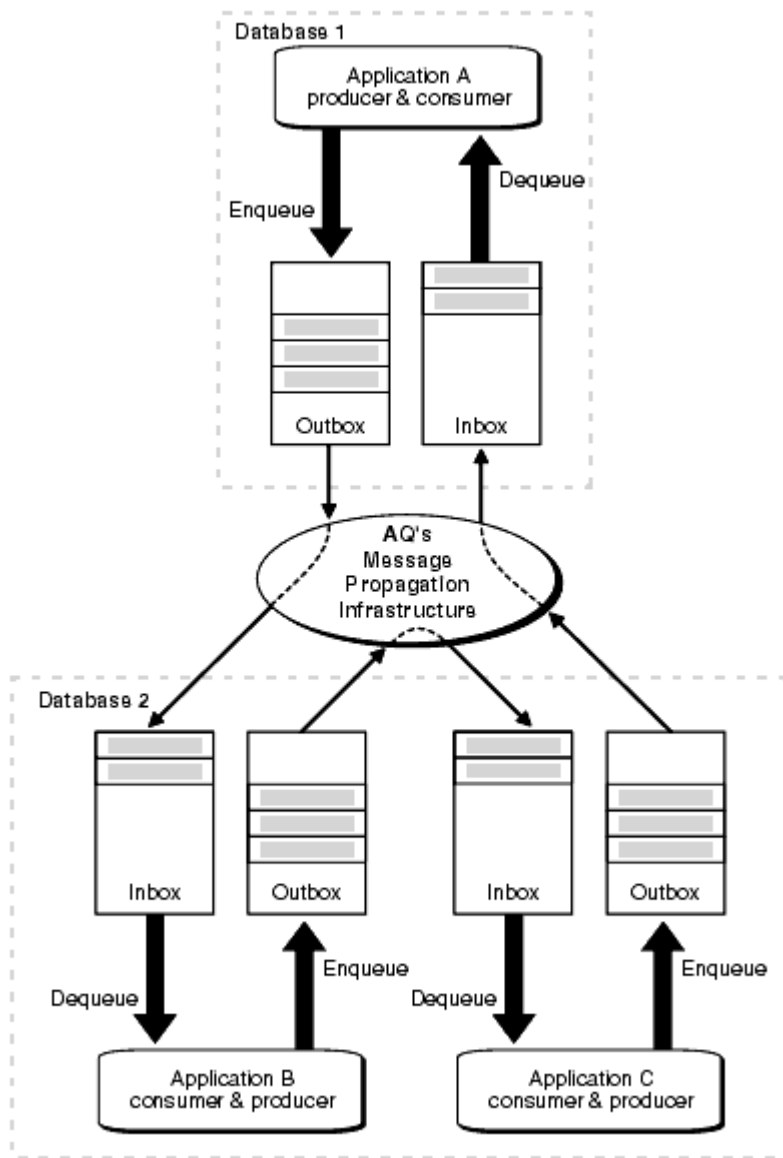
伝播におけるメッセージの優先順位および順序付け

遅延、期限切れおよび優先順位パラメータは、キューからdblinkへの伝播とキューからキューへの伝播の両方で、ローカル・コンシューマにもリモート・コンシューマにも同様に適用されます。Oracle Database Advanced Queuingは伝播による遅延を見込んで、遅延および期限切れパラメータを調整します。たとえば、期限切れが1時間で、メッセージが15分後に伝播された場合、リモート・キューでの期限切れは45分に設定されます。

受信ボックスと送信ボックス

[図1-9](#)は、Oracle Database Advanced Queuingを使用して通信する異なるデータベース上のアプリケーションを示しています。各アプリケーションには、受信メッセージを処理する受信ボックスと、送信メッセージを処理する送信ボックスがあります。アプリケーションがメッセージをエンキューするたびに、メッセージはその宛先にかかわらず送信ボックスに入ります。同様に、アプリケーションは、メッセージの送信元に関係なく、受信ボックスにあるメッセージをデキューします。

図1-9 Oracle Database Advanced Queuingでのメッセージの伝播



伝播スケジュール

キューからdblinkへの伝播スケジュールは、1組のソース(伝播元のキュー)および宛先データベース・リンクに対して定義されます。キューからキューへの伝播スケジュールは、1組のソース(伝播元のキュー)および宛先キューに対して定義されます。あるキューにいくつかのキューに伝播されることになっている複数のメッセージがある場合、宛先キューごとにスケジュールを定義する必要があります。キューからdblinkへの伝播では、特定のリモート・データベースのすべてのスケジュールが同じ頻度で行われます。キューからキューへの伝播では、各スケジュールの頻度は、その他のスケジュールに関係なく調整されます。

スケジュールは時間の枠を示し、メッセージはその枠内にソース・キューから伝播されます。この時間枠は、ネットワーク通信量、ソース・データベースの負荷、接続先データベースの負荷などの複数の要因に左右されます。存続時間が指定されない場合、時間枠は無制限の単一枠になります。枠を定期的に繰り返す必要がある場合、連続する枠の間の周期的間隔を定義するNEXT_TIME機能を使用して有限の存続時間を指定します。

スケジュールが作成されると、ジョブは自動的にジョブ・キューの機能に発行され、伝播が処理されます。

あるキューに定義された伝播スケジュールは、そのキューの有効期間中いつでも変更または削除できます。スケジュールを削除するかわりに一時的に使用不可にすることもできます。すべての管理コールは、スケジュールがアクティブかどうかに関係なく実行されます。スケジュールがアクティブの場合、コールが処理されるまでに数秒かかります。

LOBを伴うメッセージの伝播

Oracle Database Advanced Queuingを使用したラージ・オブジェクトの伝播には、2通りの方法があります。

- RAWキューからの伝播

RAWキューでは、メッセージ・ペイロードはBLOBとして保存されます。これによって、PL/SQLインタフェースを使用したときには32KBまでのデータを格納でき、OCIを使用したときには、クライアントが同じ容量のデータを連続して割り当てることができます。この方法は、リリース8.0.4以上でサポートされています。

- LOB属性を伴うオブジェクト・キューからの伝播

ユーザーは、Oracle DatabaseのLOB処理ルーチンを使用して、LOBを移入することもLOBから読み込むこともできます。LOB属性は、BLOBまたはCLOB(NCLOBではなく)です。属性がCLOBの場合、Oracle Database Advanced Queuingはソース・キューと宛先キュー間で必要なすべてのキャラクタ・セットの変換を自動的に実行します。この方法は、リリース8.1.3以上でサポートされています。



注意:

LOBを含むペイロードの場合、エンキューおよびデキュー操作を実行するには、キュー表に対する明示的な Select、Insert および Update 権限を付与する必要があります。

関連項目:

[『Oracle Database SecureFilesおよびラージ・オブジェクト開発者ガイド』](#)

伝播の統計

伝播に関する詳細なランタイム情報が収集され、伝播スケジュールごとにDBA_QUEUE_SCHEDULESビューに格納されます。この情報はキューのデザイナーおよび管理者によって、問題の解決やパフォーマンス・チューニングのために使用できます。同様に、ビューによって報告されたエラーは、問題の診断および解決に使用できます。ビューには、伝播処理をしたセッションのID、ジョブ・キュー・プロセスの名前などの追加情報も示されます。

各スケジュールには、次の伝播の詳細情報が保持されます。

- スケジュールで伝播されたメッセージ合計数
- スケジュールの中で伝播されたバイト合計数
- 伝播枠の中で伝播されたメッセージの最大数
- 伝播枠の中で伝播されたバイトの最大値
- 伝播枠の中で伝播されたメッセージの平均数
- 伝播済メッセージの平均サイズ
- 伝播済メッセージの平均時間

伝播エラー処理

伝播機能には、障害対処およびエラー・レポートが組み込まれています。たとえば、指定されたデータベース・リンクが無効な場合、リモート・データベースが使用できない場合、またはリモート・キューにエンキューできない場合、適切なエラー・メッセージがレポートされます。伝播は線形バックオフ・スキームを使用して、障害が発生したスケジュールからの伝播を再試行します。

あるスケジュールで続けて障害が発生したときは、最初の再試行は30秒後、次の再試行は60秒後、3回目の再試行は120秒後、というように続きます。再試行時間が現行の伝播枠の期限切れ時刻を超える場合は、次の再試行は、次の伝播枠の開始時刻に行われます。最大16回の再試行が行われた後、そのスケジュールは自動的に使用不可になります。

注意:



再試行が次の伝播ウィンドウに移動されると、常に移動されるようになり、指数バックオフ・スキームは再試行のスケジュールを管理しません。DBMS_AQADM.SCHEDULE_PROPAGATION の next_time パラメータで指定された日付関数の結果、ウィンドウ間の間隔が短くなると、再試行の失敗数はすぐに 16 に達し、スケジュールが無効になります。

障害のためにスケジュールが自動的に使用不可になると、関連情報がアラート・ログに書き込まれます。スケジュール障害の確認項目は次のとおりです。

- 連続的な障害の発生回数
- 障害の原因を示すエラー・メッセージ
- 直前の障害の発生時刻

この情報を調べることで、キュー管理者は障害を回復し、スケジュールを使用可能にできます。再試行の間に伝播が成功したときは、障害の数は0(ゼロ)にリセットされます。

キューからデータベース・リンクへの伝播中のアプリケーション・エラーを示している状況では、Oracle Database Advanced QueuingはそのメッセージにUNDELIVERABLEというマークを付けてalert.logに記録します。このようなエラーは、リモート・キューが存在しない場合、またはソース・キューおよびリモート・キューの型が一致しない場合に発生します。background_dump_destディレクトリのトレース・ファイルには、そのエラーに関する追加情報があります。

新しいジョブ・キュー・プロセスが開始されると、型が一致しないエラーは、型を再検証できるように削除されます。ジョブ・キュー・プロセス数に上限を設定して、伝播のビジー状態が続く場合、ジョブ・キュー・プロセスが終了して再開するまで待つ必要はありません。キューの型は、必要に応じてDBMS_AQADM.VERIFY_QUEUE_TYPESを使用して再検証できます。

注意:



キューからキューへの伝播中に型の不一致が検出されると、伝播は停止してエラーが発生します。このような場合は、DBA_SCHEDULES ビューを問い合わせ、特定の宛先への伝播中に発生した最後のエラーを判断する必要があります。このメッセージには、UNDELIVERABLE マークは付いていません。

Oracle Real Application Clustersを使用した伝播

伝播には、Oracle Real Application Clustersサポートが組み込まれています。これはユーザーおよびキュー管理者からは透過的です。伝播を処理するジョブは、キューが常駐しているキュー表の所有者と同じインスタンスに送られます。

あるインスタンスに障害があって、ソース・キューを保存しているキュー表が他のインスタンスに移行される場合は、伝播ジョブも新しいインスタンスに移行されます。これによって、インスタンス間のping操作は最小限に抑えられ、パフォーマンスが向上します。

宛先は、データベース・リンクまたは宛先キュー名によって識別されます。宛先データベースを指定すると、キューからdblinkへの伝播が行われます。別のデータベースにある複数のキューにメッセージを伝播する場合、キューからdblinkへのすべての伝播が同じ頻度で行われます。宛先キュー名を指定すると、キューからキューへの伝播が行われます。別のデータベース内の複数のキューにメッセージを伝播すると、キューからキューへの伝播により、他のスケジュールに関係なく、各スケジュールの頻度を調整できます。個別の伝播を有効化または無効化することもできます。

この新しいキューからキューへの伝播モードは、宛先Oracle RACシステムに伝播する場合、透過的なフェイルオーバーもサポー

トします。キューからキューへの伝播では、Oracle RACでキューの所有者インスタンスが失敗した場合、データベース・リンクを再指定する必要はなくなります。

関連項目:

キューからキューへの伝播の詳細は、[「キューの伝播のスケジューリング」](#)を参照してください。

伝播は、同時スケジュールをいくつでも処理できるように設計されています。ジョブ・キュー・プロセスの最大数は1000で、その一部は伝播に関連しないジョブの処理に使用できます。このように、伝播にはマルチタスキングおよびロード・バランシングのサポートが組み込まれています。

伝播アルゴリズムは、複数スケジュールが単一ジョブ・キュー・プロセスによって処理できるように設計されています。ジョブ・キュー・プロセスに対する伝播の負荷は、異なるソース・キューのメッセージ到着の割合に基づいて偏りが発生する場合があります。

あるプロセスが数個のアクティブ・スケジュールによって過負荷になっている一方で、別のプロセスは受動的なスケジュールが多いために余力がある場合、伝播は負荷が均等になるようにスケジュールを自動的に再分散します。

サード・パーティのサポート

受信者のプロトコル番号が128から255の範囲にある場合、Oracle Database Advanced Queuingは受信者のアドレスを無視するため、メッセージがOracle Database Advanced Queuingシステムによって伝播されることはありません。かわりに、サード・パーティ製のプロパゲータが、コンシューマ名として予約済の名前をデキュー操作に指定して、メッセージをデキューできます。予約済のコンシューマ名は、AQ\$_Pprotocol_numberという書式で表されます。たとえば、AQ\$_P128というコンシューマ名は、プロトコル番号128の受信者に対してメッセージをデキューするために使用されます。特定のプロトコル番号を伴うメッセージの受信者リストは、デキュー時にrecipient_listメッセージ・プロパティによって戻されます。

Oracle Database Advanced Queuingは、メッセージ・ゲートウェイを使用して、サード・パーティ製のメッセージ・システム間でメッセージを伝播することもできます。メッセージ・ゲートウェイは、メッセージをOracle Database Advanced Queuingキューからデキューして、サポートされるサード・パーティ製のメッセージ・システムに確実に配信します。メッセージ・ゲートウェイは、これらのシステムからメッセージをデキューして、Oracle Database Advanced Queuingキューにエンキューすることもできます。

HTTPを使用した伝播

Oracle Database 10g以上では、HTTPおよびHTTPS(SSLによるHTTP)経由のOracle Database Advanced Queuing伝播を設定できます。HTTP伝播では、インターネット・アクセス・インフラストラクチャが使用され、接続先データベースに接続するOracle Database Advanced Queuingサブレットをデプロイする必要があります。データベース・リンクは、接続文字列にWebサーバーのアドレスとポートを指定し、HTTPをプロトコルとすることを指定して作成する必要があります。JavaおよびXMLを実行するためのソース・データベースを作成する必要があります。作成しないと、HTTP伝播の設定は、Oracle Net Services伝播の設定と同じになります。

メッセージ・フォーマットの変換

アプリケーションでは一般に、様々なフォーマットのデータが使用されます。変換により、Oracleデータ型間のマッピングが定義されます。

[変換](#)は、ソース・データ型を入力として取得し、ターゲット・データ型のオブジェクトを戻すSQLファンクションによって表現されます。1対1のメッセージ変換のみがサポートされます。

エンキュー中にメッセージを変換するには、エンキュー・オプションでマッピングを指定します。デキュー中にメッセージを変換するに

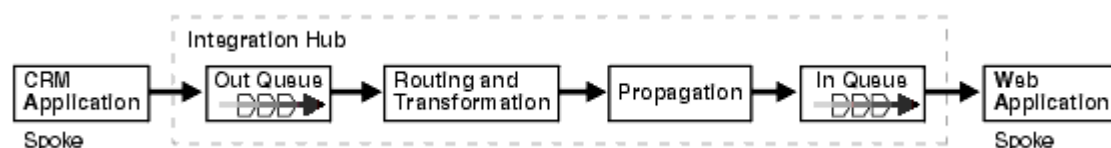
は、デキュー・オプションまたはサブスクリバの追加時にマッピングを指定します。デキュー・マッピングは、サブスクリバ・マッピングをオーバーライドします。伝播中にメッセージを変換するには、サブスクリバの追加時にマッピングを指定します。

変換を作成するには、単一のPL/SQLファンクションを作成するか、各ターゲット型の属性の式を作成します。PL/SQLファンクションでは、ターゲット型のオブジェクトまたはコンストラクタが戻されます。この表現は、単純な変換または各属性の個別の変換が困難なものに適しています。

ターゲット型の属性ごとに別々の式を作成すると、変換マッピングの作成および宛先タイプの各属性の管理が簡素化されます。これは、宛先タイプに多数の属性がある場合に有効です。

図1-10に示すとおり、キューイング機能、ルーティング機能および変換機能は、統合アプリケーション・アーキテクチャに不可欠な構成ブロックです。この図は、CRMアプリケーションのOutキューのデータが統合ハブで転送および変換され、その後、WebアプリケーションのInキューに伝播される方法を示しています。変換エンジンは、メッセージをOutキューのフォーマットからInキューのフォーマットにマップします。

図1-10 アプリケーション統合における変換



XMLデータ変換

XMLTypeでサポートされる `extract()` メソッドを使用してXMLデータを変換し、指定されたXPath式を適用した後に、XMLTypeのオブジェクトに戻すことができます。XSLPROCESSORパッケージを使用して、XSLT変換を適用することによってXMLTypeオブジェクトを変換するPL/SQLファンクションを作成することもできます。

その他のOracle Database Advanced Queuing機能

このトピックでは、キュー・モニター・コーディネータ、Oracle Internet Directoryとの統合、Oracle Enterprise Managerとの統合、保存とメッセージ履歴、メッセージ・キューのクリーン・アップ、追跡とイベント・ジャーナル、否認防止、インターネット統合などのAQ機能について説明します。

- [キュー・モニター・コーディネータ](#)
- [Oracle Internet Directoryとの統合](#)
- [Oracle Enterprise Managerとの統合](#)
- [保存およびメッセージ履歴](#)
- [メッセージ・キューのクリーン・アップ](#)
- [追跡およびイベント・ジャーナル](#)
- [否認防止](#)
- [インターネットの統合](#)

キュー・モニター・コーディネータ

10gリリース1 (10.1)より前のリリースでは、Oracle Database Advanced Queuingのタイム・マネージャ・プロセスは、キュー・モニター(QMNn)と呼ばれるバックグラウンド・プロセスとして、`init.ora`ファイルの動的なAQ_TM_PROCESSESパラメータの設定によって制御されていました。10gリリース1(10.1)以降は、時間管理をはじめとする他の多くのバックグラウンド・プロセスは、キュー・モニター・コーディネータ(QMNC)と呼ばれるコーディネータ対スレーブのアーキテクチャによって自動的に制御されています。QMNCは、システムの負荷に応じて、qXXXというスレーブを動的に起動します。このスレーブは、次のメカニズムを提供します。

- メッセージの遅延
- メッセージの期限切れ
- 再試行の遅延
- キュー表のガーベージ・コレクション
- バッファ済メッセージのメモリー管理タスク

プロセス数は自動的に決定されて絶えず調整されるため、AQ_TM_PROCESSESパラメータで設定する手間が省けます。

init.oraパラメータAQ_TM_PROCESSESの設定の必要性はなくなりましたが、引き続きサポートされています。このパラメータを設定しても(最大値は40)、プロセス数はQMNCによって自動的に調整されます。しかし、永続キューに設定したプロセス数だけは保証されます。ただし、[バッファ済キュー](#)のプロセスは、このパラメータの影響を受けません。

注意:



キュー・モニター・コーディネータを無効にするには、pfile または spfile で AQ_TM_PROCESSES = 0 を設定する必要があります。AQ_TM_PROCESSES パラメータはゼロ(0)に設定しないでください。

Oracle Internet Directoryとの統合

Oracle Internet Directoryは、Oracle Databaseに組み込まれたシステム固有のLDAPv3ディレクトリ・サービスで、電子メール・アドレス、電話番号、パスワード、セキュリティ証明書、様々なタイプのネットワーク・デバイスの構成データなど、多様な情報を集中管理します。企業全体のキューイング情報(キュー、サブスクリプションおよびイベント)は、Oracle Internet Directoryという1つの場所から検索できます。詳細は、[『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』](#)を参照してください。

Oracle Enterprise Managerとの統合

Oracle Enterprise Managerを使用すると、次の処理を実行できます。

- キュー、キュー表、伝播スケジュールおよび変換の作成および管理。
- データベース・レベルおよびキュー・レベルでのOracle Database Advanced Queuingトポロジの使用、およびキュー・エラー、キュー統計およびセッション統計の参照によるOracle Streams Advanced Queuing環境の監視。

保存およびメッセージ履歴

システム管理者は、処理済のメッセージの保存期間を指定します。Oracle Database Advanced Queuingは、各メッセージの履歴情報を格納し、キューおよびメッセージのプロパティ(遅延、期限切れ、およびローカルまたはリモートの受信者に対するメッセージの保存)を保存します。この情報には、エンキューおよびデキュー時刻、および各リクエストを実行したトランザクションの識別子が含まれます。これによって、ユーザーは関連するメッセージの履歴を保持できます。この履歴は、追跡、データ・ウェアハウスおよびデータ・マイニングの各操作の他に、特定の監査機能で使用できます。

バッファ済メッセージでは、メッセージの保存はサポートされていません。

メッセージ・キューのクリーン・アップ

Oracle Database Advanced Queuingの保存機能では、処理後のユーザー指定の有効期限が経過するとメッセージが自動的にクリーン・アップされます。

メッセージが不適切なサブスクリバのキューに誤って挿入された場合、サブスクリバ名またはメッセージ識別子を使用してデキューできます。これはメッセージを使用し、保存期間が過ぎた後にクリーン・アップされます。

特定のサブスクライバに対するメッセージをクリーン・アップするには、そのサブスクライバを削除して、その後再び追加します。サブスクライバを削除すると、そのサブスクライバに対するすべてのメッセージが削除されます。

追跡およびイベント・ジャーナル

保存されたメッセージを相互に関連付けて、順序を付けることができます。これらの順序はイベント・ジャーナルを表しており、一般にはアプリケーションによって作成されます。Oracle Database Advanced Queuingは、アプリケーションが自動的にイベント・ジャーナルを作成できるように設計されています。

否認防止

Oracle Database Advanced Queuingでは、メッセージ自体とともに、メッセージ情報のすべての履歴が保持されます。この情報によって、メッセージの送受信が証明され、送信者および受信者の否認防止に使用できます。

エンキュー時には、エンキュー元の否認防止のために次の情報が保持されます。

- エンキューを実行するOracle Database Advanced Queuingエージェント
- エンキューを実行するデータベース・ユーザー
- エンキュー時刻
- エンキューを実行するトランザクションのID

デキュー時には、デキュー元の否認防止のために次の情報が保持されます。

- デキューを実行するOracle Database Advanced Queuingエージェント
- デキューを実行するデータベース
- デキュー時刻
- デキューを実行するトランザクションのID

伝播後は、伝播の宛先キューのORIGINAL_MSGIDフィールドが、ソース・メッセージのメッセージIDに対応します。このフィールドは、伝播されたメッセージを相互に関連付けるために使用できます。これは、伝播されたメッセージのデキュー元の否認防止に有効です。

エンキュー時にメッセージとともに送信者のデジタル署名をエンキューし、デキュー時にデキュー元のデジタル署名を格納することによって、より強力な否認防止を実現できます。

インターネットの統合

インターネット上で[Simple Object Access Protocol \(SOAP\)](#)を使用してOracle Database Advanced Queuingにアクセスできます。[Internet Data Access Presentation \(IDAP\)](#)は、Oracle Database Advanced Queuing操作のためのSOAP仕様です。iDAPによってSOAPリクエストの本体にXMLメッセージ構造が定義されます。

IDAPメッセージでは、Oracle Database Advanced QueuingリクエストおよびレスポンスがXMLにカプセル化されます。IDAPは、エンキュー、デキュー、通知送信、通知登録、インターネットの標準転送プロトコルであるHTTP(S)や電子メールによる伝播などのOracle Database Advanced Queuing操作の実行に使用されます。さらに、iDAPでは、トランザクション、セキュリティ、変換、およびリクエストに対するキャラクタ・セットIDがカプセル化されます。

Oracle Internet Directory内のOracle Database Advanced Queuingエージェントに別名を作成し、この別名を、インターネット経由で送信されたIDAPドキュメントで使用して、Oracle Database Advanced Queuing操作を実行できます。別名を使用することで、Oracle Database Advanced Queuingエージェントの内部名を非公開にできます。

[図1-11](#)は、HTTP上でOracle Database Advanced Queuing操作を実行するためのアーキテクチャを示しています。主要コンポーネントは次のとおりです。

- Oracle Database Advanced Queuingクライアント・プログラム
- Oracle Database Advanced QueuingサブレットのホストWebサーバー/サブレット・コンテナ
- Oracle Databaseサーバー

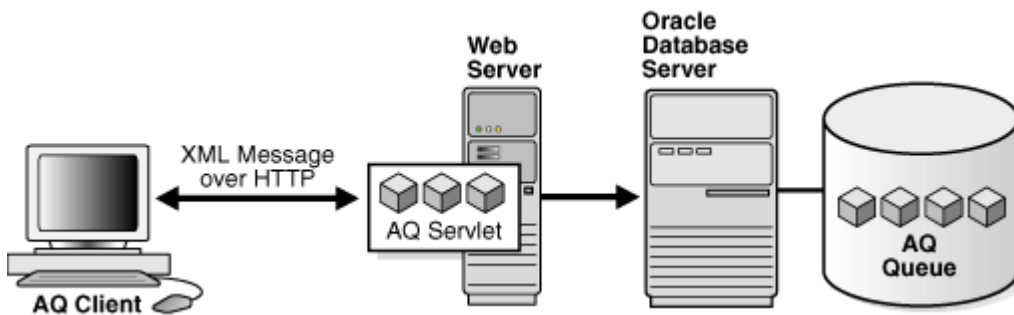
Oracle Database Advanced Queuingクライアント・プログラムは、XMLメッセージを理解し、Oracle Database Advanced Queuing操作を実行するOracle Database Advanced Queuingサブレットに、XMLメッセージ(IDAPに準拠)を送信します。Webブラウザなど、任意のHTTPクライアントを使用できます。Apache/JServまたはTomcatなど、Oracle Database Advanced QueuingサブレットをホストするWebサーバー/サブレット・ランナーにより、受信したXMLメッセージが解析されます。Oracle Database Advanced Queuingサブレットは、Oracle Databaseサーバーに接続し、ユーザー・キューに対して操作を実行します。

注意:



この機能は、Tomcat または JServ サブレット実行エンジンの他に、Apache での動作も保証されています。サブレットは、Java Servlet 2.0 以上のインターフェースをサポートする他の Web サーバーおよびサブレット実行エンジンでも動作します。

図1-11 HTTPを使用してOracle Database Advanced Queuing操作を実行するためのアーキテクチャ



Oracle Database Advanced Queuingへのインターフェース

Oracle Database Advanced Queuing機能は、このトピックに示されているインターフェースを介して使用できます。

- DBMS_AQ、DBMS_AQADMおよびDBMS_AQELMを使用したPL/SQL
- oracle.jms Javaパッケージを使用したJava Message Service(JMS)
- HTTP(S)を使用したインターネット・アクセス

注意:



oracle.AQ Java パッケージは、Oracle Database Advanced Queuing 10g リリース 1 (10.1) で非推奨になりました。既存の Java AQ アプリケーションを Oracle JMS に移行し、新しく設計する Java AQ アプリケーションには Oracle JMS を使用することをお勧めします。

関連項目:

- [『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)

2 Oracle Database Advanced Queuingの基本コンポーネント

このトピックでは、Oracle Database Advanced Queuingの基本コンポーネントを示します。

- [オブジェクト名](#)
- [タイプ名](#)
- [AQエージェント型](#)
- [AQ受信者リスト型](#)
- [AQエージェント・リスト型](#)
- [AQサブスクライバ・リスト型](#)
- [AQ登録情報リスト型](#)
- [AQ転送情報リスト型](#)
- [AQ登録情報型](#)
- [AQ通知記述子型](#)
- [AQメッセージ・プロパティ型](#)
- [AQ転送情報型](#)
- [AQ\\$_NTFN_MSGID_ARRAYタイプ](#)
- [AQ管理インターフェースの列挙定数](#)
- [AQ操作インターフェースの列挙定数](#)
- [AQバックグラウンド・プロセス](#)

関連項目:

- [Oracle Database Advanced Queuing管理インターフェース](#)
- [PL/SQLを使用したOracle Database Advanced Queuingの操作](#)

オブジェクト名

このコンポーネントは、データベース・オブジェクトに名前を付けます。

```
object_name := VARCHAR2  
object_name := [schema_name.] name
```

このネーミング規則は、キュー、キュー表およびオブジェクト型に適用されます。

オブジェクト名は、オプションの[スキーマ](#)名および名前で指定します。スキーマ名を指定しない場合は、現行のスキーマが想定されます。名前は、『[Oracle Database SQL言語リファレンス](#)』『[Oracle Database SQL言語リファレンス](#)』にある予約語のガイドラインに従って指定する必要があります。スキーマ名、エージェント名および[オブジェクト型](#)の名前はそれぞれ128バイト以下

である必要があります。一方、12cリリース2 (12.2)以降では、[キュー名](#)および[キュー表名](#)は、最大122バイトで指定できます。

タイプ名

このコンポーネントでは、キュー・タイプを定義します。

```
type_name := VARCHAR2  
type_name := object_type | "RAW"
```

オブジェクト型の属性数は最大で900です。

RAW型のペイロードを格納するために、Oracle Database Advanced Queuingでは、ペイロード・リポジトリとして[LOB](#)列を持つキュー表が作成されます。ペイロードのサイズは、32KB以下のデータにする必要があります。LOB列はRAWペイロードの格納に使用されるため、Oracle Databaseアドバンスド・キューイング管理者は、LOB表領域を選択して、キュー表作成時に `storage_clause` パラメータにLOB記憶域文字列を設定すると、LOB記憶域を構成できます。

注意:



LOBを含むペイロードの場合、エンキューおよびデキュー操作を実行するには、キュー表に対する明示的な Select、Insert および Update 権限を付与する必要があります。

AQエージェント型

このコンポーネントでは、メッセージのプロデューサまたはコンシューマを識別します。

```
TYPE AQ$_AGENT IS OBJECT (  
  name          VARCHAR2 (30) ,  
  address       VARCHAR2 (1024) ,  
  protocol      NUMBER) ;
```

マルチ・コンシューマ・キューにサブスクライバとして追加されたすべてのコンシューマは、AQ\$_AGENTパラメータに一意的値を持つ必要があります。2つのサブスクライバがAQ\$_AGENT型のNAME属性、ADDRESS属性およびPROTOCOL属性に同じ値を持つことはできません。この3つの属性のうち少なくとも1つは、2つのサブスクライバに対して異なる値にする必要があります。

多くのサブスクライバを追加するには、DBMS_AQADM.ADD_SUBSCRIBERプロシージャを繰り返し使用することで、1つのマルチ・コンシューマ・キュー当たり最大1024サブスクライバを追加できます。

この型は次の3つの属性を持ちます。

- name

この属性は、[メッセージのプロデューサ](#)または[コンシューマ](#)の名前を指定します。アプリケーション名か、アプリケーションによって割り当てられた名前のいずれかです。キュー自体が、別のキューからエンキューまたはデキューしているエージェントでも可能です。名前は、『[Oracle Database SQL言語リファレンス](#)』にある予約語のガイドラインに従って指定する必要があります。

- address

この属性はprotocolのコンテキストで解析されます。protocolが0(デフォルト)の場合、addressは [schema.]queue[@dblink]形式です。

- protocol

この属性では、アドレスを解析してメッセージを伝播するプロトコルを指定します。デフォルト値は0 (ゼロ)です。

AQ受信者リスト型

このコンポーネントでは、メッセージを受信するエージェントのリストを識別します。

```
TYPE AQ$_RECIPIENT_LIST_T IS TABLE OF aq$_agent  
INDEX BY BINARY_INTEGER;
```

AQエージェント・リスト型

このコンポーネントでは、DBMS_AQ.LISTENをリスニングするエージェントのリストを識別します。

```
TYPE AQ$_AGENT_LIST_T IS TABLE OF aq$_agent  
INDEX BY BINARY_INTEGER;
```

AQサブスクライバ・リスト型

このコンポーネントでは、このキューをサブスクライブするサブスクライバのリストを識別します。

```
TYPE AQ$_SUBSCRIBER_LIST_T IS TABLE OF aq$_agent  
INDEX BY BINARY_INTEGER;
```

AQ登録情報リスト型

このコンポーネントでは、キューへの登録リストを識別します。

```
TYPE AQ$_REG_INFO_LIST AS VARRAY(1024) OF sys.aq$_reg_info;
```

AQ転送情報リスト型

このコンポーネントでは、メッセージが転送される匿名サブスクリプションのリストを識別します。

```
TYPE AQ$_POST_INFO_LIST AS VARRAY(1024) OF sys.aq$_post_info;
```

AQ登録情報型

このコンポーネントでは、メッセージのプロデューサまたはコンシューマを識別します。

```
TYPE SYS.AQ$_REG_INFO IS OBJECT (  
  name VARCHAR2(128),  
  namespace NUMBER,  
  callback VARCHAR2(4000),  
  context RAW(2000) DEFAULT NULL,  
  qosflags NUMBER,  
  timeout NUMBER  
  ntfn_grouping_class NUMBER,  
  ntfn_grouping_value NUMBER DEFAULT 600,  
  ntfn_grouping_type NUMBER,  
  ntfn_grouping_start_time TIMESTAMP WITH TIME ZONE,  
  ntfn_grouping_repeat_count NUMBER);
```

このコンポーネントの属性は、次のとおりです。

表2-1 AQ\$_REG_INFO型属性

属性	説明
name	サブスクリプション名を指定します。単一コンシューマ・キューの登録の場合は <i>schema.queue</i> 形式のサブスクリプション名、マルチ・コンシューマ・キューの登録の場合は <i>schema.queue:consumer_name</i> 形式の名前となります。
namespace	サブスクリプションのネームスペースを指定します。Oracle Database AQ キューから通知を受信するには、namespace を DBMS_AQ.NAMESPACE_AQ に設定する必要があります。DBMS_AQ.POST または OCISubscriptionPost () を介して他のアプリケーションから通知を受信するには、namespace を DBMS_AQ.NAMESPACE_ANONYMOUS に設定する必要があります。
callback	メッセージ通知時に実行されるアクションを指定します。HTTP 通知の場合、 <code>http://www.company.com:8080</code> を使用します。電子メール通知の場合、 <code>mailto://xyz@company.com</code> を使用します。PLSQLCALLBACK プロシージャに対する RAW メッセージ・ペイロードの場合は、 <code>plsql://schema.procedure?PR=0</code> を指定します。PLSQLCALLBACK プロシージャに対する XML 変換済のユーザー定義タイプ・メッセージ・ペイロードの場合は、 <code>plsql://schema.procedure?PR=1</code> を指定します。
context	コールバック関数に渡すコンテキストを指定します。
qosflags	次の 1 つ以上の値に設定して、サービスの通知品質を指定できます。 <ul style="list-style-type: none"> ● NTFN_QOS_RELIABLE: この値は、信頼できる通知が必要であることを示します。信頼できる通知がインスタンス間で存続し、データベースが再起動します。 ● NTFN_QOS_PAYLOAD - この値は、ペイロード配信が必要であることを指定します。これは、クライアント通知および RAW キューに対してのみサポートされます。 ● NTFN_QOS_PURGE_ON_NTFN - この値は、この登録位置に最初の通知が配信されると、登録が自動的にパージされるように指定します。
ntfn_grouping_class	グループ化の基準を指定するために現在設定できるのは、次のフラグのみです。デフォルト値は 0(ゼロ)です。ntfn_grouping_class が 0(ゼロ)の場合、他のすべての通知グループ属性は 0(ゼロ)である必要があります。 <ul style="list-style-type: none"> ● NTFN_GROUPING_CLASS_TIME - 通知が時間別にグループ化されます。ユーザーが時間の値を指定すると、時間の終了時に通知がパブリッシュされます。
ntfn_grouping_value	グループ化通知の周期を秒で指定します。ntfn_grouping_repeat_count が使い果されるまで、グループ化通知が周期的に送信される時間を表します。

属性	説明
ntfn_grouping_type	<ul style="list-style-type: none"> ● NTFN_GROUPING_TYPE_SUMMARY - 間隔内で発生したすべての通知のサマリー。(デフォルト) ● NTFN_GROUPING_TYPE_LAST - 間隔内で発生した最後の通知。
ntfn_grouping_start_time	通知のグループ化の開始時間。通知のグループ化は、ユーザーが指定した時間に開始できますが、この時間はタイムゾーン付きの有効なタイムスタンプである必要があります。グループ化を使用する際に ntfn_grouping_start_time が指定されない場合、デフォルトでタイムゾーンの現在のタイムスタンプに設定されます。
ntfn_grouping_repeat_count	<p>グループ化の通知が通知のグループ化の繰返し件数で指定した回数送信された後、通常の通知に戻ります。ntfn_grouping_repeat_count を指定しない場合、デフォルトで次のように設定されます。</p> <ul style="list-style-type: none"> ● NTFN_GROUPING_FOREVER - グループ化の通知を永久的に送信します。

AQ通知記述子型

このコンポーネントは、通知時にAQ PL/SQLのコールバックが受信するOracle Databaseアドバンスド・キューイングの記述子を指定します。

```
TYPE SYS.AQ$_DESCRIPTOR IS OBJECT (
  queue_name      VARCHAR2(61),
  consumer_name   VARCHAR2(30),
  msg_id          RAW(16),
  msg_prop        MSG_PROP_T,
  gen_desc        AQ$_NTFN_DESCRIPTOR,
  msgid_array     SYS.AQ$_NTFN_MSGID_ARRAY,
  ntfnRecdInGrp  NUMBER);
```

次の属性が含まれます。

表2-2 AQ\$_DESCRIPTOR属性

属性	説明
queue_name	メッセージがエンキューされ、通知されたキューの名前を指定します。
consumer_name	マルチ・コンシューマ・キューのコンシューマの名前
msg_id	メッセージの指定番号
msg_prop	MSG_PROP_T 型で指定されたメッセージ・プロパティ
gen_desc	タイムアウト指定を示します。

属性	説明
msgid_array	グループ化の通知のメッセージ ID リスト
ntfnsRecdInGrp	グループで受信した通知

AQメッセージ・プロパティ型

メッセージ・プロパティ型msg_prop_tには、次のコンポーネントがあります。

```
TYPE AQ$_MSG_PROP_T IS OBJECT (
  priority      number,
  delay         number,
  expiration    number,
  correlation   varchar2(128),
  attempts     number,
  recipient_list aq$_recipient_list_t,
  exception_queue varchar2(51),
  enqueue_time  date,
  state        number,
  sender_id    aq$_agent,
  original_msgid raw(16),
  delivery_mode number);
```

タイムアウト指定型AQ\$_NTFN_DESCRIPTORには、次の単一コンポーネントがあります。

```
TYPE AQ$_NTFN_DESCRIPTOR IS OBJECT (
  NTFN_FLAGS number);
```

NTFN_FLAGSは、指定のタイムアウト後に通知が削除されている場合は1、それ以外の場合は0に設定されます。

関連項目:

[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のMESSAGE_PROPERTIES_T型に関する項

AQ転送情報型

このコンポーネントでは、メッセージの転送先となる匿名サブスクリプションを指定します。

```
TYPE SYS.AQ$_POST_INFO IS OBJECT (
  name          VARCHAR2(128),
  namespace    NUMBER,
  payload       RAW(2000));
```

次の3つの属性があります。

- name

この属性では、転送する匿名サブスクリプションの名前を指定します。

- namespace

この属性では、匿名サブスクリプションの名前空間を指定します。DBMS_AQ.POSTまたはOCISubscriptionPost()を

使用して他のアプリケーションから通知を受信するには、namespaceをDBMS_AQ.NAMESPACE_ANONYMOUSに設定する必要があります。

- payload

この属性では、匿名サブスクリプションに転送するペイロードを指定します。デフォルトはNULLです。

AQ\$_NTFN_MSGID_ARRAYタイプ

このコンポーネントは、AQ名前空間のグループ化の通知データを、最大VARRAYサイズ、値2³⁰で格納します。

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

AQ管理インタフェースの列挙定数

INFINITE、TRANSACTIONAL、NORMAL_QUEUEなどの列挙定数を値として選択する場合、定数はそれを定義するパッケージのスコープとともに指定する必要があります。

管理インタフェースに関連付けられているすべてのタイプには、DBMS_AQADMを付加する必要があります。次に例を示します。

```
DBMS_AQADM.NORMAL_QUEUE
```

[表2-3](#)は、Oracle Database Advanced Queuing管理インタフェースの列挙定数を示しています。

表2-3 Oracle Database Advanced Queuing管理インタフェースの列挙定数

パラメータ	オプション
retention	0, 1, 2... INFINITE
message_grouping	TRANSACTIONAL、NONE
queue_type	NORMAL_QUEUE、EXCEPTION_QUEUE、NON_PERSISTENT_QUEUE
delivery_mode	BUFFERED、PERSISTENT、PERSISTENT_OR_BUFFERED

注意:



非永続キューは、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨です。かわりにバッファ済メッセージを使用することをお勧めします。

AQ操作インタフェースの列挙定数

BROWSE、LOCKED、REMOVEなどの列挙定数を使用する場合、PL/SQL定数はそれを定義するパッケージのスコープとともに指定する必要があります。

操作インタフェースに関連付ける型の前には、必ずDBMS_AQADMを付加します。次に例を示します。

```
DBMS_AQ.BROWSE
```

表2-4は、Oracle Database Advanced Queuing操作インタフェースの列挙定数を示しています。

表2-4 Oracle Database Advanced Queuing操作インタフェースの列挙定数

パラメータ	オプション
visibility	IMMEDIATE、ON_COMMIT
dequeue mode	BROWSE、LOCKED、REMOVE、REMOVE_NODATA
navigation	FIRST_MESSAGE、NEXT_MESSAGE、NEXT_TRANSACTION
state	WAITING、READY、PROCESSED、EXPIRED
wait	FOREVER、NO_WAIT
delay	NO_DELAY
expiration	NEVER
namespace	NAMESPACE_AQ、NAMESPACE_ANONYMOUS
delivery_mode	BUFFERED、PERSISTENT、PERSISTENT_OR_BUFFERED
quosflags	NTFN_QOS_RELIABLE、NTFN_QOS_PAYLOAD、NTFN_QOS_PURGE_ON_NTFN
ntfn_grouping_class	NTFN_GROUPING_CLASS_TIME
ntfn_grouping_type	NTFN_GROUPING_TYPE_SUMMARY、NTFN_GROUPING_TYPE_LAST
ntfn_grouping_repeat_count	NTFN_GROUPING_FOREVER

AQバックグラウンド・プロセス

このトピックでは、Oracle Database Advanced Queuingのバックグラウンド・プロセスについて説明します。

- [キュー・モニター・プロセス](#)
- [ジョブ・キュー・プロセス](#)
- [AQバックグラウンド・アーキテクチャ](#)

キュー・モニター・プロセス

AQ_TM_PROCESSESパラメータは指定せず、システムの自動チューニングを利用することをお勧めします。

多数のOracle Database Advanced Queuingタスクが、バックグラウンドで実行されます。これには、READY状態に指定されたDELAYでのメッセージの変換、メッセージの期限切れ、例外キューへのメッセージの移動、バッファ済メッセージのオーバーフ

ローおよびリカバリ、その他の類似する操作が含まれます。

Oracle Database AQを使用するときに、AQ_TM_PROCESSESを設定する必要はなくなりました。値が指定されている場合、その値がQxxプロセスの開始時に考慮されます。ただし、Qxxプロセスの数は、AQ_TM_PROCESSESで指定した値とは異なる場合があります。

バックグラウンド・プロセスを使用不可にするAPIと使用可能にするAPIを分離する必要はありません。その制御は、AQ_TM_PROCESSESをゼロまたはゼロ以外に設定することで行います。ただし、AQ_TM_PROCESSESパラメータは指定せず、システムの自動チューニングを利用することをお勧めします。

注意:



キュー・モニター・コーディネータを無効にするには、pfile または spfile で AQ_TM_PROCESSES = 0 を設定する必要があります。AQ_TM_PROCESSES パラメータはゼロ(0)に設定しないでください。

ジョブ・キュー・プロセス

伝播およびPL/SQL通知は、ジョブ・キュー(Jnnn)・プロセスによって処理されます。

JOB_QUEUE_PROCESSESパラメータを指定する必要はなくなりました。データベース・スケジューラが、伝播および通知ジョブに必要なジョブ・キュー・プロセスを自動的に開始します。

AQバックグラウンド・アーキテクチャ

Oracle Database Advanced Queuing 12cリリース1 (12.1)では、3層設計のAQバックグラウンド・アーキテクチャが新たに採用されています。

- Tier1 (AQPC): アドバンスド・キューイング・プロセス・コーディネータと呼ばれる単一のバックグラウンド・プロセスがインスタンスの開始時に作成されます。これは、様々なマスター・プロセスの作成および管理を実行します。コーディネータ統計は、GV\$AQ_BACKGROUND_COORDINATORを使用して表示できます。
- Tier2 (QM**): キュー・モニターと呼ばれる多くのマスター・プロセスがあります。それぞれ、異なるタイプのジョブを処理します。ジョブのタイプには、通知(Emonプール)、キュー・モニター(シャード・キュー時間マネージャの処理)、クロス・プロセスなどがあります。

注意:



QMNC および EMNC などの古いプロセスは、新しいマスター・プロセスのいずれかに含まれます。

ジョブは、そのタスクを実行するために、複数のサーバー・プロセス(Q***)に渡る独自のスケジュール・メカニズムを必要とする作業タイプとして定義できます。マスター・プロセス統計およびそのジョブは、GV\$AQ_JOB_COORDINATORを使用して表示できます。

- Tier3(Q***): 前述のすべてのマスター・プロセスに対する単一のサーバー・プロセス・プールがあります。各プロセスは、一度に1つのマスター・プロセスに関連付けられます。ただし、元のマスターを使用する必要性がなくなった場合、別のマスターに再スケジュールできます。これらのサーバーにより、各マスターのジョブが実行され、パフォーマンスおよびスケラビ

リティが向上します。サーバー・プロセス統計および現在のマスターの関連付けは、GV\$AQ_SERVER_POOLを使用して表示できます。

3 Oracle Database Advanced Queuing: プログラム・インタフェース

次のトピックでは、作業に必要な様々な言語オプションと要素、およびOracle Database Advanced Queuing (AQ)アプリケーション環境を準備するときに考慮する必要がある問題点を説明します。

注意:



Java パッケージ oracle. AQ は、10g リリース 1(10.1)では非推奨になりました。既存の Java AQ アプリケーションを Oracle JMS(または他の Java API)に移行し、新しく設計する Java AQ アプリケーションには Oracle JMS(または他の Java API)を使用することをお勧めします。

トピック:

- [Oracle Database Advanced Queuingへのアクセスに必要なプログラム・インタフェース](#)
- [PL/SQLを使用したOracle Database Advanced Queuingへのアクセス](#)
- [OCIを使用したOracle Database Advanced Queuingへのアクセス](#)
- [OCCIを使用したOracle Database Advanced Queuingへのアクセス](#)
- [Oracle Java Message Service \(Oracle JMS\)を使用したOracle Database Advanced Queuingへのアクセス](#)
- [Oracle Database Advanced Queuing XMLサブレットを使用したOracle Database Advanced Queuingへのアクセス](#)
- [Oracle Database Advanced Queuingのプログラム・インタフェース別の比較](#)

Oracle Database Advanced Queuingへのアクセスに必要なプログラム・インタフェース

この表は、Oracle Database Advanced Queuingのプログラム・インタフェースと、各インタフェースでサポートされる機能、および構文の参照先を示しています。

表3-1 Oracle Database Advanced Queuingのプログラム・インタフェース

言語	プリコンパイラまたはインタフェース・プログラム	サポートされる機能	構文の参照先
PL/SQL	DBMS_AQADM および DBMS_AQ パッケージ	管理および操作	『Oracle Database PL/SQL パッケージおよびタイプ・リファレンス』
C	Oracle Call Interface(OCI)	操作のみ	『Oracle Call Interface プログラマーズ・ガイド』

言語	プリコンパイラまたはインタフェース・プログラム	サポートされる機能	構文の参照先
Java (JMS)	JDBC API を介しての Oracle JMS パッケージ	管理および操作	『Oracle Database Advanced Queuing Java API リファレンス』
AQ XML サブレット	Internet Data Access Presentation (iDAP)	操作のみ	『Oracle XML DB 開発者ガイド』

PL/SQLを使用したOracle Database Advanced Queuingへのアクセス

PL/SQLパッケージDBMS_AQADMおよびDBMS_AQは、ネイティブOracle Database Advanced Queuingインタフェースを使用した、Oracle Database Advanced Queuingの管理および操作機能へのアクセスをサポートします。

これらの機能には、次のものがあります。

- [キュー](#)、[キュー表](#)、[非永続キュー](#)、[シャード・キュー](#)、マルチ・コンシューマ・キュー/トピック、RAW [メッセージ](#)または構造化データを持つメッセージの作成
- キュー表、キュー、マルチ・コンシューマ・キュー/トピックの取得
- キュー表、キュー/トピックの変更
- キュー/トピックの削除
- キュー/トピックの開始または停止
- 権限の付与および取消し
- [サブスクライバ](#)の追加、削除、変更
- Oracle Database Advanced Queuingインターネット・エージェントの追加、削除または変更
- Oracle Database Advanced Queuingインターネット・エージェントに対するデータベース・ユーザー権限の付与または取消し
- [伝播スケジュール](#)の有効化、無効化、変更
- シングル・[コンシューマ](#)・キュー(Point-to-Pointモデル)へのメッセージのエンキュー
- マルチ・コンシューマ・キュー/トピック([パブリッシュ・サブスクライブ](#)モデル)へのメッセージのパブリッシュ
- マルチ・コンシューマ・キューのメッセージに対するサブスクライブ
- キューのメッセージのブラウズ
- キュー/トピックからのメッセージの受信
- メッセージの非同期受信登録
- メッセージのリスニング: 複数キュー/トピック
- 匿名サブスクリプションへのメッセージの転送
- [Lightweight Directory Access Protocol](#)(LDAP)サーバーのエージェントのバインドまたはアンバインド
- LDAPサーバーのOracle Database Advanced Queuingオブジェクトに対する別名の追加または削除

PL/SQLのDBMS_AQADMおよびDBMS_AQで使用可能な機能の詳細は、表3-2から表3-9を参照してください。

関連項目:

構文、パラメータ、パラメータ・タイプ、戻り値、例を含むDBMS_AQADMおよびDBMS_AQの詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』を参照してください。

OCIおよびThin JDBCドライバを使用したOracle Database Advanced Queuingへのアクセス

Oracle Call Interface (OCI)は、ネイティブOracle Database Advanced Queuingインタフェースを介して、Oracle Database Advanced Queuingの各機能へのインタフェースを提供します。

OCIクライアントは、次のアクションを実行できます。

- メッセージのエンキュー
- メッセージのデキュー
- 複数キュー上のメッセージのリスニング
- メッセージ通知の受信登録

さらに、OCIクライアントは、OCISubscriptionRegisterを使用して、キューの新規メッセージの[非同期](#)通知を受信できます。シャード・キューは、OCIクライアントをサポートしていません。

Oracle Type Translator

ユーザー定義のペイロード型のキューでは、OracleタイプのOCI/OCCIマッピングの生成にOracle Type Translatorを使用する必要があります。OCIクライアントは、Oracle Database Advanced Queuing記述子のメモリーおよびメッセージ・ペイロードを解放する必要があります。

関連項目:

構文の詳細は、『[Oracle Call Interfaceプログラマーズ・ガイド](#)』のOCIおよびアドバンスド・キューイングおよびパブリッシュ・サブスクライブの通知に関する項を参照してください。

Oracle Database Advanced Queuingのプログラム・インタフェース別の比較

次の各トピックでは、Oracle Database Advanced Queuing管理インタフェースとOracle Database Advanced Queuing操作インタフェースを一覧および比較しています。

表3-2から表3-9は、Oracle Database Advanced Queuingの各プログラム・インタフェースで使用可能な機能を用途別に示したものです。ユース・ケースについては、『[Oracle Database Advanced Queuing管理インタフェースからPL/SQLを使用したOracle Database Advanced Queuing操作](#)』、および『[Oracle Java Message Serviceの基本操作からOracle Java Message Serviceの共有インタフェース](#)』で説明しています。

Oracle Database Advanced Queuing管理インタフェース

この表は、Oracle Database Advanced Queuingの管理における、PL/SQLとJava (JMS)のプログラム・インタフェースの同等の機能を示しています。

表3-2 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 管理インタフェース

ユース・ケース	PL/SQL	Java (JMS)
コネクション・ファクトリの作成	該当なし	AQjmsFactory.getQueue ConnectionFactoryAQjmsFactory.getTopic ConnectionFactory
LDAP サーバーでの ConnectionFactory の 登録	該当なし	AQjmsFactory.register ConnectionFactory
キュー表の作成	DBMS_AQADM.CREATE_QUE UE_TABLE	AQjmsSession.createQueueTable
キュー表の取得	<i>schema.queue_table_na me</i> を使用します	AQjmsSession.getQueueTable
キュー表の変更	DBMS_AQADM.ALTER_QUEU E_TABLE	AQQueueTable.alter
キュー表の削除	DBMS_AQADM.DROP_QUEUE _TABLE	AQQueueTable.drop
キューの作成	DBMS_AQADM.CREATE_QUE UE	AQjmsSession.createQueue
キューの取得	<i>schema.queue_name</i> を使 用します	AQjmsSession.getQueue
マルチ・コンシューマ対応の キュー表でのマルチ・コン シューマ・キュー/トピックの作 成	DBMS_AQADM.CREATE_QUE UE	AQjmsSession.createTopic
マルチ・コンシューマ・キュー/ トピックの取得	<i>schema.queue_name</i> を使 用します	AQjmsSession.getTopic
キュー/トピックの変更	DBMS_AQADM.ALTER_QUEU E	AQjmsDestination.alter
キュー/トピックの開始	DBMS_AQADM.START_QUEU E	AQjmsDestination.start
キュー/トピックの停止	DBMS_AQADM.STOP_QUEUE	AQjmsDestination.stop

ユース・ケース	PL/SQL	Java (JMS)
キュー/トピックの削除	DBMS_AQADM. DROP_QUEUE	AQjmsDestination. drop
シャード・キューの作成	DBMS_AQADM. CREATE_SHARDED_QUEUE	AQjmsDestination. createJMSShardedQueue
シャード・キューの削除	DBMS_AQADM. DROP_SHARDED_QUEUE	AQjmsDestination. dropJMSShardedQueue
シャード・キューの変更	DBMS_AQADM. ALTER_SHARDED_QUEUE	なし。PL/SQL API を使用してください。
システム権限の付与	DBMS_AQADM. GRANT_SYSTEM_PRIVILEGE	AQjmsSession. grantSystem Privilege
システム権限の取消し	DBMS_AQADM. REVOKE_SYSTEM_PRIVILEGE	AQjmsSession. revokeSystem Privilege
キュー/トピック権限の付与	DBMS_AQADM. GRANT_QUEUE_PRIVILEGE	AQjmsDestination. grantQueue Privilege AQjmsDestination. grantTopic Privilege
キュー/トピック権限の取消し	DBMS_AQADM. REVOKE_QUEUE_PRIVILEGE	AQjmsDestination. revokeQueue Privilege AQjmsDestination. revokeTopic Privilege
キュー・タイプの検証	DBMS_AQADM. VERIFY_QUEUE_TYPES	サポートされません
サブスクライバの追加	DBMS_AQADM. ADD_SUBSCRIBER	表 3-6 を参照してください
サブスクライバの変更	DBMS_AQADM. ALTER_SUBSCRIBER	表 3-6 を参照
サブスクライバの削除	DBMS_AQADM. REMOVE_SUBSCRIBER	表 3-6 を参照
伝播のスケジューリング	DBMS_AQADM. SCHEDULE_PROPAGATION	AQjmsDestination. schedule Propagation
伝播スケジュールの有効化	DBMS_AQADM. ENABLE_PROPAGATION_SCHEDULE	AQjmsDestination. enable PropagationSchedule
伝播スケジュールの変更	DBMS_AQADM. ALTER_PROPAGATION_SCHEDULE	AQjmsDestination. alter PropagationSchedule
伝播スケジュールの無効化	DBMS_AQADM. DISABLE_PROPAGATION_SCHEDULE	AQjmsDestination. disable PropagationSchedule
伝播スケジュールの解除	DBMS_AQADM. UNSCHEDULE_PROPAGATION	AQjmsDestination. unschedule Propagation
Oracle Database Advanced Queuing インターネット・エージェントの作成	DBMS_AQADM. CREATE_AQ_AGENT	サポートされません

ユース・ケース	PL/SQL	Java (JMS)
Oracle Database Advanced Queuing インターネット・エージェントの変更	DBMS_AQADM.ALTER_AQ_AGENT	サポートされません
Oracle Database Advanced Queuing インターネット・エージェントの削除	DBMS_AQADM.DROP_AQ_AGENT	サポートされません
Oracle Database Advanced Queuing インターネット・エージェントへのデータベース・ユーザー権限の付与	DBMS_AQADM.ENABLE_AQ_AGENT	サポートされません
Oracle Database Advanced Queuing インターネット・エージェントからのデータベース・ユーザー権限の取消し	DBMS_AQADM.DISABLE_AQ_AGENT	サポートされません
LDAP サーバーにおけるキュー、エージェント、ConnectionFactory の別名の追加	DBMS_AQADM.ADD_ALIAS_TO_LDAP	サポートされません
LDAP サーバーにおけるキュー、エージェント、ConnectionFactory 用の別名の削除	DBMS_AQADM.DEL_ALIAS_FROM_LDAP	サポートされません

Oracle Database Advanced Queuing操作インタフェース

次の各表は、様々な利用方法別に、Oracle Database Advanced Queuingの操作における、プログラム・インタフェース PL/SQL、OCI、Oracle Database Advanced Queuing XMLサブレットおよびJMSの同等の機能を示しています。

表3-3 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較：操作インタフェース—コネクション、セッション、メッセージの作成での利用方法

ユース・ケース	AQ XMLサー			
	PL/SQL	OCI	ブレット	JMS

ユース・ケース	PL/SQL	OCI	AQ XMLサー ブレット	JMS
接続の作成	該当なし	OCIServer Attach	Web サーバー を使用して、認 証後に HTTP コネクションを オープンする。	AQjmsQueueConnectionFactory .createQueueConnectio TopicConnectionFactory .createTopicConnection
セッションの作 成	該当なし	OCISession Begin	HTTP サーブ レットは最初の SOAP リクエス トで自動的に 開始される。	QueueConnection.createQueue SessionTopicConnection.createTopic Session
RAW メッセー ジの作成	メッセージに SQL RAW 型 を使用する。	メッセージに OCIRaw を使 用する。	XML メッセージ にメッセージ・ペ イロードの 16 進表示を提供 する。例: <raw>023f452 3</raw>	サポートされません
構造化データ を持つメッセー ジの作成	メッセージに SQL Oracle オブジェクト型 を使用する。	メッセージに SQL Oracle オブジェクト型 を使用する。	JMS キュー以 外の (AQ\$_JMS_*型 ではな い)Oracle オ ブジェクト型 キューでは、 <message payload>に指 定されている XML をキュー 表のペイロード の SQL 型に マッピングする 必要がある。 JMS キューで は、 <jms_text_me ssage>、 <jms_map_mes sage>、 <jms_bytes_m	Session.createTextMessage Session.createObjectMes Session.createMapMessage Session.createBytesMessa Session.createStreamMessage AQjmsSession.createAdtMessage

ユース・ケース	PL/SQL	OCI	AQ XMLサー ブレット	JMS
			essage>、 <jms_object_ message>のい ずれかを、 <message_pay load>内の XML に指定す る必要がある。	
メッセージ・プロ デューサの作成	該当なし	該当なし	該当なし	QueueSession.createSender TopicSession.createPubl

表3-4 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—シングル・コンシューマ・キューへのメッセージのエンキュー(Point-to-Pointモデルでの利用方法)

ユース・ケース	PL/SQL	OCI	AQ XMLサーブレッ ト	JMS
シングル・コン シューマ・キュー へのメッセージ のエンキュー	DBMS_AQ. enqueue	OCI AQEnq	<AQXmlSend>	QueueSender.send
キューへのメッ セージのエン キュー: 可視 性オプションの 指定	DBMS_AQ. enqueue 次のものに visibility を指 定する。	OCI AQEnq OCI AQEnqOptions に OCI_ATTR_VISIBILITY を指定する。 OCI AQEnqOptions	<AQXmlSend> 次のものに <visibility>を指 定する <producer_options >	サポートされません
シングル・コン シューマ・キュー へのメッセージ のエンキュー: メッセージ・プロ パティの優先順 位と期限切れ の指定	DBMS_AQ. enqueue 次のものに priority、 expiration を 指定する。 MESSAGE_PROP ERTIES	OCI AQEnq 次のものに OCI_ATTR_PRIORITY、 OCI_ATTR_EXPIRATION を指定する。 OCI AQMsgProperties	<AQXmlSend> 次のものに <priority>、 <expiration>を指 定する。 <message_header>	の実行時に、priority および TimeToLive を指定する。 QueueSender.send または .setTimeToLive および MessageProducer.setPriority

ユース・ケース	PL/SQL	OCI	AQ XMLサブプレット	JMS
				その後に次を指定します QueueSender.send
シングル・コンシューマ・キューへのメッセージのエンキュー: メッセージ・プロパティの相関識別子、遅延、 例外キュー の指定	DBMS_AQ.enqueue 次のものに correlation、delay、exception_queue を指定する。 MESSAGE_PROPERTIES	OCIAQEnq 次のものに OCI_ATTR_CORRELATION、OCI_ATTR_DELAY、OCI_ATTR_EXCEPTION_QUEUE を指定する。 OCIAQMsgProperties	<AQXmlSend> 次のものに <correlation_id>、<delay>、<exception_queue> を指定する。 <message_header>	Message.setJMSCorrelationID 遅延および例外キューは、次のプロパティ固有のメッセージ・プロパティで指定。 JMSOracleDelay JMSOracleException その後に次を指定します QueueSender.send
シングル・コンシューマ・キューへのメッセージのエンキュー: ユーザー定義のメッセージ・プロパティの指定	サポートされません プロパティはペイロードの一部である必要がある。	サポートされません プロパティはペイロードの一部である必要がある。	<AQXmlSend> 次のものに、<name>と <int_value>、<string_value>、<long_value>などを指定する。 <user_properties>	Message.setIntProperty Message.setStringProperty Message.setBooleanProperty など。その後にを指定する。 QueueSender.send
シングル・コンシューマ・キューへのメッセージのエンキュー: メッセージ変換 の指定	DBMS_AQ.enqueue 次のものに transformation を指定する。 ENQUEUE_OPTIONS	OCIAQEnq 次のものに OCI_ATTR_TRANSFORMATION を指定する。 OCIAQEnqOptions	<AQXmlSend> 次のものに <transformation> を指定する <producer_options>	AQjmsQueueSender.setTransformation その後に次を指定します QueueSender.send

表3-5 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—マルチ・コンシューマ・キュー/トピックへのメッセージのパブリッシュ(パブリッシュ・サブスクライブ・モデルでの利用方法)

ユース・ケース	PL/SQL	OCI	AQ XMLサブプレット	JMS
マルチ・コンシューマ・キュー/トピックへのメッセージのパブリッシュ: デフォルト	DBMS_AQ.enqueue 次のものの recipient_list を NULL に	OCIAQEnq 次のものの OCI_ATTR_RECIPIENT_LIST を NULL に設定する。	<AQXmlPublish>	TopicPublisher.publish

ユース・ケース	PL/SQL	OCI	AQ XMLサブプレット	JMS
のサブスクリプション・リストの使用	設定する。 MESSAGE_PROPERTIES	OCIAQMsgProperties		
マルチ・コンシューマ・キュー / トピックへのメッセージのパブリッシュ: 特定の受信者リストの使用 脚注 1 を参照	DBMS_AQ.enqueue 次のものに受信者リストを指定する。 MESSAGE_PROPERTIES	OCIAQEnq 次のものに OCI_ATTR_RECIPIENT_LIST を指定する。 OCIAQMsgProperties	<AQXmlPublish> 次のものに <recipient_list> を指定する。 <message_header>	AQjmsTopic Publisher.publish AQjmsAgent の配列として受信者を指定する。
マルチ・コンシューマ・キュー / トピックへのメッセージのパブリッシュ: メッセージ・プロパティの優先順位と期限切れの指定	DBMS_AQ.enqueue 次のものに priority、 expiration を指定する。 MESSAGE_PROPERTIES	OCIAQEnq 次のものに OCI_ATTR_PRIORITY、 OCI_ATTR_EXPIRATION を指定する。 OCIAQMsgProperties	<AQXmlPublish> 次のものに <priority>、 <expiration>を指定する。 <message_header>	の実行時に、priority および TimeToLive を指定する。 TopicPublisher.publish または MessageProducer.setTimeToLive および MessageProducer.setPriority その後次に指定します TopicPublisher.publish
マルチ・コンシューマ・キュー / トピックへのメッセージのパブリッシュ: 送信オプション(相関識別子、遅延、例外キュー)の指定	DBMS_AQ.enqueue 次のものに correlation、 delay、 exception_queue を指定する。 MESSAGE_PROPERTIES	OCIAQEnq 次のものに OCI_ATTR_CORRELATION_ID、 OCI_ATTR_DELAY、 OCI_ATTR_EXCEPTION_QUEUE を指定する。 OCIAQMsgProperties	<AQXmlPublish> 次のものに <correlation_id>、 <delay>、 <exception_queue> を指定する。 <message_header>	Message.setJMS CorrelationID 遅延および例外キューは、次のプロパティ固有のメッセージ・プロパティで指定。 JMS_OracleDelay JMS_OracleExcpQ その後次に指定します TopicPublisher.publish
トピックへのメッセージのパブリッシュ	サポートされません	サポートされません プロパティはペイロードの一	<AQXmlPublish> 次のものに、<name>と	Message.setInt PropertyMessage.setString PropertyMessage.setBoolean

ユース・ケース	PL/SQL	OCI	AQ XMLサブレット	JMS
シュ: ユーザー定義のメッセージ・プロパティの指定	プロパティはペイロードの一部である必要がある。	部である必要がある。	<int_value>、 <string_value>、 <long_value>などを指定する。 <user_properties>	Property など。その後を指定する。 TopicPublisher. publish
トピックへのメッセージのパブリッシュ シュ: メッセージ変換の指定	DBMS_AQ.enqueue 次のものに transformationを指定する。 ENQUEUE_OPTIONS	OCIAQEnq 次のものに OCI_ATTR_TRANSFORMATIONを指定する。 OCIAQEnqOptions	<AQxmlPublish> 次のものに <transformation>を指定する <producer_options>	AQjmsTopic Publisher.set Transformation その後次に指定します TopicPublisher. publish

表3-6 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—マルチ・コンシューマ・キュー/トピックのメッセージに対するサブスクライブ(パブリッシュ・サブスクライブ・モデルでの利用方法)

ユース・ケース	PL/SQL	OCI	AQ XMLサブレット	JMS
サブスクライバの追加	管理インタフェースを参照	サポートされません	サポートされません	TopicSession. createDurableSubscriberAQjmsSession. createDurableSubscriber
サブスクライバの変更	管理インタフェースを参照	サポートされません	サポートされません	TopicSession. createDurableSubscriberAQjmsSession. createDurableSubscriber 新しいセレクトクを使用する。
サブスクライバの削除	管理インタフェースを参照	サポートされません	サポートされません	AQjmsSession. unsubscribe

表3-7 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—キューのメッセージのブラウズでの利用方法

ユース・ケース	PL/SQL	OCI	AQ XMLサブレット	JMS
キュー/トピックのメッセージのブラウズ	DBMS_AQ.dequeue 次のものの dequeue_modeをBROWSEに	OCIAQDeq 次のものの OCI_ATTR_DEQ_MODEをBROWSEに設定す	<AQxmlReceive> 次のものの <dequeue_mode>をBROWSEに指定	QueueSession. createBrowserQueueBrowser.getE トピックでは未サポート oracle. jms. AQjmsSession. createBrowseroracle. jms. TopicBrowser. getEn

ユース・ケース			AQ XMLサブ	
	PL/SQL	OCI	レット	JMS
	設定する。	る。	する。	
	DEQUEUE_OPTIONS	OCIAQDeqOptions	<consumer_options>	
キュー/トピックのメッセージのブラウズ:	DBMS_AQ.dequeue	OCIAQDeq	<AQXmlReceive>	AQjmsSession.createBrowser
ブラウズ中のメッセージのロック	次のものの dequeue_mode を LOCKED に設定する。	次のものの OCI_ATTR_DEQ_MODE を LOCKED に設定する。	次のものの <dequeue_mode> を LOCKED に設定する。	次のものの locked を TRUE に設定する。 QueueBrowser.getEnumeration トピックでは未サポート
	DEQUEUE_OPTIONS	OCIAQDeqOptions	<consumer_options>	oracle.jms.AQjmsSession.createBrowser oracle.jms.TopicBrowser.getEn

表3-8 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—キュー/トピックからのメッセージの受信での利用方法

ユース・ケース			AQ XMLサブ	
	PL/SQL	OCI	レット	JMS
メッセージ受信用の接続の起動	該当なし	該当なし	該当なし	Connection.start
メッセージ・コンシューマの作成	該当なし	該当なし	該当なし	QueueSession.createQueueReceiverTopicSession.createDurableSubscriber AQjmsSession.createTopicReceiver
キュー/トピックからのメッセージのデキュー: 可視性の指定	DBMS_AQ.dequeue 次のものに visibility を指定する。	OCIAQDeq 次のものに OCI_ATTR_VISIBILITY を指定する。	<AQXmlReceive> 次のものに <visibility>を指定する	サポートされません
キュー/トピックからのメッセージのデキュー: 変換の指定	DBMS_AQ.dequeue 次のものに transformation を指定する。	OCIAQDeq 次のものに OCI_ATTR_TRANSFORMATION を指定する。	<AQXmlReceive> 次のものに <transformation>を指定する	AQjmsQueueReceiver.setTransformation AQjmsTopicSubscriber.setTransformation AQjmsTopicReceiver.setTransformation
	DEQUEUE_OPTIONS	OCIAQDeqOptions	<consumer_options>	
	DEQUEUE_OPTIONS	OCIAQDeqOptions	<consumer_options>	

ユース・ケース	PL/SQL	OCI	AQ XMLサブレット	JMS
キュー/トピックからのメッセージのデキュー: ナビゲーション・モードの指定	DBMS_AQ.dequeue 次のものに navigation を指定する。 DEQUEUE_OPTIONS	OCIAQDeq 次のものに OCI_ATTR_NAVIGATION を指定する。 OCIAQDeqOptions	<AQXMLReceive> 次のものに <navigation>を指定する <consumer_options>	AQjmsQueueReceiver. setNavigationModeAQjmsTopicSubscriber setNavigationModeAQjmsTopicReceiver setNavigationMode
シングル・コンシューマ・キューからのメッセージのデキュー	DBMS_AQ.dequeue 次のものの dequeue_mode を REMOVE に 設定する。 DEQUEUE_OPTIONS	OCIAQDeq 次のものの OCI_ATTR_DEQ_MODE を REMOVE に 設定する。 OCIAQDeqOptions	<AQXMLReceive>	QueueReceiver.receive または QueueReceiver.receive NoWait または AQjmsQueueReceiver.receiveNoData
マルチ・コンシューマ・キュー/トピックからのメッセージのデキュー: サブスクリプション名を使用	DBMS_AQ.dequeue 次のものの dequeue_mode を REMOVE に 設定し、 consumer_name にサブスクリプション名を設定する。 DEQUEUE_OPTIONS	OCIAQDeq 次のものの OCI_ATTR_DEQ_MODE を REMOVE に 設定し、 OCI_ATTR_CONSUMER_NAME にサブスクリプション名を設定する。 OCIAQDeqOptions	<AQXMLReceive> 次のものに <consumer_name>を指定する。 <consumer_options>	サブスクリプション名を使用してトピックに永続 TopicSubscriberを作成し、その後次の 指定する。 TopicSubscriber.receive または TopicSubscriber.receiveNoWait または AQjmsTopicSubscriber.receiveNoData
マルチ・コンシューマ・キュー/トピックからのメッセージのデキュー: 受信者名を使用	DBMS_AQ.dequeue 次のものの dequeue_mode を REMOVE に 設定し、 consumer_name に受信者名 を設定する。 DEQUEUE_OPTIONS	OCIAQDeq 次のものの OCI_ATTR_DEQ_MODE を REMOVE に 設定し、 OCI_ATTR_CONSUMER_NAME に受信者名 を設定する。 OCIAQDeqOptions	<AQXMLReceive> 次のものに <consumer_name>を指定する。 <consumer_options>	受信者名を使用してトピックに TopicReceiver する。その後、次のいずれかを指定する。 AQjmsSession.create TopicReceiverAQjmsTopicReceiver.receive または AQjmsTopicReceiver.receiveNoWait または AQjmsTopicReceiver.receiveNoData

表3-9 Oracle Database Advanced Queuingのプログラム・インタフェース別の比較: 操作インタフェース—キュー/トピック

ユース・ケース	PL/SQL	OCI	AQ XMLサブレット	JMS
メッセージの非同期受信: シングル・コンシューマ・キューから	PL/SQL コールバック・プロシージャを定義する。 次のものを使用して登録する。 DBMS_AQ. REGISTER	OCISubscription Register queue_name をサブスクリプション名として指定する。 OCISubscription Enable	<AQXmlRegister> <destination>にキュー名を指定し、次のものに通知メカニズムを指定する。 <notify_url>	キューに対して QueueReceiver を作成した後、次のものを指定する。 QueueReceiver.set MessageListener
メッセージの非同期受信: マルチ・コンシューマ・キュー/トピックから	PL/SQL コールバック・プロシージャを定義する。 次のものを使用して登録する。 DBMS_AQ. REGISTER	OCISubscription Register queue:OCI_ATTR_CONSUMER_NAME をサブスクリプション名として指定する。 OCISubscription Enable	<AQXmlRegister> <destination>にキュー名、 <consumer_name>にコンシューマ、 <notify_url>に通知メカニズムを指定する	トピックに対して TopicSubscriber または TopicReceiver を作成した後、次のものを指定する。 TopicSubscriber.set MessageListener
メッセージのリスニング: 複数キュー/トピック	-	-	-	-
メッセージのリスニング: 1つ(または多数)のシングル・コンシューマ・キュー	DBMS_AQ. LISTEN agent_list のすべてのエージェントの agent_name に NULL を使用する。	OCIAQListen agent_list のすべてのエージェントの agent_name に NULL を使用する。	サポートされません	1つの QueueSession に複数の QueueReceivers を作成した後、次のものを指定する。 QueueSession.set MessageListener
メッセージのリスニング: 1つ(または多数)のマルチ・コンシューマ・キュー/トピック	DBMS_AQ. LISTEN agent_list のすべてのエージェントに agent_name を指定する。	OCIAQListen agent_list のすべてのエージェントに agent_name を指定する。	サポートされません	1つの TopicSession に複数の TopicSubscribers または TopicReceivers を作成した後、次のものを指定する。 TopicSession.set MessageListener

OCCIを使用したOracle Database Advanced Queuingへのアクセス

C++アプリケーションはOCCIを使用できますが、これには、メッセージ・クライアントによるOracle Database Advanced Queuingへのアクセスを可能にするOracle Database Advanced Queuingインタフェースのセットが含まれます。

OCCI AQでは、メッセージが使用可能なデータベースでのメッセージの送受信、およびパブリッシュ/サブスクライブに必要な操作関数をすべてサポートしています。メッセージ選択ルールに基づいた同期および非同期メッセージの受入れが可能です。シャード・キューは、OCCIクライアントをサポートしていません。

関連項目:

[『Oracle C++ Call Interfaceプログラマーズ・ガイド』](#)のOracle Database Advanced Queuingに関する項

Oracle Java Message Service (Oracle JMS)を使用したOracle Database Advanced Queuingへのアクセス

Java Message Service(JMS)は、Sun社、オラクル社、IBM社およびその他のベンダーによって定められたメッセージングの標準規格です。JMSは、JMSクライアントが企業のメッセージ関連製品の機能にアクセスする方法を定義する、インタフェースおよび対応するセマンティクスの集合です。Oracle Java Message Service (Oracle JMS)には、JMS標準に基づいた、Oracle Database Advanced Queuing用のJava APIが用意されています。

[Oracle Java Message Service](#) (Oracle JMS)は、標準のJMSインタフェースをサポートしている他、標準以外の管理操作や他の機能をサポートする拡張機能を備えています。

標準[Java Message Service](#) (JMS)機能には次のような機能があります。

- キューを使用したPoint-to-Point通信モデル
- トピックを使用したパブリッシュ・サブスクライブ通信モデル
- `ObjectMessage`、`StreamMessage`、`TextMessage`、`BytesMessage`、`MapMessage`の各メッセージ型
- 非同期および同期のメッセージ配信
- メッセージ・ヘッダー・フィールドまたはプロパティに基づくメッセージ選択

Oracle JMSの拡張機能は、次のとおりです。

- キュー表、キューおよびトピック作成用の管理API
- トピックの受信者リストを使用したPoint-to-Multipoint通信
- 宛先間でのメッセージの伝播。アプリケーションでのリモート・サブスクライバの定義が可能になります。
- トランザクション・セッションのサポート。1つのトランザクションでJMS操作およびSQL操作が可能になります。
- メッセージがデキューされた後のメッセージ保存
- メッセージ遅延。一定の遅延の後にメッセージが表示されます。
- 例外処理。メッセージを正常に処理できない場合、メッセージが例外キューに移動されます。
- `AdtMessage`のサポート

これらはOracleオブジェクトとしてデータベースに格納されるため、メッセージのペイロードをエンキュー後に問い合わせることができます。サブスクリプションは、メッセージ・プロパティのみでなく、これらのメッセージの内容にも定義できます。

- トピックのブラウズ

これにより、永続サブスクライバがメッセージをパブリッシュ・サブスクライブ(トピック)先でブラウズできます。オプションで、これらのサブスクライバがブラウズ済メッセージをパージして、そのサブスクライバについてはメッセージがOracle Database

Advanced Queuingによって保持されないようにできます。

関連項目:

- 『Java Message Service Specification』、バージョン1.1、2002年3月18日、Sun Microsystems, Inc.
- 『[Oracle Database Advanced Queuing Java APIリファレンス](#)』

標準およびOracle JMSアプリケーションへのアクセス

標準JMSインタフェースはjavax.jmsパッケージにあります。Oracle JMSインタフェースはoracle.jmsパッケージにあります。Oracle JMSインタフェースを使用するには、DBMS_AQINおよびDBMS_AQJMSパッケージに対するEXECUTE権限が必要です。これらの権限は、AQ_USER_ROLEまたはAQ_ADMINISTRATOR_ROLEを使用しても取得できます。メッセージの送信または受信を行うには、適切なシステムおよびキュー権限またはトピック権限が必要です。

Oracle JMSは、[Java Database Connectivity](#) (JDBC)を使用してデータベースに接続するため、データベース外ではJDBC OCIドライバまたはJDBC Thinドライバを使用してOracle JMSアプリケーションを実行できます。

JDBC OCIドライバまたはJDBC Thinドライバの使用

データベース外で実行されるクライアントでJMSを使用するには、CLASSPATHに適切な[JDBCドライバ](#)、[Java Naming and Directory Interface](#) (JNDI)のjarファイルおよびOracle Database Advanced Queuingのjarファイルを指定する必要があります。

Oracle Databaseは、JDK 1.2、JDK 1.3、JDK 1.4、JDK 5.nおよびすべてのclasses12*.jarファイルをサポートしていないことに注意してください。JDK 6.nおよびJDK 7.nでそれぞれojdbc6.jarファイルおよびojdbc7.jarファイルを使用する必要があります。次のjarファイルとzipファイルは使用しているJDKのリリースに基づくCLASSPATHにある必要があります。

JDK 1.5.xの場合、CLASSPATHに次のパスを含める必要があります。

```
ORACLE_HOME/jdbc/lib/ojdbc6.jar
```

JDK 1.6.xの場合、CLASSPATHに次のパスを含める必要があります。

```
ORACLE_HOME/jdbc/lib/ojdbc7.jar
```

いずれかのJDKバージョンに次のファイルを使用します。

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

JServerでのOracleサーバー・ドライバの使用

アプリケーションがJServer内で実行される場合、JServerのインストール時に自動的にロードされたOracle JMSクラスにアクセスする必要があります。これらのクラスが使用できない場合は、SQLスクリプト\$ORACLE_HOME/rdbms/admin/initjmsを使用して、jmscommon.jarの後にaqapi.jarをロードする必要があります。

Oracle Database Advanced Queuing XMLサーブレットを使用した Oracle Database Advanced Queuingへのアクセス

Oracle Database Advanced Queuing XMLサーブレットを使用して、HTTPでOracle Database Advanced Queuingにアクセスできますが、これには、Internet Data Access Presentation (IDAP)と呼ばれるSimple Object Access Protocol (SOAP)およびOracle Database Advanced Queuing XMLメッセージ・フォーマットが使用されます。

Oracle Database Advanced Queuingサーブレットを使用すると、クライアントは次のアクションを実行できます。

- シングル・コンシューマ・キューへのメッセージの送信
- マルチ・コンシューマ・キュー/トピックへのメッセージのパブリッシュ
- キューからのメッセージの受信
- メッセージ通知の受信登録

4 Oracle Database Advanced Queuingの管理

次のトピックでは、Oracle Database Advanced Queuingを管理する方法を説明します。

- [Oracle Database Advanced Queuing互換性パラメータ](#)
- [キューのセキュリティとアクセス制御](#)
- [キュー表のエクスポート/インポート](#)
- [Oracle Enterprise Managerのサポート](#)
- [Oracle Database Advanced QueuingとXAの併用](#)
- [キュー管理の制限事項](#)
- [伝播の管理](#)

Oracle Database Advanced Queuing互換性パラメータ

バッファ済メッセージを格納するキューは、互換性を8.1以上に設定して作成する必要があります。

次の機能を使用するには、init.oraのcompatibleパラメータおよび[キュー表](#)のcompatibleパラメータを8.1以上に設定する必要があります。

- キュー・レベルのアクセス制御
- Oracle Real Application Clusters環境のサポート
- [パブリッシュ・サブスクライブ](#)用のルールベースのサブスクライバ
- 非同期通知
- 送信元の識別
- 履歴管理情報の記憶域の分離
- 保護キュー

データベースの互換性が10.0である場合のみ、大文字と小文字が混在するキュー名、キュー表名およびサブスクライバ名を指定できますが、名前を二重引用符で囲む必要があります。したがって、abc.efgは、スキーマがABCで名前がEFGであることを示しますが、“abc”.“efg”はスキーマがabcで名前がefgであることを示します。

キューのセキュリティとアクセス制御

次のトピックでは、Oracle Database Advanced Queuingのキューのセキュリティおよびアクセス制御について説明します。

- [Oracle Database Advanced Queuingのセキュリティ](#)
- [キューのセキュリティ](#)
- [キュー権限とアクセス制御](#)
- [OCIアプリケーションとキュー・アクセス](#)
- [伝播に必要なセキュリティ](#)

Oracle Database Advanced Queuingのセキュリティ

構成情報は、DBMS_AQADMパッケージ内のプロシージャを使用して管理できます。

最初は、SYSおよびSYSTEMのみに、DBMS_AQADMおよびDBMS_AQ内のプロシージャの実行権限が付与されています。この2つのパッケージのEXECUTE権限が付与されたユーザーは、自分のスキーマで、キューの作成、管理および使用ができます。他のスキーマのキューを作成および管理する場合は、The MANAGE_ANY AQシステム権限が使用されます。

[Java Message Service\(JMS\)API](#)のユーザーには、DBMS_AQJMSおよびDBMS_AQINに対するEXECUTE権限が必要です。

トピック:

- [管理者ロール](#)
- [ユーザー・ロール](#)
- [Oracle Database Advanced Queuingオブジェクト型へのアクセス](#)

関連項目:

AQシステム権限の詳細は、[「Oracle Database Advanced Queuingシステム権限の付与」](#)を参照してください。

管理者ロール

AQ_ADMINISTRATOR_ROLEには、キュー管理に必要なすべての権限が付与されます。

ロールに付与された権限によって、権限受領者は次のことができるようになります。

- データベースのあらゆる[スキーマ](#)に対するすべての[キュー](#)管理操作(キューやキュー表の作成など)の実行
- データベースのすべてのキューに対する[エンキュー](#)および[デキュー](#)操作の実行
- そのキューの作業負荷を監視するための統計ビューへのアクセス
- DBMS_TRANSFORMを使用した変換の作成
- DBMS_AQELM内のあらゆるプロシージャの実行
- DBMS_AQJMS内のあらゆるプロシージャの実行

注意:

- ユーザーはAQ_ADMINISTRATOR_ROLEがなくても、自分のスキーマでキューを作成したり、他のAQ DDLを実行できます。同様に、DBMS_AQADMパッケージの実行権限のみがユーザーに必要です。
- AQ_ADMINISTRATOR_ROLEでは、SYSまたはSYSTEMスキーマにAQオブジェクトを作成できません。そのため、任意のスキーマでキュー表やキューを作成できるという前述の説明に、SYSおよびSYSTEMスキーマは含まれません。
- AQ_ADMINISTRATOR_ROLEでは、SYSまたはSYSTEMスキーマのキューでエンキューやデキューはできません。



ユーザー・ロール

AQ_USER_ROLEでは、エンキューまたはデキューを行うには権限が不十分なため、このロールを付与することは避けてください。

ユーザーにこのレベルの制御を付与する場合は、データベース管理者がDBMS_AQADM.GRANT_SYSTEM_PRIVILEGEおよびDBMS_AQADM.REVOKE_SYSTEM_PRIVILEGEを直接データベース・ユーザーに対して実行し、システム権限ENQUEUE_ANYおよびDEQUEUE_ANYを付与するオプションがあります。

アプリケーション開発者がキューに権限を付与するには、DBMS_AQADM.GRANT_QUEUE_PRIVILEGEおよびDBMS_AQADM.REVOKE_QUEUE_PRIVILEGEを実行して、オブジェクト・レベルの権限を付与および取り消します。

データベース・ユーザーは、DBMS_AQに対するEXECUTE権限以外に、オブジェクト・レベルまたはシステム・レベルの明示的な権限を付与されなくても、自分のスキーマでのキューのエンキューおよびデキューが可能です。

注意:

- クライアントが OCI または PL/SQL の場合に、自分のスキーマでエンキューまたはデキューできるようにするには、DBMS_AQ パッケージの EXECUTE 権限のみがユーザーに必要になります。



クライアントが JDBC または JMS の場合は、これに加えて、DBMS_AQIN パッケージの実行権限も必要になります。

- 自分のスキーマ内のキューでエンキューまたはデキューする場合、AQ_USER_ROLE はユーザーに必要ありません。

Oracle Database Advanced Queuingオブジェクト型へのアクセス

Oracle Database Advanced Queuingの内部オブジェクトは、すべてPUBLICからアクセスできます。

キューのセキュリティ

Oracle DatabaseのOracle Database Advanced Queuing管理者は、キューを作成できます。キューを作成するとき、DBMS_AQADM.CREATE_QUEUE_TABLEのcompatibleパラメータのデフォルト値は、compatibleパラメータのデフォルト値です。

エンキューまたはデキューするユーザーには、DBMS_AQのEXECUTE権限の他、ターゲット・キューでのエンキューまたはデキュー権限、もしくはENQUEUE_ANY/DEQUEUE_ANYシステム権限のいずれかが必要です。

キュー権限とアクセス制御

キューに対するオブジェクト・レベルの権限は、付与または取消しが可能です。システム・レベルの各種権限を付与または取り消すこともできます。

[表4-1](#)に、Oracle Database Advanced Queuingのすべての共通操作とそれらの操作を実行する場合に必要な権限を示します。

表4-1 操作および必要な権限

操作	必要な権限
所有するキューに対する	DBMS_AQADM の EXECUTE 権限が必要です。その他の権限は不要です。

操作	必要な権限
CREATE/DROP/MONITOR	
すべてのキューに対する CREATE/DROP/MONITOR	DBMS_AQADM の EXECUTE 権限の他、AQ_ADMINISTRATOR_ROLE が付与されている別のユーザーから、このロールを取得する必要があります(最初は、SYS および SYSTEM から AQ_ADMINISTRATOR_ROLE が付与されます)
所有するキューに対する ENQUEUE/ DEQUEUE	DBMS_AQ の EXECUTE 権限が必要です。その他の権限は不要です。
他のユーザーが所有するキューに対する ENQUEUE/ DEQUEUE	DBMS_AQ の EXECUTE 権限の他、DBMS_AQADM.GRANT_QUEUE_PRIVILEGE を使用して、所有者から権限を取得する必要があります。
すべてのキューに対する ENQUEUE/ DEQUEUE	DBMS_AQ の EXECUTE 権限の他、DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE を使用して、Oracle Database Advanced Queuing 管理者から、ENQUEUE ANY QUEUE または DEQUEUE ANY QUEUE システム権限を取得する必要があります。

OCIアプリケーションとキュー・アクセス

Oracle Call Interface(OCI)アプリケーションでキューにアクセスするには、セッション・ユーザーに、アクセス先のキューのオブジェクト権限、またはシステム権限(ENQUEUE ANY QUEUEまたはDEQUEUE ANY QUEUE)のいずれかを付与する必要があります。

DBMS_AQのEXECUTE権限とセッション・ユーザーの権限が照合されて確認されることはありません。

伝播に必要なセキュリティ

Oracle Database Advanced Queuingでは、データベース・リンクを介してメッセージを伝播します。

[伝播](#)ドライバは、ソース・キューの所有者としてソース・キューからデキューするため、ソース・キューに対して明示的なアクセス権を付与する必要はありません。宛先では、データベース・リンクのログイン・ユーザーには、ENQUEUE ANY QUEUE権限を付与するか、または宛先キューに対するエンキュー権限を付与する必要があります。ただし、データベース・リンクのログイン・ユーザーが宛先のキュー表も所有している場合は、明示的なOracle Database Advanced Queuingの権限を付与しないでください。

関連項目:

[オブジェクト・キューからの伝播](#)

Oracle RACの非シャード・バッファ済メッセージに必要なセキュリティ

内部的には、Oracle RAC上のバッファ済キューはインスタンス間でdblinksを使用することがあります。Oracle RACのバッファ済キューにエンキューまたはデキューする定義者権限のパッケージでは、パッケージのユーザーにINHERIT REMOTE PRIVILEGESを付与する必要があります。

キュー表のエクスポート/インポート

キュー表がエクスポートされると、キュー表データおよびPL/SQLコードの無名ブロックがエクスポート・ダンプ・ファイルに書き込まれます。キュー表がインポートされると、インポート・ユーティリティがこれらのPL/SQL無名ブロックを実行して、メタデータをデータ・ディクショナリに書き込みます。

Oracle AQでは、ユーザーのエクスポートとともに登録をエクスポートしません。クライアント登録を利用するすべてのアプリケーションでは、インポートされたデータベースにクライアントが存在しないことを考慮する必要があります。

注意:

バッファ済メッセージをエクスポートまたはインポートすることはできません。



データベースのスキーマが同じで、かつ同名のキュー表がエクスポート・ダンプに存在する場合は、データベース・キューを空にしてからキュー表とキューをインポートしてください。そうしないと、インポートしたキューのメタデータが破損する可能性があります。

トピック:

- [キュー表データのエクスポート](#)
- [キュー表データのインポート](#)
- [データ・ポンプ・エクスポートとデータ・ポンプ・インポート](#)

キュー表データのエクスポート

キューをエクスポートすると、必然的に基になるキュー表および関連するディクショナリ表もエクスポートされます。キューは、キュー表単位でもエクスポートできます。

複数の受信者がいるキュー表のエクスポート

非シャード・キューの場合、複数の受信者をサポートするキュー表は、次の表と対応付けられています。

- デキュー [索引構成表](#)(IOT)
- 時間管理索引構成表
- サブスクライバ表
- 履歴IOT

シャード・キューは、次のオブジェクトに関連付けられます。

- キュー表
- デキュー表
- 時間管理表
- オプションの例外キュー・マップ表
- 前述の表の索引
- 順序

- ルール・セットおよび評価コンテキスト

全データベース・モード、ユーザー・モードおよび表モード・エクスポートで、これらの表は自動的にエクスポートされます。[「エクスポート・モード」](#)を参照してください。

メタデータ表には、キュー表の一部の表のROWIDが含まれているため、インポート処理の際、メタデータ表のインポート時に使用不可になったROWIDの記録が生成されます。インポート処理の一部として、使用不可になったROWIDはキュー・システムによって自動的に修正されるため、このメッセージは無視して構いません。ただし、インポート時に別の問題が発生した場合(ロールバック・セグメント領域の不足など)は、ユーザーが問題を修正してインポートを繰り返す必要があります。

エクスポート・モード

エクスポートの操作モードには、全データベース・モード、ユーザー・モードおよび表モードがあります。キュー表の増分エクスポートはサポートされていません。

全データベース・モードでは、キュー表、すべての関連表、システム・レベルの権限付与およびプライマリ・オブジェクトとセカンダリ・オブジェクトの権限付与が自動的にエクスポートされます。

ユーザー・モードでは、キュー表、すべての関連表およびプライマリ・オブジェクトの権限付与が自動的にエクスポートされます。ただし、FROMUSER TOUSER句を使用したあるスキーマから別のスキーマへのユーザー・レベルのエクスポートはサポートされていません。

表モードでは、キュー表、すべての関連表およびプライマリ・オブジェクトの権限付与が自動的にエクスポートされます。たとえば、非シャード・マルチ・コンシューマ・キュー表をエクスポートする場合、次の表が自動的にエクスポートされます。

- AQ\$_queue_table_I(デキューIOT)
- AQ\$_queue_table_T (時間管理IOT)
- AQ\$_queue_table_S(サブスクライバ表)
- AQ\$_queue_table_H(履歴IOT)

シャード・キューの場合、次の表が自動的にエクスポートされます。

- queue_table
- AQ\$_queue_name_L (デキュー表)
- AQ\$_queue_name_T (時間管理表)
- AQ\$_queue_name (例外マップ表)
- AQ\$_queue_name_V (評価コンテキスト)
- queue_name_R (ルール・セット)

キュー表データのインポート

エクスポートと同様に、キューをインポートすると、必ず基になるキュー表および関連するディクショナリ・データもインポートされます。キュー表データがインポートされると、インポート・ユーティリティがダンプ・ファイルのPL/SQL無名ブロックを実行して、メタデータをデータ・ディクショナリに書き込みます。

複数の受信者がいるキュー表のインポート

複数の受信者をサポートする非シャード・キュー表は、次の表と対応付けられています。

- デキューIOT
- 時間管理IOT

- サブスクライバ表
- 履歴IOT

シャード・キューは、次のオブジェクトに関連付けられます。

- キュー表
- デキュー・ログ表
- 時間管理表
- オプションの例外キュー・マップ表
- 前述の表の索引
- 順序
- ルール・セットおよび評価コンテキスト

これらのオブジェクトは、キュー表自体とともにインポートする必要があります。

インポートIGNOREパラメータ

すでにデータが含まれているキュー表には、キュー・データをインポートしないでください。キュー表をインポートする場合は、インポート・ユーティリティのIGNOREパラメータを必ずNOに設定する必要があります。IGNOREパラメータがYESに設定されており、すでに存在しているキュー表がダンプ・ファイル中の表定義と互換性がある場合は、ダンプ・ファイルの各行がそのキュー表にロードされます。それと同時に、古いキュー表定義が失われ、再作成されます。インポート前のキュー表定義は失われ、複製された行がキュー表に現れます。

データ・ポンプ・エクスポートとデータ・ポンプ・インポート

データ・ポンプでは、キュー表の置換モードとスキップ・モードがサポートされています。

置換モードでは、既存のキュー表が削除され、エクスポート・ダンプ・ファイルの新規キュー表に置き換えられます。スキップ・モードでは、既存のキュー表はインポートされません。

キュー表のTRUNCATEモードおよびAPPENDモードはサポートされていません。この場合の動作は置換モードと同様です。

関連項目:

データ・ポンプ・エクスポートおよびデータ・ポンプ・インポートの詳細は、[『Oracle Databaseユーティリティ』](#)を参照してください。

Oracle Enterprise Managerでの非シャード・キューのサポート

Oracle Enterprise Managerでは、Oracle Database Advanced Queuingの大抵の管理機能がサポートされています。Oracle Database Advanced Queuingの機能は、Enterprise Managerコンソールのナビゲーション・ツリーの分散ノードに配置されています。

Oracle Enterprise Managerで使用可能な機能は次のとおりです。

- スキーマ・マネージャに含まれるキュー項目を使用した各種プロパティの参照
- キューの作成、開始、停止および削除
- 伝播のスケジューリングおよびスケジュール解除

- サブスクライバの追加および削除
- データベースのすべてのキューに対する伝播スケジュールの表示
- データベースのすべてのキューに対するエラーの表示
- メッセージ・キューの表示
- 権限の付与および取消し
- 変換の作成、変更または削除

Oracle Database Advanced QueuingとXAの併用

Oracle Database Advanced Queuing OCIインタフェースを使用する場合は、`xa_open`文字列で「Objects=T」を指定する必要があります。これは、オブジェクト・モードで、XAにクライアント側キャッシュを初期化させます。OCIまたはPro*Cから、PL/SQLラッパーを介してOracle Database Advanced Queuingを使用する場合、この処理は必要ありません。

Pro*Cドキュメントの[ラージ・オブジェクト](#)(LOB)メモリ管理の概念は、Oracle Database Advanced Queuing RAWメッセージと関連性はありませんが、これはOracle Database Advanced Queuingが単純なRAWバッファ抽象を提供するためです(ただし、これらはLOBとして格納されます)。

サービス間(`xa_start`境界と`xa_end`境界の間など)でデキュー処理を続ける場合に、Oracle Database Advanced Queuingナビゲーション・オプションを使用するときは、FIRST_MESSAGEを使用してデキュー位置をリセットする必要があります。これは、`xa_end`が終了すると、XAがカーソル・フェッチ状態を取り消すためです。リセットしないと、ナビゲーション・オプションの指定順序が正しくないというエラー・メッセージ(ORA-25237)が表示されます。

関連項目:

- XAの詳細は、[『Oracle Database開発ガイド』](#)のOracle XAを使用したトランザクション・モニターの処理に関する項を参照してください
- [『Pro*C/C++プログラマーズ・ガイド』](#)のLarge Objects(LOB)に関する項を参照してください。

キュー管理の制限事項

次のトピックでは、キュー管理の制限事項について説明します。

- [サブスクライバ](#)
- [キュー表または関連IOTに対するDMLの未サポート](#)
- [REFペイロード属性を持つオブジェクト・キューからの伝播](#)
- [メッセージ・ペイロード\(実際に通信される情報\)内のコレクション型](#)
- [キュー表およびキューにおけるシノニム](#)
- [オブジェクト型におけるシノニム](#)
- [表領域のPoint-in-Timeリカバリ](#)
- [仮想プライベート・データベース](#)

注意:



データベースの互換性が 10.0 である場合のみ、大文字と小文字が混在するキュー名、キュー表名およびサブスクライバ名を指定できますが、名前を二重引用符で囲む必要があります。したがって、abc. efg は、スキーマが ABC で名前が EFG であることを示しますが、“abc”. “efg”はスキーマが abc で名前が efg であることを示します。

サブスクライバ

各キューのローカル・サブスクライバ数は1,000以内です。

また、各リモートの接続先のデータベースで対応可能なリモート・サブスクライバ数は32です。

キュー表または関連IOTに対するDMLの未サポート

Oracle Database Advanced Queuingでは、キュー表または関連索引構成表(IOT)に対するデータ操作言語(DML)はサポートされていません。

キュー表を修正するためにサポートされている唯一の手段は、提供されているAPIを使用することです。キュー表やIOTは、DML ([データ操作言語](#))操作を行うと一貫性がなくなり、実質的に破損する可能性があります。

REFパイロード属性を持つオブジェクト・キューからの伝播

Oracle Database Advanced Queuingでは、パイロードにREF属性を持つオブジェクト・キューからの伝播はサポートされていません。

メッセージ・パイロード(実際に通信される情報)内のコレクション型

オブジェクトに含まれていないVARRAYを使用して、メッセージ・パイロードを組み立てることはできません。

また、現時点では、[メッセージ・パイロード](#)内で埋込みオブジェクトとしてネストした表を使用できません。ただし、1つ以上の[VARRAY](#)を含む[オブジェクト型](#)を作成し、これを基礎にしてキュー表を作成できます([例4-1](#)を参照してください)。

例4-1 VARRAYを含むオブジェクトの作成

```
CREATE TYPE number_varray AS VARRAY(32) OF NUMBER;  
CREATE TYPE embedded_varray AS OBJECT (col1 number_varray);  
EXECUTE DBMS_AQADM.CREATE_QUEUE_TABLE(  
  queue_table => 'QT',  
  queue_payload_type => 'embedded_varray');
```

キュー表およびキューにおけるシノニム

キューおよびキュー表におけるシノニムを解決するOracle Database Advanced Queuing PL/SQLコールはありません。

シノニムの作成は可能ですが、Oracle Database Advanced Queuingインタフェースに適用することは避けてください。

オブジェクト型におけるシノニム

オブジェクト型のシノニムを作成している場合、それをDBMS_AQADM.CREATE_QUEUE_TABLEで使用することはできません。エラーORA-24015が発生します。

表領域のPoint-in-Timeリカバリ

現在は、Oracle Database Advanced Queuingでは、表領域のPoint-in-Timeリカバリはサポートされていません。

キュー表が表領域に作成されると、Point-in-Timeリカバリは使用できません。Oracle Database Advanced Queuingは、通常のPoint-in-Timeリカバリをサポートしています。

仮想プライベート・データベース

Oracle Database Advanced Queuingのキュー表でセキュリティ・ポリシーを指定することで、仮想プライベート・データベースとともにOracle Database Advanced Queuingを使用できます。

デキュー中に、適用するポリシーのデキュー条件(deq_cond)または相関識別子を使用します。デキュー条件として「1=1」を使用できます。デキュー条件または相関識別子を使用しないとデキューでエラーが発生します。

注意:



デキュー条件または相関識別子を使用すると、メッセージのデキュー順序が不定になり、キューのソート順序は考慮されません。

伝播の管理

次のトピックでは、Oracle Database Advanced Queuingの伝播の管理について説明します。

- [伝播に必要なEXECUTE権限](#)
- [オブジェクト・キューからの伝播](#)
- [伝播の最適化](#)
- [伝播中の障害対策](#)

注意:



伝播が正常に動作するために、キューaq\$_prop_notify_Xが中止または削除されたり、表aq\$_prop_table_Xが削除されることがないようにしてください。

伝播に必要なEXECUTE権限

伝播ジョブの所有者はSYSですが、伝播はキュー表の所有者のセキュリティ・コンテキスト内で発生します。

以前は、伝播ジョブの所有者は、伝播をスケジューリングするユーザーで、伝播が発生するのは、伝播スケジュールを設定するユーザーのセキュリティ・コンテキスト内でした。キュー表の所有者には、DBMS_AQADMパッケージのEXECUTE権限が必要です。権限がない場合、Oracleデータベース・スナップショット・プロセスは、エラー識別子SYS.DBMS_AQADMが定義されていないトレース・ファイルを伝播および生成しません。伝播には、キュー表の所有者が所有するプライベート・データベース・リンクを使用できます。接続文字列に指定されたユーザー名には、リモート・データベース上のDBMS_AQパッケージおよびDBMS_AQADMパッケージに対するEXECUTEアクセス権限が必要です。

オブジェクト・キューからの伝播

BFILEROBJを伴うオブジェクト・キューからの伝播がサポートされています。

BFILEROBJを使用したオブジェクト・キューの伝播を可能にするには、ソース・キューの所有者が、BFILEROBJの格納ディレクトリに対応するディレクトリ・オブジェクトの読取り権限を取得する必要があります。データベース・リンクのユーザーは、接続先データベースで、BFILEROBJのディレクトリに対応するディレクトリ・オブジェクトの書き込み権限を取得する必要があります。

AQの伝播では、NOT FINAL型はサポートされません。データベース・リンクを指定せずに、オブジェクト・キューからBFILEROBJを伝播することはできません。

関連項目:

ディレクトリ・オブジェクトの詳細は、『[Oracle Database SQL言語リファレンス](#)』のディレクトリの作成に関する項を参照してください。

伝播の最適化

AQ伝播ジョブはOracle Schedulerで実行されます。伝播は、次の方法でスケジュールできます。

- 専用のスケジュールでは、伝播は永久的にまたは指定された期間で実行されます。このモードは、最も低い伝播の待ち時間を提供します。
- 周期的なスケジュールでは、伝播は指定された間隔で周期的に実行されます。これは、伝播をバッチ・モードで実行する場合に使用します。
- イベント・ベース・システムでは、伝播するメッセージがある場合に伝播が開始されます。このモードでは、迅速なレスポンス時間を持続しながら、使用可能なリソースを効果的に使用できます。

管理者は、アプリケーション・パフォーマンス要件に最も適したスケジュールを選択します。

Oracle Schedulerは、伝播スケジュールが必要とする数のジョブ・キュー・プロセスを開始します。スケジューラはスループットで最適化するため、システムの負荷が大きい場合、いくつかの伝播ジョブを実行しないことがあります。リソース・マネージャを使用すると、スケジュール決定の制御が向上します。特に、伝播ジョブを異なるリソース・グループに関連付けると、適切なスケジュールが可能になり、負荷が大きい状況に役に立ちます。

JOB_QUEUE_PROCESSES数を設定するとき、DBAは、この値が、伝播するメッセージの伝播元のキュー数、およびメッセージの伝播先の宛先(キューではない)の数によって決まることを理解しておく必要があります。

伝播は、スケジューリング・アルゴリズムによって処理されます。このアルゴリズムは使用可能なジョブ・キュー・プロセスを最適化し、メッセージがソース・キューにエンキューされてから宛先に現れるまでの時間を最小限にするため、[OLTP](#)とほぼ同様の動作を実現します。アルゴリズムが操作できるスケジュールの数は無制限で、様々な障害に対しても対処できます。伝播は、使用可能なジョブ・キュー・プロセスの最適使用を試みますが、開始するジョブ・キュー・プロセスの数はレプリケーション・ジョブのような伝播に関連しないジョブの存在にも依存します。したがって、このスケジューリング・アルゴリズムによって最高の結果を得るためには、次のガイドラインを利用することが重要です。

スケジューリング・アルゴリズムは、次のようにジョブ・キュー・プロセスを使用します(ここでは、アクティブ・スケジュールが適切に設定されているとします)。

- アクティブ・スケジュール数がジョブ・キュー・プロセスの半数に満たない場合、アクティブ・スケジュール数と同数のジョブ・キュー・プロセスが取得されます。

- アクティブ・スケジュール数がジョブ・キュー・プロセス数の半数を超える場合、ジョブ・キュー・プロセスの半数を取得した後で、取得した各ジョブ・キュー・プロセスに複数のアクティブ・スケジュールを割り当てます。
- システムがオーバーロードになっている(すべてのスケジュールが伝播のためにビジーになっている)場合、可用性に応じて、ジョブ・キュー・プロセス総数が-1になるまで追加のジョブ・キュー・プロセスが取得されます。
- あるプロセスによって操作されるアクティブ・スケジュールがどれも伝播の必要なメッセージを持っていない場合、そのジョブ・キュー・プロセスは解放されます。
- このアルゴリズムによって、負荷が大きいプロセスから小さいプロセスにスケジュールを転送することで自動ロード・バランシングを行い、オーバーロードのプロセスがなくなるようにします。

伝播中の障害対策

スケジューリング・アルゴリズムは、様々な障害に対して強力に対処します。メッセージの伝播を妨げる一般的な障害は次のとおりです。

- データベース・リンクに失敗
- リモート・データベースが使用できない
- リモート・キューが存在しない
- リモート・キューが開始されていない
- メッセージをリモート・キューにエンキューしようとしたときにセキュリティ違反が発生した

前述のいずれかの状態が生じると、該当するエラー・メッセージがDBA_QUEUE_SCHEDULESビューにレポートされます。

スケジュールでエラーが発生した場合、そのスケジュール内のメッセージの伝播は、再試行間隔(失敗数に応じて異なる)の後に再度試行されます。再試行の回数がシステムで定義された最大数を超えると、そのスケジュールは無効になります。

エラーの原因になっていた問題が解決し、スケジュールが有効化されても、最終エラーの発生した日付、時刻およびメッセージを示すエラー・フィールドに、そのエラー情報が残ります。このフィールドがリセットされるのは、そのスケジュールによってメッセージが正常に伝播されたときです。

関連項目:

[Oracle Database Advanced Queuingのトラブルシューティング](#)

5 Oracle Database Advanced Queuingのパフォーマンスおよび拡張性

次のトピックでは、Oracle Database Advanced Queuing (AQ)のパフォーマンスおよび拡張性に関する事柄について説明します。

- [シャード・キュー](#)
- [非シャード・キュー](#)
- [パフォーマンス・ビュー](#)

シャード・キュー

[シャード・キュー](#)では、異なるエンキュー・セッションからのメッセージを並列でデキューできるため、エンキュー・デキューのスループットが特にOracle RACインスタンス間で向上します。キューの各シャードは、セッション内のエンキュー時刻に基づいて順序付けされ、シャード間の順序付けはベストエフォートです。シャード・キューは、エンキュー元およびデキュー元がそれら同士で競合しないように、表のパーティションを自動的に管理します。また、シャード・キューは、インメモリーのメッセージ・キャッシュを使用して、パフォーマンスを最適化し、エンキューおよびデキューのディスクおよびCPUのオーバーヘッドを削減します。

シャード・キューの利点およびトレードオフは、次のとおりです。

- 特に各サブスクリバが各インスタンスに複数のデキュー元を持つ場合に、シャード・キューにより、Oracle Real Application Clusters (Oracle RAC)上の単一のキューの拡張性が向上します。
- シャード・キューは、増加したメモリー使用量をトレードオフしてパフォーマンスを向上します。

この項では、次の項目について説明します。

- [シャード・キューおよびメッセージ・キャッシュ](#)
- [シャード・キューおよびエンキュー/デキュー・メッセージ](#)
- [シャード・キューおよびネイティブJMSサポート](#)
- [シャード・キューおよびパーティション化](#)
- [シャード・キューおよびOracle Real Application Clusters \(Oracle RAC\)](#)
- [シャード・キューの制限事項](#)
- [シャード・キューのチューニング](#)

シャード・キューおよびメッセージ・キャッシュ

シャード・キューは、専用のメッセージ・キャッシュを導入しており、これを使用すると、SGAの使用量をトレードオフでき、スループットの向上、待ち時間の削減、同時処理の向上を図れます。パーティション化と組み合わせると、メッセージ・キャッシュで、一部の問合せ、DML操作および索引に対する必要性が低くなります。メッセージ・キャッシュは、デキュー元の処理がエンキュー元に追いついている場合、およびメッセージ・キャッシュが、各シャード・キューのエンキュー元およびデキュー元に対して、メッセージ(ペイロードを含む)を十分に格納できる大きさである場合に最も効果的です。メッセージ・キャッシュでは、Streamsプールが使用されます。シャード・キューが、Streamsのレプリケーション機能と同じインスタンスでStreamsプールを共有する場合、SET_MIN_STREAMS_POOLおよびSET_MAX_STREAMS_POOLなどのDBMS_AQADMプロシージャを使用して、Streamsプールのメモリー割当てを微調整できます。

関連項目:

詳細は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)を参照してください。

シャード・キューおよびエンキュー/デキュー・メッセージ

スループットを向上させ、オーバーヘッドおよび待ち時間を削減するために、エンキューおよびデキューは、メッセージ・キャッシュ、ルール・エンジン、および可能な場合はバックグラウンド処理を使用するように最適化されています。次に例を示します。

- シャード・キューは、新しいルール・エンジンの改善点を利用します。
- メッセージ・キャッシュ内にペイロードのあるメッセージは、デキュー中にディスクから再読取りする必要がありません。
- 相関IDによるデキューまたはその他のJMSプロパティは、ディスクにアクセスすることなく頻繁に評価できます。
- シャード・キューにおけるパーティション操作により、効率的な一括処理が実装されます。

シャード・キューおよびネイティブJMSサポート

シャード・キューは、次のネイティブ・サポートを有しています。

- 非永続サブスクライバ
- JMSペイロード
- 優先順位

シャード・キューでは、永続メッセージと非永続メッセージの両方がサポートされます。非永続メッセージはメッセージ・キャッシュ内のメモリーに格納され、ディスクには格納されません。そのため、非永続メッセージは、インスタンスのクラッシュ時や停止時に失われます。

シャード・キューは、JMS要件を満たす2種類のサブスクライバをネイティブにサポートしています。

- 非永続サブスクライバ: これらのサブスクライバは、サブスクライバがアクティブ中にメッセージがパブリッシュされた場合のみ、選択されたトピックについてのメッセージを受信します。このサブスクリプションは、異なるセッション間で共有できません。
- 永続サブスクライバ: これらのサブスクライバは、サブスクライバが非アクティブ中にパブリッシュされたメッセージを含めて、トピックについてパブリッシュされたすべてのメッセージを受信します。複数のデータベース・セッションで、同じサブスクリプションを共有できます。

シャード・キューは、ADTを使用してJMSペイロードを格納しません。JMSメッセージは、データベースのスカラー列に格納されます。TEXT、BYTES、MAP、STREAM、OBJECTなどのJMSメッセージ・タイプは、ペイロードのサイズとタイプに応じて、キュー表のTEXT/RAWまたはCLOB/BLOBのスカラー列にJMSペイロードを格納します。JMSメッセージ・プロパティは、ユーザー定義プロパティに属性アクセス機能が定義されているキュー表のCLOB (SecureFile)列に格納されます。ペイロードおよびユーザー・プロパティは、ADTとして格納されずに、RAW、VARCHAR2または保護ファイル列に格納されます。JMSヘッダー・プロパティおよびJMSプロバイダ情報は、それ自体のスカラー列に格納されます。

シャード・キューでは、0 (最も低い優先度)から9 (最も高い優先度)の範囲で整数の優先順位の値がサポートされており、JMS標準で定義されているように、デフォルトは4です。

シャード・キューおよびパーティション化

シャード・キューは、キュー表に使用される、基礎となるパーティション化された表を自動的に管理します。このようなパーティション管理は、フォアグラウンドまたはバックグラウンドで実行されます。各シャードは、エンキューされたメッセージのセッションレベルの順序付けを提供します。各エンキュー・セッションにはシャードが割り当てられます。各シャードは、一連のサブシャードで構成されます。各サブシャードは、1つのパーティションにマップされます。メッセージは、エンキュー時に表パーティションに自動的に割り当てられます。

新しいパーティションは、デキュー元がエンキュー元の処理に追いついておらず、キュー表を大きくする必要がある場合など、必要に応じて自動的に作成されます。パーティション内のすべてのメッセージがデキューされて必要がなくなった場合、パーティションは切り捨てられ再利用されます。メッセージ・キャッシュは、デキュー元による要求に応じて、メッセージをパーティションからメモリーに自動的にロードします。グローバル索引は、シャード・キューの基礎となる、パーティション化された表に作成できません。ローカル索引も、パーティション化された表では通常推奨されません。これらの索引が必要であるがパフォーマンスが低下する場合は、非シャード・キューの使用を検討してください。

シャード・キューおよびOracle Real Application Clusters (Oracle RAC)

シャード・キューは、可能な場合にはインスタンス間通信を回避しながら、自動的にエンキュー・セッションの順序付けを行います。インスタンス間通信が必要な場合もあります。たとえば、シャード・キューで、1つのOracleインスタンス上に単一のエンキュー・セッションがあり、別のインスタンスに単一のデキュー・セッションがある場合、シャード・キューはOracle RACインスタンス間でメッセージを転送します。パフォーマンスを向上させるため、メッセージの転送は、エンキュー・トランザクションに対してアトミックではありません。シャードにメッセージがないインスタンスに接続している場合、デキュー元でORA-25228が発生する場合があります。

ほとんどの場合、スループットを改善し、インスタンス間のオーバーヘッドを削減するために、各サブスクリバに対し複数のデキュー元を持つか、各Oracle RACインスタンスに単一のコンシューマ・キューを持つことを検討します。単一メッセージを指定するデキュー・セレクトを使用している場合は例外です。Oracle RACデータベースのメッセージ識別子により、共有キューからメッセージをデキューする場合、メッセージを含む共有キューのデキュー所有権を割り当てられたインスタンスに接続する必要があります。そうしないと、メッセージはデキュー・セッションへのデキューで使用できません。すべてのデキューが単一インスタンスで実行される場合、メッセージはこのインスタンスに自動的に転送されます。そのため、メッセージIDで広くデキューする共有単一コンシューマ・キューの場合は、共有キューのすべてのデキュー・セッションが単一インスタンスに接続することを検討してください。同様に、メッセージIDで広くデキューする共有複数コンシューマ・キューの場合は、各サブスクリバのすべてのデキュー・セッションが単一インスタンスに接続することを検討してください。サービスを使用してデキュー・セッションの特定のインスタンスへの接続を簡略化することができます。

シャード・キューの制限事項

現在、シャード・キューでは、次のOracle Databaseの機能はサポートされていません。

- メッセージの保存
- グループ化トランザクション
- サブスクリバ通知およびOCIコールバック通知の匿名での転送はサポートされません。PL/SQLコールバック通知はサポートされます。
- メッセージ・ゲートウェイ
- JMS伝播およびリモート・サブスクリバなど、JMSのOracle拡張機能
- 1つのキュー表に複数のキュー。シャード・キューは、CREATE_SHARDED_QUEUEインタフェースを介して作成されます。
- (JMS標準で指定されている)メッセージ優先順位、エンキュー時刻の順以外の順序付け。

- JDBC thick (OCI)ドライバ。
- シャード・キューおよび非シャード・キュー間の伝播
- メッセージ変換

シャード・キューのチューニング

シャード・キューは次の状態のときに最適に機能します。

- 各サブスクリバのデキュー元が各インスタンス内にあります。
- サブスクリバがエンキュー元の速度に追いついています。各Oracle RACインスタンス上の各サブスクリバに対して複数のデキュー元を設定することを検討してください。

メッセージ・キャッシュは、デキュー元の処理がエンキュー元に追いついている場合、およびキャッシュが、各シャード・キューのエンキュー元およびデキュー元に対して、メッセージ(ペイロードを含む)を十分に格納できる大きさである場合に最も効果的です。シャード・キューを使用する場合、次のいずれかを行う必要があります。

- パラメータSTREAMS_POOL_SIZEの設定

このパラメータは、Oracle Databaseでシャード・キュー・メッセージ・キャッシュに使用可能な共有メモリのサイズを制御します。指定されていない場合は、共有プール・サイズの最大10%までがStreamsプールに割り当てられます。

SGA_TARGET初期化パラメータに0以外の値を設定すると、Oracleの自動共有メモリ管理機能によってStreamsプールのサイズが管理されます。STREAMS_POOL_SIZE初期化パラメータにも0以外の値を設定した場合、この値は、Streamsプールの最小値として自動共有メモリ管理によって使用されます。

STREAMS_POOL_SIZE初期化パラメータに0以外の値を設定し、SGA_TARGETパラメータに0(ゼロ)を設定した場合、StreamsプールのサイズはSTREAMS_POOL_SIZEパラメータによって設定された値(バイト)です。

STREAMS_POOL_SIZEとSGA_TARGETの初期化パラメータがどちらも0に設定されている場合は、デフォルトにより、データベースで最初にStreamsプールを使用したときに、バッファ・キャッシュから共有プールの10%に相当する容量のメモリがStreamsプールに割り当てられます。

関連項目:

- Streams処理を使用したSTREAMS_POOL共有の詳細な制御は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_AQADM.set_min_streams_pool () および DBMS_AQADM.set_max_streams_pool () に関する項を参照してください。

- SGA自動チューニングの有効化

Streamsプールの使用量およびSGAを使用するその他のコンポーネントの使用量に基づいて、SGAからStreamsプールに適切な量のメモリが自動的に割り当てられます。その他のコンポーネントには、バッファ・キャッシュ、ライブラリ・キャッシュなどがあります。STREAMS_POOL_SIZEを指定した場合は、下限として使用されます。

- シャード・キューの手動チューニング

シャード・キューは、メッセージ・キャッシュにSTREAMS_POOLメモリーを割り当てることによってチューニングできます。

GV\$AQ_MESSAGE_CACHE_ADVICEビューは、現在のメッセージ・ロードのスナップショットに基づいて、STREAMS_POOLの割当量に関するアドバイスを提供します。高負荷である期間中に、列INST_ID、SIZE_FOR_ESTIMATEおよびESTD_SIZE_TYPEを選択します。ESTD_SIZE_TYPEは3つの値(MINIMUM、PREFERREDまたはMAXIMUM)のいずれかです。各ESTD_SIZE_TYPE値について複数のOracle RACインスタンスにおけるSIZE_FOR_ESTIMATEの最大値を見つけま

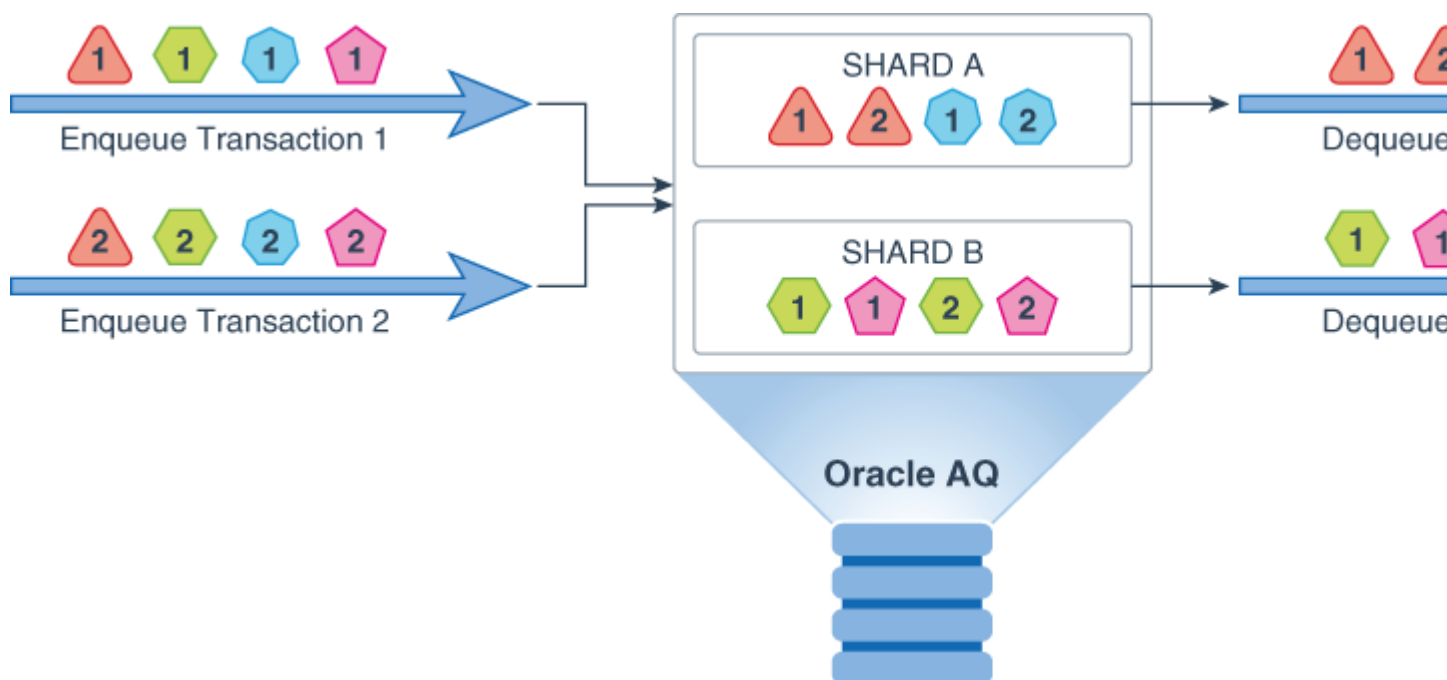
す。メッセージ・キャッシュのパフォーマンスをよくなるために、STREAMS_POOLに少なくともMINIMUM推奨値を設定することをお勧めします。STREAMS_POOLにMAXIMUM推奨値より大きい値を設定すると、パフォーマンスが少しよくなります。STREAMS_POOLにPREFERRED推奨値を設定すると、領域とパフォーマンスの適度なトレードオフが提供されるように試みられます。MAXIMUMサイズ推奨値がPREFERRED推奨値より大幅に大きい場合は、シャード・キューに孤立したサブスクライバがないこと、またはデキュー元がエンキュー・ロードに追いつくことができるように、さらにデキュー元をインスタンスに追加する必要があるかどうかを確認します。STREAMS_POOLのチューニングは、高負荷である期間に複数回行い、メッセージ・ロードの特性が変化したときにも行う必要があります。

ユーザー・シャーディング

アプリケーションは、シャード・キューのメッセージの共有方法を決定できます。このような場合、特定のシャードでメッセージをエンキューするように、アプリケーションで明示的に指定します。

たとえば、アプリケーションに、赤色、緑色、青色およびピンク色の異なるキーを持つ4種類のメッセージがあるとします。各エンキュー・セッションは、トランザクションでそれらのメッセージをエンキューできます。シャードAは、赤色および青色のメッセージを格納するように設定されています。シャードBは、緑色およびピンク色のメッセージを格納するように設定されています。また、各シャードには、シングル・コンシューマ・キューまたはJMSキューのためのアクティブなデキュー・セッションが1つのみ設定されています。同様に、各シャードには、マルチ・コンシューマ・キューまたはJMSトピックのためのデキュー・セッションがサブスクライバごとに1つのみ設定されています。そのデキュー・セッションは、デキュー元セッションの存続期間中はそのシャードに固定されたままになります。

次の例では、エンキュー・トランザクションが平行でエンキューを実行します。



アプリケーションは実行時に新しいシャードを追加できます。アプリケーションは、新しいキーを追加することによって、新しいタイプのメッセージを実行時に追加することもできます。たとえば、オレンジ色と紫色のキーを持つ新しい2つのタイプが導入されて、3番目のシャードCが追加されます。シャードBは、オレンジ色のメッセージを格納するように設定されます。シャードCは、紫色のメッセージを格納するように設定されます。

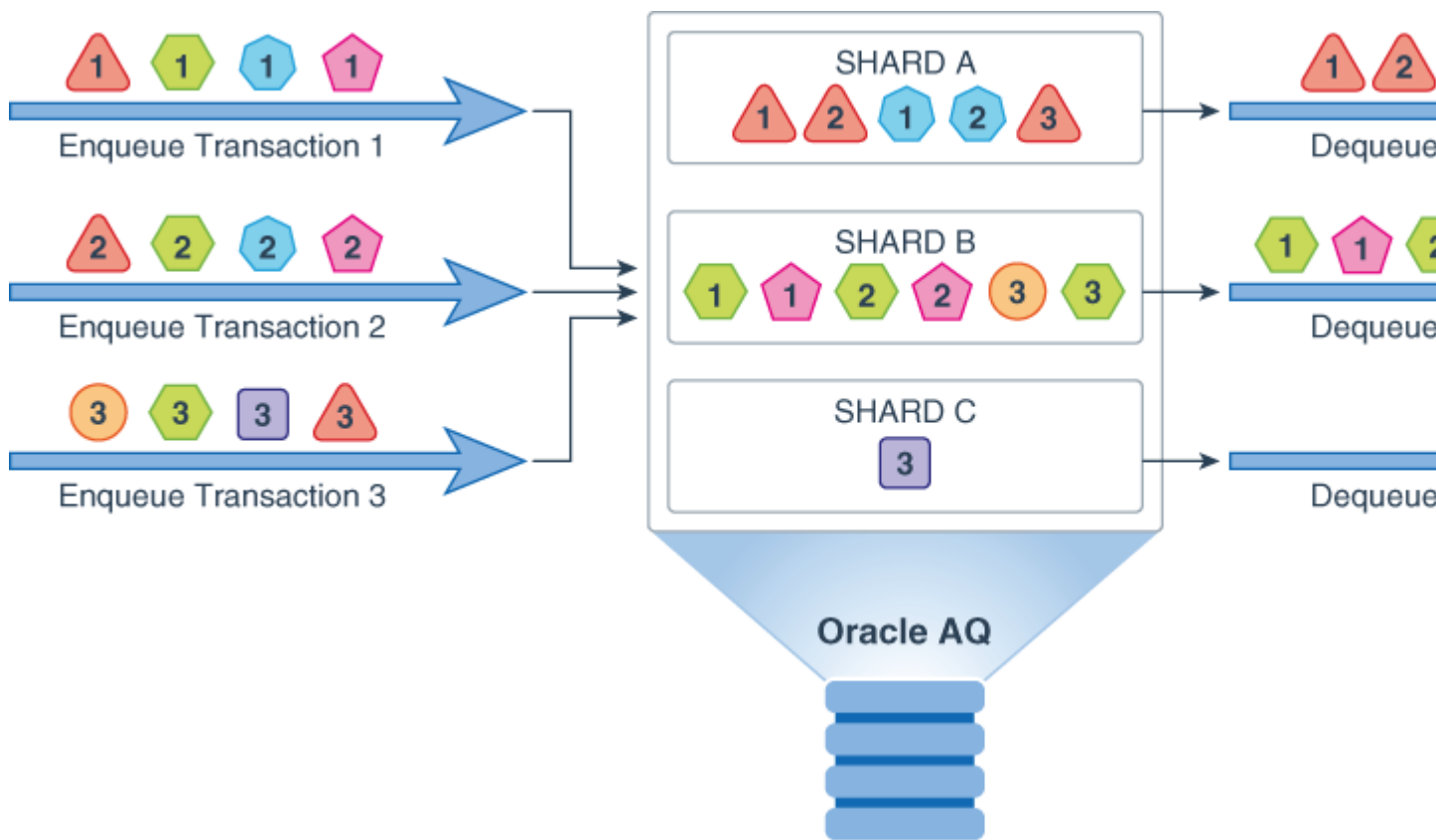


図 adque_usershard2.eps の説明

Oracle RAC データベースでは、シャードは常にインスタンスが所有します。最初は、そのシャードで最初のメッセージがエンキューされるインスタンスがシャードを所有します。データベース・インスタンスが停止すると、シャードの所有者インスタンスが変わる可能性があります。

ユーザー・シャーディングでは、セッションが実行されているインスタンスが所有していないシャードに、ユーザーがメッセージをエンキューできます。そのような場合は、クロス・インスタンス・エンキューがトリガーされます。クロス・インスタンス・エンキューをサポートするために、他のインスタンスで受信されたエンキュー・リクエストは、RAC インターコネクトを介してシャードの OWNER INSTANCE に転送されます。listener.ora の REMOTE_LISTENER パラメータも設定して、クロス・インスタンス・エンキュー・リクエストを正しいインスタンスに転送できるようにする必要があります。内部的には、Oracle RAC データベースのシャード・キューで、インスタンス間のデータベース・リンクが使用される場合があります。Oracle RAC データベースのシャード・キューでクロス・インスタンス・エンキューを実行する定義者権限の PL/SQL パッケージでは、パッケージのユーザーに INHERIT REMOTE PRIVILEGES を付与する必要があります。

ユーザー・シャーディングの制限事項

ユーザー・シャーディングには次の制限事項があります。

- PL/SQL エンキュー・コールでは、クロス・インスタンス・エンキューは有効にされません。
- 配列のエンキューでは、クロス・インスタンス・エンキューは有効にされません。

クロス・インスタンス・エンキューは、Java および OCI クライアントを介して実行できます。

関連項目:

詳細は、Oracle Database PL/SQL パッケージ・プロシージャおよびタイプ・リファレンスの [DBMS_AQADM](#) を参照してください。

非シャード・キュー

この項には次のトピックが含まれます:

- [非シャード・キューの永続メッセージのパフォーマンスの概要](#)
- [永続メッセージの基本的なチューニングのヒント](#)
- [伝播のチューニングのヒント](#)
- [バッファ済メッセージのチューニング](#)

永続メッセージの基本的なチューニングのヒント

Oracle Database Advanced Queuing表のレイアウトは、他のデータベース表および索引のレイアウトと同様です。

関連項目:

チューニングの推奨事項は、[『Oracle Databaseパフォーマンス・チューニング・ガイド』](#)を参照してください。

メモリー要件

Oracle RACデータベース以外で、最適なマルチコンシューマ・デキュー・パフォーマンスを得るには、Streamsプール・サイズは20MB以上である必要があります。

永続キューイング・デキュー操作は、特に同時処理の場合、パフォーマンスを最適化するためにストリーム・プールを使用します。ただし、これは要件ではなく、コードが自動的に最適度の低いコード・パスに切り替わります。

シャード・キューは、高スループットなメッセージ・システムで最適なパフォーマンスを得るためにメッセージ・キャッシュを導入しています。理想として、Streamsプール・サイズは、シャード・キュー内のメッセージの予想されるバックログをキャッシュするのに十分なサイズである必要があります。

格納パラメータの使用方法

格納パラメータを指定するには、キュー表の作成時にstorage_clauseパラメータを使用します。

格納パラメータは、キュー表で作成される他のIOTおよび表に継承されます。キュー表の表領域には、そのキュー表に関連付けられたすべてのオブジェクトのデータを処理できるだけの空き領域が必要です。保存を指定すると、履歴表およびキュー表が相当な大きさになる可能性があります。

自動セグメント領域管理(ASSM)を使用することをお勧めします。使用しない場合、initrans、空きリストおよび空きリスト・グループを高度な同時処理のAQパフォーマンスにチューニングする必要があります。

PCTFREEを増やすと、キュー表/IOTブロック内のメッセージの数を削減できます。これにより、同時処理の際のブロック・レベルの競合が減ります。

キュー表の作成時に指定される格納パラメータは、キュー表、IOTおよび索引で共有されます。これは、DBMS_REDEFINITIONを使用してオンライン再定義で個別に変更できます。

I/O構成

Oracle Database Advanced Queuingでは、I/Oが非常に頻繁に行われるため、通常、あらゆるボトルネックが解消され

るよう、I/Oをチューニングする必要があります。

関連項目:

[『Oracle Databaseパフォーマンス・チューニング・ガイド』](#)のI/O構成および設計に関する項を参照してください。

単一の非シャード・キュー表でのエンキュー・プロセスとデキュー・プロセスの同時実行

メッセージを一定のフローで処理する環境では、エンキュー・プロセスとデキュー・プロセスの両方を同時に実行する必要が生じる場合があります。キュー表とキューが1つずつしか存在しないメッセージ配信システムの場合、すべてのプロセスが同じセグメントを同時に処理する必要があります。これでは適切なパフォーマンス・レベルで、多数のメッセージを配信することはできません。

最適な同時プロセス数は、システム・リソースの可用性によって異なります。たとえば、CPUを4つ持つシステムでは、2つの同時エンキュー・プロセスおよび2つの同時デキュー・プロセスから始めるのが妥当です。必要な数の[メッセージ](#)を配信できないシステムでは、プロセス数を増やすのではなく、複数のサブスクリバを使用して、ロード・バランシングを実行します。

キューのエンキューおよびデキュー率をチューニングすると、通常、キュー・サイズを小さく制限できます。大幅に拡大および縮小するキューには索引およびIOTがあり、バランスに欠けるため、パフォーマンスに影響します。

マルチ・コンシューマ・キューでは、プロセス数を増やすのではなく、ロード・バランシングの複数のサブスクリバを使用して競合を減らします。水平拡張性を実現するには、複数のキュー表を使用します。

シャード・キューのチューニングの詳細は、[シャード・キューのチューニング](#)を参照してください。

単一の非シャード・キュー表でのエンキュー・プロセスとデキュー・プロセスのシリアル実行

エンキュー・プロセスとデキュー・プロセスをシリアルに実行すると、同一のデータ・セグメントにおける競合が、同時プロセスの場合よりも少なくなります。ただし、システムからのメッセージ配信にかかる合計時間は、両プロセスを同時実行する場合よりも長くなります。

プロセス数を増加させることは、エンキュー・プロセスおよびデキュー・プロセスの両方に効果的です。特にシングル・コンシューマ・キューでプロセス数を増加すると、メッセージのスルーput率はデキュー元よりエンキュー元で高くなる場合があります。マルチ・コンシューマ・キューのデキュー・プロセスはさらに拡張されます。

キュー表の索引の作成

次の条件が満たされる場合は、非シャード・キュー表に索引を作成すると役に立ちます。

- 相関識別子を使用したデキュー

基になるキュー表AQ\$_QueueTableNameの列corr_idの索引を作成すると、デキューが迅速に処理されます。

- 条件を使用したデキュー

これは、基になるキュー表で、SELECTのwhere句に対する条件の追加に相当します。QueueTableNameの索引を作成すると、このSELECT文のパフォーマンスが向上します。

非シャード・キューに関するその他のヒント

永続メッセージの基本的なチューニングに関するその他のヒントを示します。

- 統計が収集済で、メッセージ取得の最適な問合せ計画が選択されていることを確認してください。デフォルトでは、キュー表は統計の自動収集からロックされています。統計を代理キュー・メッセージ・ロードで収集して、ロックすることをお勧めします。

- キュー表索引およびIOTは、AQバックグラウンド・プロセスで自動的に結合されます。ただし、必要に応じて監視および結合を続ける必要があります。自動セグメント領域管理(ASSM)のオンライン縮小操作は、同様の目的で使用します。バランスの取れた索引は、キュー・モニターCPUの使用を削減し、エンキュー・デキューのパフォーマンスを最適化します。
- バックグラウンド・タスクを実行できるよう、十分なキュー・モニター・プロセスが実行されていることを確認してください。キュー・モニターは、他の重大なバックグラウンド・アクティビティのためにも実行されている必要があります。複数のqmnプロセスが負荷を分担するため、十分な数があることを確認してください。これらは自動的にチューニングされますが、必要に応じて最小数に強制変更できます。
- 待機時間のあるデキューは、専用サーバー・プロセスでのみ使用することをお勧めします。共有サーバー環境では、(待機時間を含めた)コールの期間中、共有サーバー・プロセスがデキュー操作にのみ占有されます。このようなプロセスが多数存在すると、パフォーマンスおよび拡張性に重大な問題が発生し、共有サーバー・プロセスがデッドロック状態となる可能性があります。
- デキュー・トランザクションを長時間実行するとキューのデキュー競合を悪化させるため、避ける必要があります。
- 1つのトランザクションへのマルチ・コンシューマ・キューの複数のデキュー操作のバッチ処理は、最適なスループットを提供します。
- メッセージの優先順位を使用していない場合、NEXTをナビゲーション・モードとして使用してください。提供されるセマンティクスは同様ですが、パフォーマンスが向上します。
- BROWSEモードのデキューの次がREMOVEの場合、REMOVE_NODATAデキュー・モードを使用してください。

伝播のチューニングのヒント

伝播は、リモート(またはローカル)・キュー表に対して追加のINSERTを行う特殊なデキュー操作です。単一スケジュールからの伝播は、複数のジョブ・キュー・プロセスにパラレル化されません。そうではなく負荷分散されます。伝播は、リモート(またはローカル)・キュー表に対して追加のINSERTを行う特殊なデキュー操作です。単一スケジュールからの伝播は、複数のジョブ・キュー・プロセスにパラレル化されません。かわりに、ロード・バランシングされます。

拡張性を向上させるには、システム・リソース(CPU)の可用性に応じて[伝播](#)スケジュールの数を設定します。

[トランザクション](#)・キュー表からの伝播率と[非トランザクション](#)(デフォルト)キュー表からの伝播率は、多少異なります。これは、非トランザクション・キューのバッチ処理サイズはOracle Database Advanced Queuingが決定するのに対して、トランザクション・キューのバッチ・サイズは主にユーザー・アプリケーションが決定するためです。

伝播の最適化はバッチで行われます。リモート・キューが異なるデータベースに存在する場合、Oracle Database Advanced Queuingは、2フェーズ・コミットの必要性を回避するための順序アルゴリズムを使用します。メッセージが同じ宛先の複数のキューに送信される必要がある場合は、複数回送信されます。メッセージが宛先の同じキュー内のマルチ・コンシューマに送信される必要がある場合、1回のみ送信されます。

バッファ済メッセージのチューニング

Oracle Real Application Clusters環境におけるバッファ済メッセージ操作が最も高速となるのは、キューのOWNER_INSTANCEを対象とする場合です。

非シャード・キューの永続メッセージのパフォーマンスの概要

永続メッセージは、エンキューされるとデータベース表に格納されます。永続メッセージに対するキュー操作のパフォーマンス特性は、基になるデータベース操作のパフォーマンス特性と同等です。

[エンキュー](#)操作のコード・パスは、3つの索引構成表を持つ複数列の[キュー表](#)に対するSELECTおよびINSERTに相当します。[デキュー](#)操作のコード・パスは、複数列の表のSELECT操作およびデキュー索引構成表のDELETE操作に相当します。Oracle

RACを使用しないで適切なストリーム・プール・メモリーが存在する場合などの多くの使用例で、デキュー操作は最適化され、複数列の表のSELECT操作に相当します。

注意:



表内の[キュー](#)の数によって、パフォーマンスが影響を受けることはありません。

非シャード・キューとOracle Real Application Clusters

Oracle Real Application Clusters(Oracle RAC)を使用することで、キュー・データに対する高可用性のアクセスを実現できます。

キューの入口点および出口点(先頭および末尾)は、かなりのホット・スポットになる可能性があります。ホット・スポットが存在すると、Oracle RACを十分に拡張できない場合があるため、1つのキューに通常アクセスするインスタンスは1つに限定します。メッセージを管理しているインスタンスに障害が発生しても、障害の発生していない別のインスタンスの1つを使用することで、そのメッセージの即時処理が可能になります。非シャード・キューでホット・スポットが発生している場合は、かわりにシャード・キューを使用することを検討してください。

Oracle RACのインスタンス・アフィニティを8.1互換のキュー表に対応付けることができます。様々なインスタンスでq1およびq2を使用する場合は、キュー表にALTER_QUEUE_TABLEまたはCREATE_QUEUE_TABLEを使用して、primary_instanceを適切なinstance_idに設定できます。

関連項目:

- [キュー表の作成](#)
- [キュー表の変更](#)
- [シャード・キューおよびOracle Real Application Clusters \(Oracle RAC\)](#)

共有サーバー環境でのOracle Database Advanced Queuing

キュー操作の拡張性は、基になるデータベース操作の拡張性と同等です。

waitオプションを指定して適用されたデキュー操作は、成功するかまたは待機期間が経過するまで戻れません。共有サーバー環境では、(待機時間を含めた)コールの期間中、共有サーバー・プロセスがデキュー操作にのみ占有されます。このようなプロセスが多数存在すると、パフォーマンスおよび拡張性に重大な問題が発生し、共有サーバー・プロセスがデッドロック状態となる可能性があります。そのため、waitオプションを指定したデキュー・リクエストを適用する場合は、専用のサーバー・プロセスを使用することをお勧めします。この制限は、強制ではありません。

関連項目:

waitオプションの詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のDEQUEUE_OPTIONS_T型に関する項を参照してください。

パフォーマンス・ビュー

Oracleでは、システム・パフォーマンスの監視およびトラブルシューティングのための次のビューを提供します。

- [V\\$AQ_MESSAGE_CACHE_STAT: シャード・キューのメモリー管理](#)
- [V\\$AQ_SHARDED_SUBSCRIBER_STAT: シャード・キュー・サブスクライバの統計](#)
- [V\\$AQ_MESSAGE_CACHE_ADVICE: シミュレートされたメトリック](#)
- [V\\$AQ_REMOTE_DEQUEUE_AFFINITY: デキュー・アフィニティ・インスタンス・リスト](#)
- [V\\$PERSISTENT_QUEUES: インスタンスにあるすべてのアクティブな永続キュー](#)
- [V\\$PERSISTENT_SUBSCRIBERS: インスタンスにある永続キューのすべてのアクティブなサブスクライバ](#)
- [V\\$PERSISTENT_PUBLISHERS: インスタンスにある永続キューのすべてのアクティブなパブリッシャ](#)
- [V\\$BUFFERED_QUEUES: インスタンスにあるすべてのバッファ済キュー。](#)
- [V\\$BUFFERED_SUBSCRIBERS: インスタンスにあるすべてのバッファ済キューのサブスクライバ](#)
- [V\\$BUFFERED_PUBLISHERS: インスタンスにあるすべてのバッファ済パブリッシャ](#)
- [V\\$PERSISTENT_QMN_CACHE: 永続キューのバックグラウンド・タスクのパフォーマンス統計](#)
- [V\\$AQ: データベース内の状態ごとのメッセージ数](#)
- [V\\$AQ_BACKGROUND_COORDINATOR: AQのマスター・バックグラウンド・コーディネータ・プロセス\(AQPC\)のパフォーマンス統計](#)
- [V\\$AQ_JOB_COORDINATOR: 1コーディネータ当たりのパフォーマンス統計](#)
- [V\\$AQ_NONDUR_REGISTRATIONS: 非永続登録](#)
- [V\\$AQ_SERVER_POOL: 全サーバーのパフォーマンス統計](#)
- [V\\$AQ_CROSS_INSTANCE_JOBS: クロス・プロセス・ジョブの説明](#)
- [V\\$AQ_NONDUR_REGISTRATIONS: 非永続登録](#)
- [V\\$AQ_NOTIFICATION_CLIENTS: セキュアなOCIクライアント接続](#)
- [V\\$AQ_SUBSCRIBER_LOAD: 永続サブスクライバ](#)
- [V\\$AQ_NONDUR_SUBSCRIBER: 非永続サブスクライバ](#)
- [V\\$AQ_NONDUR_SUBSCRIBER_LWM: 非永続サブスクライバのLWM](#)
- [V\\$AQ_MESSAGE_CACHE: パフォーマンス統計](#)

これらのビューは、自動ワークロード・リポジトリ(AWR)と統合されます。ユーザーは、2つのAWRスナップショットを元にレポートを生成し、エンキュー率、デキュー率およびその他のキュー/サブスクライバ別の統計を計算できます。

6 Oracle Database Advanced Queuingへのインターネット・アクセス

非シャード・キューではSOAPを使用してインターネットを介してOracle Database Advanced Queuing (AQ)にアクセスできます。IDAPはOracle Database Advanced Queuing操作のためのSOAP仕様です。

IDAPは、SOAP ([Simple Object Access Protocol](#))リクエストの本体のXMLメッセージ構造を定義します。IDAP ([Internet Data Access Presentation](#))構造のメッセージは、HTTPを使用してインターネット上で転送されます。

ユーザーは、iDAPインタフェースを使用して通知に登録できます。

トピック:

- [インターネット経由のOracle Database Advanced Queuing操作の概要](#)
- [Oracle Database Advanced Queuing XMLサブレットのデプロイ](#)
- [Internet Data Access Presentation\(IDAP\)](#)
- [リクエストおよびレスポンスのIDAPドキュメント](#)
- [電子メールによるメッセージの通知](#)

インターネット経由のOracle Database Advanced Queuing操作の概要

この項では次のトピックについて説明します。

- [Oracle Database Advanced Queuingのインターネット操作アーキテクチャ](#)
- [インターネット・メッセージ・ペイロード](#)
- [POSTリクエストを送信するユーザーを認証するWebサーバーの構成](#)
- [HTTPを使用したクライアント・リクエスト](#)
- [HTTPを使用したOracle Database Advanced Queuingサブレットのレスポンス](#)
- [HTTPとHTTPSを使用したOracle Database Advanced Queuing伝播](#)

Oracle Database Advanced Queuingのインターネット操作アーキテクチャ

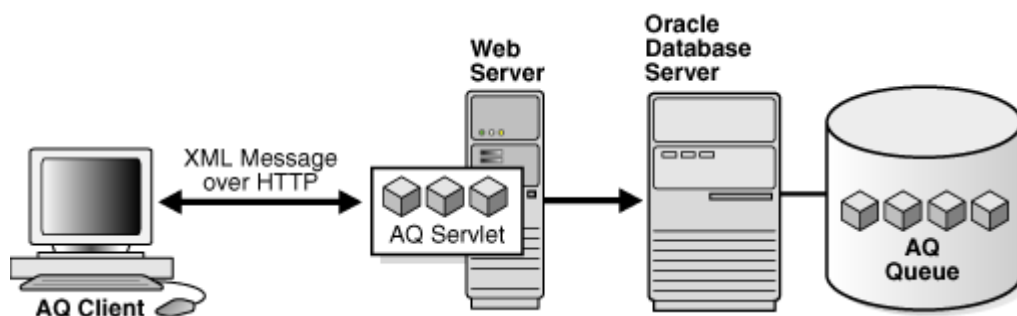
この図は、HTTP上でOracle Database Advanced Queuing操作を実行するためのアーキテクチャを示しています。

主要コンポーネントは次のとおりです。

- Oracle Database Advanced Queuingクライアント・プログラム
- Oracle Database Advanced QueuingサブレットのホストWebサーバー/サブレット・コンテナ
- Oracle Databaseサーバー

Webブラウザまたはその他の任意のHTTPクライアントを、Oracle Database Advanced Queuingのクライアント・プログラムとして使用できます。これらは、XMLメッセージ(IDAP準拠)をOracle Database Advanced Queuingサブレットに送信し、サブレットによって受信XMLメッセージが解析されます。Oracle Database Advanced Queuingサブレットは、Oracle Databaseサーバーに接続し、ユーザー・キューに対して操作を実行します。

図6-1 HTTPを使用してOracle Database Advanced Queuing操作を実行するためのアーキテクチャ



インターネット・メッセージ・ペイロード

Oracle Database Advanced Queuingは、3つのタイプのメッセージ(RAW、OracleオブジェクトおよびJMS)をサポートします。これらのメッセージ型はすべて、SOAPおよびWebサービスを使用してアクセスできます。

キューにRAW、OracleオブジェクトまたはJMS ([Java Message Service](#))フォーマットのメッセージが含まれる場合、XMLのペイロードはエンキュー中に適切な内部フォーマットに変換され、キューに格納されます。デキュー中、前述のいずれかのフォーマットのメッセージを含むキューからメッセージが取得された場合、そのメッセージはクライアントに送信される前にXMLに変換されます。

メッセージ・ペイロード型は、操作が実行されているキュー・タイプによって異なります。

RAWキュー

RAWキューの内容は、バイト列です。XMLメッセージに、次の例のようにメッセージ・ペイロードを16進数で指定する必要があります。例: <raw>023f4523</raw>

Oracleオブジェクト型のキュー

JMSキューではない(AQ\$_JMS_*型ではない)Oracle[オブジェクト型](#)のキューの場合、ペイロードの型はキューを格納するキュー表の作成時に指定した型によって決まります。XML要素の内容は、キュー表のオブジェクト型の属性にマップする必要があります。

JMS型のキュー/トピック

JMS型のキュー(ペイロード型がAQ\$_JMS_*のキュー)の場合、JMS型に応じて4つのXML要素があります。iDAPでは、次のJMS型のキューまたはトピックがサポートされています。

- TextMessage
- MapMessage
- BytesMessage
- ObjectMessage

iDAPでは、ペイロード型StreamMessageのJMSキューはサポートされていません。

POSTリクエストを送信するユーザーを認証するWebサーバーの構成

サーブレットのインストール後、Oracle Database Advanced QueuingサーブレットにPOSTリクエストを送信するすべてのユーザーを認証するようにWebサーバーを構成する必要があります。Oracle Database Advanced Queuingサーブレットは、認証されたユーザーのみにサーブレットへのアクセスを許可します。ユーザーが認証されていない場合、サーブレットからエラーが戻ります。

Webサーバーを構成してアクセスを制限するには、複数の方法があります。一般的な方法は、SSL上でBasic認証(ユーザー名およびパスワード)を行う方法およびクライアント証明書を使用する方法です。サーブレットへのアクセスを制限する方法につい

ては、Webサーバーのドキュメントを参照してください。

Oracle Database Advanced Queuingサブレットの場合、Webサーバーへの接続に使用されるユーザー名は、Oracle Database Advanced Queuing HTTPエージェントまたはOracle Database Advanced Queuingインターネット・ユーザーと呼ばれます。

HTTPを使用したクライアント・リクエスト

Oracle Database Advanced Queuingクライアントは、サーバーへの接続をオープンし、HTTPを使用してOracle Database Advanced Queuingサブレットへのリクエストを開始します。クライアントは、HTTPのBasic認証(SSLを使用する場合も使用しない場合も含む)またはSSL証明書ベースのクライアント認証を使用してサーバーにログインします。クライアントが、SEND、PUBLISH、RECEIVEまたはREGISTERリクエストを表すXMLメッセージを構成します。

リモート・サーバーで、クライアントがサブレットにHTTPのPOSTリクエストを送信します。

関連項目:

[リクエストおよびレスポンスのIDAPドキュメント](#)

ユーザー・セッションおよびトランザクション

クライアントが認証され、Oracle Database Advanced Queuingサブレットに接続すると、ユーザーのかわりにHTTPセッションが作成されます。そのセッションで初めてリクエストを行うと、新しいデータベース・トランザクションが暗黙的に開始します。このトランザクションは、明示的にコミットまたは終了するまでオープン状態のままになります。サブレットからのレスポンスには、HTTPヘッダーにセッションIDがCookieとして含まれます。

クライアントが同一のトランザクションで作業を継続する場合、後続のリクエストにはセッションIDのCookieを含むこのHTTPヘッダーを含める必要があります。これは、ほとんどのWebブラウザで自動的に行われます。ただし、クライアントがJavaまたはCのクライアントを使用してリクエストを転送する場合、プログラムで行う必要があります。

トランザクションを終了するには、明示的なコミットまたはロールバックを適用する必要があります。COMMITまたはROLLBACKリクエストは、他のOracle Database Advanced Queuing操作の一部に組み込むこともできます。

HTTPを使用したOracle Database Advanced Queuingサブレットのレスポンス

サーバーは、クライアントのHTTP(S)接続を受け入れて、クライアントにより指定されたユーザー(Oracle Database Advanced Queuingエージェント)を認証します。サーバーはPOSTリクエストを受信して、Oracle Database Advanced Queuingサブレットを起動します。

このリクエストがこのクライアントからの最初のリクエストである場合、新しいHTTPセッションが作成されます。XMLメッセージが解析され、メッセージの内容が検証されます。HTTPヘッダーにクライアントによってセッションIDが渡される場合、この操作はそのセッションのコンテキストで実行されます。

サブレットが、エージェントが操作を実行しようとしているオブジェクト(キューおよびトピック)を判断します。サブレットが、このOracle Database Advanced Queuingエージェントにマップするデータベース・ユーザーのリスト全体を検索します。いずれかのユーザーがリクエストに指定されたキュー/トピックへのアクセス権限を取得している場合、Oracle Database Advanced Queuingサブレットのスーパーユーザーは、このユーザーのためにセッションを作成します。

HTTPセッションにアクティブなトランザクションが存在しない場合は、新しいデータベース・トランザクションが開始されます。明示的なCOMMITまたはROLLBACKリクエストが行われるまで、セッションの後続のリクエストは同じトランザクションの一部になります。トランザクションは、そのトランザクションがコミットされるまで反映されません。トランザクションがアクティブでない状態が120秒間続

と、そのトランザクションは自動的に終了します。

要求された操作が実行されます。レスポンスがXMLメッセージとしてフォーマットされ、クライアントに戻されます。レスポンスには、セッションIDがCookieとしてHTTPヘッダーに含まれます。

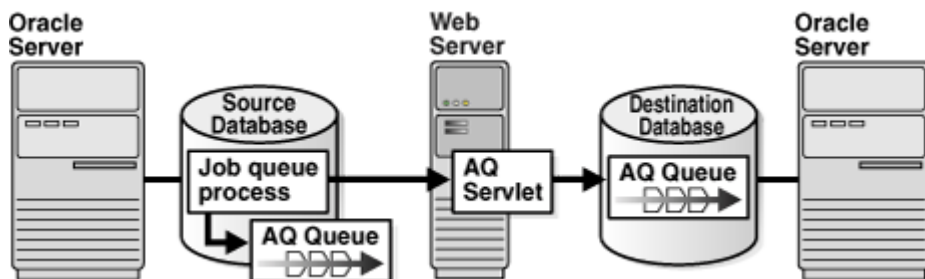
関連項目:

[ユーザー・セッションおよびトランザクション](#)

HTTPとHTTPSを使用したOracle Database Advanced Queuing伝播

Oracle Net ServicesではなくHTTPおよびHTTPS(SSLによるHTTP)上で伝播できます。HTTPは、Oracle Net Servicesとは異なり、ファイアウォールを簡単に構成できます。図に示すように、伝播を行うバックグラウンド・プロセスがOracle Database Advanced Queuingサブレットにメッセージを転送し、そのサブレットが接続先データベースにメッセージをエンキューします。

図6-2 HTTPでのOracle Database Advanced Queuing伝播



既存のコードを変更せずに、アプリケーションがHTTPでのOracle Database Advanced Queuing伝播を使用するように設定できます。HTTPでのOracle Database Advanced Queuing伝播を使用するアプリケーションは、Oracle Net Services接続文字列を使用してデータベース・リンクを再作成するのみで、その他を変更しないで再びOracle Net Servicesでの伝播に容易に戻すことができます。

Oracle Database Advanced Queuing XMLサブレットのデプロイ

AQサブレットは任意のWebサーバー(Tomcatなど)を使用してデプロイできます。Tomcatを使用してAQ XMLサブレットをデプロイするステップは、次のとおりです。

1. JDK 1.8.xの場合は、CLASSPATHに次のように挿入します。

```
ORACLE_HOME/jdbc/lib/ojdbc8.jar
ORACLE_HOME/jlib/jndi.jar
ORACLE_HOME/jlib/jta.jar
ORACLE_HOME/jlib/orai18n.jar
ORACLE_HOME/jlib/orai18n-collation.jar
ORACLE_HOME/jlib/orai18n-mapping.jar
ORACLE_HOME/jlib/orai18n-utility.jar
ORACLE_HOME/lib/http_client.jar
ORACLE_HOME/lib/lclasses12.zip
ORACLE_HOME/lib/servlet.jar
ORACLE_HOME/lib/xmlparserv2.jar
ORACLE_HOME/lib/xschem.jar
ORACLE_HOME/lib/xsu12.jar
ORACLE_HOME/rdbms/jlib/aqapi.jar
ORACLE_HOME/rdbms/jlib/aqxml.jar
ORACLE_HOME/rdbms/jlib/jmscommon.jar
```

```
ORACLE_HOME/rdbms/jlib/xdm.jar
```

2. 次のJARファイルをtomcat/libディレクトリにコピーします。

```
ORACLE_HOME/jdbc/lib/ojdbc8.jar  
ORACLE_HOME/jlib/jndi.jar  
ORACLE_HOME/jlib/jta.jar  
ORACLE_HOME/lib/http_client.jar  
ORACLE_HOME/lib/lclasses12.zip  
ORACLE_HOME/lib/servlet.jar  
ORACLE_HOME/lib/xmlparserv2.jar  
ORACLE_HOME/lib/xschem.jar  
ORACLE_HOME/lib/xsu12.jar  
ORACLE_HOME/rdbms/jlib/aqapi.jar  
ORACLE_HOME/rdbms/jlib/aqxml.jar  
ORACLE_HOME/rdbms/jlib/jmscommon.jar  
ORACLE_HOME/rdbms/jlib/xdm.jar
```

3. キューにアクセスするWebアプリケーション・ユーザーのtomcat-users.xmlファイルを適切に作成または更新します。次に例を示します。

User	Password
john	welcome

4. ステップ3で作成したTomcatユーザーがAQキューにアクセスする前に認証されるように、データベースにキューをセットアップしてAQエージェントを作成します。DBAはDBMS_AQADM.CREATE_AQ_AGENTおよびDBMS_AQADM.ENABLE_DB_ACCESSプロシージャを使用する必要があります。たとえば、Tomcatで作成されたユーザーがJOHNで、データベースで作成されたAQエージェントがAQXMLUSERであると仮定した場合、HTTPを使用してAQサブレットにアクセスするには、次の問合せを実行します。

```
EXECUTE dbms_aqadm.create_aq_agent(agent_name=>'JOHN', enable_http =>true);  
EXECUTE dbms_aqadm.enable_db_access('JOHN', 'AQXMLUSER');
```

ここで、AQXMLUSERはデータベースに作成したAQユーザーです。

DBAは次の問合せを使用してインターネットAQユーザー・エージェントの詳細を確認できます。

```
SELECT agent_name, db_username, http_enabled FROM aq$internet_users ;
```

5. oracle.AQ.xml.AQxmlServletクラスを拡張するAQ XMLサブレットをデプロイします。
6. 次のようにTomcatインスタンスを開始または停止します。
 - a. sh tomcat/bin/startup.shを使用してTomcatインスタンスを開始します。
 - b. sh tomcat/bin/shutdown.shを使用してTomcatインスタンスを停止します。
 - c. Tomcatのログについては、tomcat/logs/catalina.outファイルを確認します。

関連トピック

- [『Fusion Middleware Oracle WebLogic Server JMSの構成と管理』](#)

Internet Data Access Presentation(IDAP)

Internet Data Access Presentation(iDAP)は、Content-Typeヘッダーのtext/xmlを使用して、SOAPリクエストの本体を指定します。

XMLでは、iDAPリクエストおよびレスポンス・メッセージは、次のように表現されます。

- すべてのリクエスト・タグおよびレスポンス・タグは、SOAP名前空間で有効です。
- Oracle Database Advanced Queuing操作はiDAP名前空間で有効です。
- 送信者は、SOAP本体のiDAP要素および属性に名前空間を含めます。
- 受信者は、適切な名前空間を含むSOAPメッセージを処理し、不適切な名前空間を含むリクエストの場合はリクエストが無効であるというエラーを戻します。
- SOAP名前空間の値は<http://schemas.xmlsoap.org/soap/envelope/>です。
- iDAP名前空間の値は<http://ns.oracle.com/AQ/schemas/access>です。

SOAPメッセージの構造

次のトピックでは、SOAPによってメッセージ・リクエストまたはレスポンスが構造化される仕組みを説明します。

- [SOAPエンベロープ](#)
- [SOAPヘッダー](#)
- [SOAP本体](#)

SOAPエンベロープ

これは、XMLツリーのルートまたは最上位要素です。タグはSOAP:Envelopeです。SOAPは、グローバル属性SOAP:encodingStyleを定義し、これはSOAP仕様によって記述されるルールのかわりに使用されるシリアライズ・ルールを示します。

この属性は、任意の要素で使用でき、この要素と、このような要素が含まれていないすべての子要素で有効です。この属性が省略されている場合、親要素によってオーバーライドされた場合を除き、型の指定に従うことを示します。

名前空間で修飾されている場合、SOAPエンベロープには名前空間宣言およびその他の属性も含まれます。本体には、名前空間で修飾された他のサブ要素を続けることができます。

SOAPヘッダー

これは、ルート下の最初の要素です。タグはSOAP:Headerです。SOAPヘッダーは、トランザクションIDなどの必要な情報を渡します。

ヘッダーは、SOAP:EnvelopeというXML要素の子としてエンコードされます。ヘッダーは名前要素で識別され、名前空間で修飾されます。ヘッダー・エントリは、埋込み要素としてエンコードされます。

SOAP本体

これは、Oracle Database Advanced QueuingのXML文書です。タグはSOAP:Bodyで、メソッド名を名前とする最初のサブ要素が含まれます。

このメソッド・リクエスト要素には、各入出力パラメータの要素が含まれます。要素名はパラメータ名です。また、本体には、エラーの情報を示すSOAP:Faultも含まれます。Oracle Database Advanced QueuingのXML文書の名前空間は、<http://ns.oracle.com/AQ/schemas/access>となります。

SOAPメソッドの起動

メソッドの起動は、リクエストのヘッダーおよび本体を作成し、戻されたレスポンスのヘッダーおよび本体を処理することによって実行されます。リクエストおよびレスポンスのヘッダーは、標準の転送プロトコル固有のヘッダーおよび拡張ヘッダーで構成されます。

HTTPヘッダー

HTTPリクエスト・ヘッダー内のPOSTメソッドにより、SOAPのメソッドが起動されます。リクエストにはヘッダーSOAPMethodNameが含まれている必要があります。このヘッダーの値は、ターゲットに対して起動するメソッドを示します。この値の形式は *URI#methodName* です。

次に例を示します。

```
SOAPMethodName: http://ns.oracle.com/AQ/schemas/access#AQXmlSend
```

インタフェースに使用するURIは、ペイロードのSOAP:Body部に存在する、メソッド名要素の暗示または指定された名前空間修飾に一致する必要があります。メソッド名に#文字を含めることはできません。

メソッド起動本体

SOAPメソッドの起動は、メソッド・リクエストおよびメソッド・レスポンス(オプション)で構成されます。SOAPのメソッド・リクエストおよびメソッド・レスポンスは、それぞれHTTPのリクエストおよびレスポンスで、その内容は、ルート要素および必須の本体要素で構成されるXML文書です。

この項では、このXML文書をSOAPペイロードと呼びます。

SOAPペイロードの定義は次のとおりです。

- SOAPのルート要素は、XMLツリーの最上位要素です。
- SOAPペイロードのヘッダーには、リクエストとともに送信する必要がある追加情報が含まれます。
- メソッド・リクエストは、パラメータ用の追加要素を持つXML要素として表現されます。これはSOAP:Body要素の最初の子です。このリクエストは、次の項で説明するOracle Database Advanced QueuingのXMLクライアント・リクエストの1つに含まれます。
- レスポンスは、戻り値、またはクライアントに戻されるエラーまたは例外です。

受信サイトでは、リクエストの結果は次のいずれかになります。

- 受信サイトのHTTPインフラストラクチャが、リクエストを受信および処理できません。この場合、HTTPインフラストラクチャはSOAPインフラストラクチャにヘッダーおよび本体を渡します。
- 受信サイトのHTTPインフラストラクチャが、リクエストを受信および処理できません。この場合、HTTPレスポンスには状態フィールドにHTTPエラーが含まれ、XMLの本体は含まれません。
- 受信サイトのSOAPインフラストラクチャが、入力パラメータをデコードし、サーバー・アドレスで示される適切なサーバーにディスパッチし、メソッド・リクエストに示されているメソッドに意味的に対応するアプリケーション・レベルのファンクションを起動できます。この場合、メソッド・リクエストの結果はレスポンスまたはエラーで構成されます。
- 受信サイトのSOAPインフラストラクチャが、入力パラメータをデコードし、サーバー・アドレスで示される適切なサーバーにディスパッチし、メソッド・リクエストに示されているインタフェースまたはメソッドに意味的に対応するアプリケーション・レベルのファンクションを起動できません。この場合、メソッドの結果はエラーになり、受信側でのインフラストラクチャのディスパッチを正常終了させることができません。

最後の2つのケースの場合は、拡張性の目的で追加のメッセージ・ヘッダーをリクエストの結果に表示できます。

メソッド・リクエストの結果

リクエストの結果は、リクエスト/レスポンスの形式で提供されます。HTTPのレスポンスでは、Content-Typeヘッダーがtext/xmlである必要があります。

SOAPの結果は成功を示し、エラーは失敗を示します。メソッド・レスポンスには、結果とエラーの両方が含まれることはありません。

ん。

リクエストおよびレスポンスのIDAPドキュメント

SOAPメッセージの本体は、iDAPメッセージです。このXML文書の名前空間は、<http://ns.oracle.com/AQ/schemas/access>となります。

本体は次の内容を表しています。

- クライアントの[エンキュー](#)、[デキュー](#)および登録リクエスト
- クライアントのエンキュー、デキューおよび登録リクエストに対するサーバーのレスポンス
- サーバーからクライアントへの通知



注意:

Oracle Database Advanced Queuing インターネット・アクセスは、8.1 以上の形式のキューのみに対してサポートされています。

シャード・キューは SOAP を介したインターネット・アクセスをサポートしていません。

この項の内容は次のとおりです。

- [IDAPクライアントのエンキュー・リクエスト](#)
- [IDAPクライアントのデキュー・リクエスト](#)
- [IDAPクライアントの登録リクエスト](#)
- [IDAPクライアントのトランザクション・コミット・リクエスト](#)
- [IDAPクライアントのトランザクション・ロールバック・リクエスト](#)
- [エンキュー・リクエストへのIDAPサーバー・レスポンス](#)
- [デキュー・リクエストへのIDAPサーバー・レスポンス](#)
- [登録リクエストへのIDAPサーバー・レスポンス](#)
- [iDAPコミット・レスポンス](#)
- [iDAPロールバック・レスポンス](#)
- [IDAP通知](#)
- [エラー発生時のIDAPレスポンス](#)

IDAPクライアントのエンキュー・リクエスト

クライアントは、リクエストの送信およびパブリッシュについて、シングル・コンシューマ・キューへのエンキューにはAQXMLSend、マルチ・コンシューマ・キュー/トピックへのエンキューにはAQXMLPublishを使用します。

AQXMLSendおよびAQXMLPublishには、次の要素が含まれています。

- [producer_options](#)

- [message_set](#)
- [message_header](#)
- [message_payload](#)
- [AQXmlCommit](#)

producer_options

これは必須要素です。次の子要素が含まれます。

- destination

この要素は必須です。メッセージの送信先[キュー](#)またはトピックを指定します。送信先の値の解析方法を決定する、オプションのlookup_type属性を持ちます。lookup_typeがDATABASEの場合(デフォルト)、送信先はschema.queue_nameとして解析されます。lookup_typeがLDAPの場合、送信先の解決にLDAPサーバーが使用されます。
- visibility

この要素は省略可能です。エンキューがいつ可視になるかを決定します。デフォルトはON_COMMITで、現行のトランザクションがコミットするとエンキューが可視になります。IMMEDIATEが指定されると、エンキューの状態は、リクエスト完了直後に可視になります。エンキューは現行のトランザクションの一部ではありません。操作がそれ自体でトランザクションを形成します。
- transformation

この要素は省略可能です。メッセージをエンキューする前に、PL/SQL [変換](#)を開始するように指定します。

message_set

これは必須要素で、1つ以上のメッセージが含まれます。各メッセージは[message_header](#)および[message_payload](#)で構成されます。

message_header

この要素は省略可能です。次の子要素が含まれます。

- sender_id

message_header要素が含まれている場合は、アプリケーション固有の識別子を指定するsender_id要素を含める必要があります。sender_id要素には、agent_name、address、protocolおよびagent_aliasの各要素を含めることができます。agent_alias要素は、LDAPを使用して名前、アドレスおよびプロトコルを解決します。
- message_id

この要素は省略可能です。デキュー時に提供されるメッセージの一意の識別子です。
- correlation

この要素は省略可能です。メッセージの相関識別子です。
- delay

この要素は省略可能です。メッセージが処理可能になるまでの期間(秒)を指定します。
- expiration

この要素は省略可能です。メッセージがデキュー可能な期間(秒)を指定します。このパラメータは、遅延からのオフセットです。デフォルトでは、メッセージは期限切れになりません。期限切れになる前にデキューされない場合、メッセージはEXPIRED状態で[例外キュー](#)に移されます。
- priority

この要素は省略可能です。メッセージの優先度を指定します。優先度は、負数も含めたあらゆる数値で指定できます。数値が小さいほど高い優先度を示します。

- recipient_list

この要素は省略可能です。デフォルトのサブスクライバ・リストに優先する受信者のリストです。各受信者は recipient_list で recipient 要素として表現され、これには agent_name、address、protocol および agent_alias 要素を含めることができます。agent_alias 要素は、LDAP を使用して名前、アドレスおよびプロトコルを解決します。

- message_state

この要素は省略可能です。メッセージの状態を指定します。デキュー時に自動的に決定されます。message_state が 0 (ゼロ) の場合、メッセージは処理可能な状態です。1 の場合、メッセージは遅延の指定秒数に達していません。2 の場合、メッセージは処理され、保存されています。3 の場合は、メッセージが例外キューに移動されたことを示します。

- exception_queue

この要素は省略可能です。デキューが max_retries 以内の回数で成功しなかった場合、またはメッセージの有効期限が切れた場合、メッセージの移動先キューの名前を指定します。例外キューのメッセージはすべて EXPIRED 状態です。

指定した例外キューが移動時に存在しない場合、そのメッセージは、キュー表に関連付けられているデフォルトの例外キューに移動され、アラート・ログに警告ログが記録されます。デフォルトの例外キューが使用されると、デキュー時にパラメータが NULL 値を戻します。

message_payload

これは必須要素です。送信先キューまたはトピックのペイロード型により、異なる要素が含まれます。様々なペイロード型については、[「IDAPクライアントのデキュー・リクエスト」](#)で説明しています。

AQXmlCommit

これはオプションの空要素です。この要素が含まれている場合、トランザクションはリクエストの終了時にコミットされます。

関連項目:

IDAPメッセージ・ペイロードの説明は、[「インターネット・メッセージ・ペイロード」](#)を参照してください。

IDAPクライアントのデキュー・リクエスト

クライアントのデキュー・リクエストでは、次の要素を含む AQXmlReceive を使用します。

- [consumer_options](#)
- [AQXmlCommit](#)

consumer_options

これは必須要素です。次の子要素が含まれます。

- destination

この要素は必須です。メッセージの受信先キューまたはトピックを指定します。destination 要素は、送信先の値の解析方法を決定する、オプションの lookup_type 属性を持ちます。lookup_type が DATABASE の場合 (デフォルト)、送信先は schema.queue_name として解析されます。lookup_type が LDAP の場合、送信先の解決に LDAP サーバーが使用されます。

- consumer_name

この要素は省略可能です。[コンシューマ](#)の名前を指定します。コンシューマ名が一致するメッセージのみアクセスされます。キューがマルチ・コンシューマ用に設定されていない場合、このフィールドは指定しないでください。

- wait_time

この要素は省略可能です。検索基準に一致するメッセージが存在していない場合の待機秒数を指定します。

- selector

この要素は省略可能です。メッセージの選択に使用する基準を指定します。子要素correlation、message_idまたはconditionを含められます。

デキューのcondition要素は、SQL問合せのWHERE句に類似した構文を使用するブール式で指定します。このブール式には、メッセージ・プロパティ、ユーザー・オブジェクトのペイロード・データ・プロパティおよびPL/SQLまたはSQL関数に関する条件を含めることができます。メッセージ・プロパティには、priority、corr_idおよびキュー表内の他の列を含めることができます。

メッセージ・ペイロードにデキュー条件を指定するには、句に[オブジェクト型](#)の属性を使用します。各属性には、ペイロードを格納するキュー表の特定の列を示す識別子として、接頭辞tab.user_dataが必要です。

デキューのcondition要素は4000文字以内です。



注意:

デキュー条件または関連識別子を使用すると、メッセージのデキュー順序が不定になり、キューのソート順序は考慮されません。

- visibility

この要素は省略可能です。デキューがいつ可視になるかを決定します。デフォルトはON_COMMITで、現行のトランザクションがコミットするとデキューが可視になります。IMMEDIATEが指定されると、デキューの状態は、リクエスト完了直後に可視になります。デキューは現行のトランザクションの一部ではありません。操作がそれ自体でトランザクションを形成します。

- dequeue_mode

この要素は省略可能です。デキューに対応付けられたロック動作を指定します。可能な値は、REMOVE、BROWSE、およびLOCKEDです。

REMOVEはデフォルトで、メッセージは読み取られて削除されます。メッセージは保存プロパティに基づいてキュー表に保存されます。BROWSEを指定すると、メッセージのロックを取得しないで、メッセージを読み取ります。これは、SELECT文と同じです。LOCKEDを指定すると、メッセージを読み取って書き込みロックを取得します。ロックは、トランザクションの間、存続します。これはUPDATE文の選択と同じです。

- navigation_mode

この要素は省略可能です。取り出されるメッセージの位置を指定します。最初に位置が判断されます。次に検索基準が適用されます。最後にメッセージが取り出されます。可能な値は、FIRST_MESSAGE、NEXT_MESSAGEおよびNEXT_TRANSACTIONです。

FIRST_MESSAGEは、使用可能で検索基準と一致する最初のメッセージを取り出します。これにより、位置はキューの先頭にリセットされます。NEXT_MESSAGEはデフォルトで、使用可能で検索基準と一致する次のメッセージを取り出します。前のメッセージがメッセージ・グループに属する場合は、Oracle Database Advanced Queuingにより、検索

基準に一致し、メッセージ・グループに属する次の使用可能なメッセージが取り出されます。NEXT_TRANSACTIONによって、現行のトランザクション・グループの残りのメッセージがスキップされ、次のトランザクション・グループの最初のメッセージが取り出されます。このオプションは、現行のキューに対してメッセージをグループ化できる場合のみ使用できます。

- transformation

この要素は省略可能です。メッセージをデキューした後に、PL/SQL [変換](#)を開始するように指定します。

AQXmlCommit

これはオプションの空要素です。この要素が含まれている場合、トランザクションはリクエストの終了時にコミットされます。

IDAPクライアントの登録リクエスト

登録のクライアント・リクエストにはAQXmlRegisterが使用され、これにはregister_options要素が含まれている必要があります。register_options要素には、次の子要素が含まれます。

- destination

この要素は必須です。通知の登録先キューまたはトピックを指定します。destination要素は、送信先の値の解析方法を決定する、オプションのlookup_type属性を持ちます。lookup_typeがDATABASEの場合(デフォルト)、送信先はschema.queue_nameとして解析されます。lookup_typeがLDAPの場合、送信先の解決にLDAPサーバーが使用されます。

- consumer_name

この要素は省略可能です。マルチ・コンシューマ・キューまたはトピックのコンシューマ名を指定します。シングル・コンシューマ・キューの場合はこのパラメータを指定しないでください。

- notify_url

この要素は必須です。メッセージがエンキューされたときの通知の送信先を指定します。書式はhttp://url/mailto://email_addressまたはplsql://pl/sql_procedureを使用できます。

IDAPクライアントのトランザクション・コミット・リクエスト

セッション中にユーザーが実行するすべてのアクションのコミット・リクエストには、AQXmlCommitを使用します。

COMMITリクエストのフォーマットは、次のようになります。

```
<?xml version="1.0"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <AQXmlCommit xmlns="http://ns.oracle.com/AQ/schemas/access"/>
  </Body>
</Envelope>
```

IDAPクライアントのトランザクション・ロールバック・リクエスト

セッション中にユーザーが実行するすべてのアクションのロールバック・リクエストには、AQXmlRollbackを使用します。可視性をIMMEDIATEに設定して実行するアクションは、ロールバックされません。

IDAPクライアントのROLLBACKリクエストのフォーマットは次のとおりです。

```
<?xml version="1.0"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <AQXmlRollback xmlns="http://ns.oracle.com/AQ/schemas/access"/>
  </Body>
```

エンキュー・リクエストへのIDAPサーバー・レスポンス

シングル・コンシューマ・キューへのエンキュー・リクエストに対するレスポンスでは、AQXmlSendResponseを使用します。

次の要素が含まれています。

- status_response

この要素には、子要素status_code、error_codeおよびerror_messageが含まれます。status_code要素は、成功した場合は値0、失敗した場合は値-1を取ります。error_code要素には、Oracleエラー・コードが含まれます。error_message要素には、エラーの説明が含まれます。

- send_result

この要素には、子要素destinationおよびmessage_idが含まれます。destination要素では、メッセージが送信された場所を指定します。message_id要素では、送信された各メッセージを一意に識別します。

マルチ・コンシューマ・キューまたはトピックへのエンキュー・リクエストに対するレスポンスでは、AQXmlPublishResponseを使用します。次の要素が含まれています。

- status_response

この要素には、子要素status_code、error_codeおよびerror_messageが含まれます。status_code要素は、成功した場合は値0、失敗した場合は値-1を取ります。error_code要素には、Oracleエラー・コードが含まれます。error_message要素には、エラーの説明が含まれます。

- publish_result

この要素には、子要素destinationおよびmessage_idが含まれます。destination要素では、メッセージが送信された場所を指定します。message_id要素では、送信された各メッセージを一意に識別します。

デキュー・リクエストへのIDAPサーバー・レスポンス

デキュー・リクエストに対するレスポンスでは、AQXmlReceiveResponseを使用します。

次の要素が含まれています。

- status_response

この要素には、子要素status_code、error_codeおよびerror_messageが含まれます。status_code要素は、成功した場合は値0、失敗した場合は値-1を取ります。error_code要素には、Oracleエラー・コードが含まれます。error_message要素には、エラーの説明が含まれます。

- receive_result

この要素には、子要素destinationおよびmessage_setが含まれます。destination要素では、メッセージが送信された場所を指定します。message_set要素では、デキューされたメッセージのセットを指定します。

登録リクエストへのIDAPサーバー・レスポンス

登録リクエストに対するレスポンスでは、AQXmlRegisterResponseを使用します。

これには、[「デキュー・リクエストへのIDAPサーバー・レスポンス」](#)で説明したstatus_response要素が含まれます。

IDAPコミット・レスポンス

コミット・リクエストに対するレスポンスでは、AQXmlCommitResponseを使用します。

これには、[「デキュー・リクエストへのIDAPサーバー・レスポンス」](#)で説明したstatus_response要素が含まれます。コミット・リクエストに対するレスポンスは、次のような書式になります。

```
<?xml version = '1.0' ?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <AQXmlCommitResponse xmlns="http://ns.oracle.com/AQ/schemas/access">
      <status_response>
        <status_code>0</status_code>
      </status_response>
    </AQXmlCommitResponse>
  </Body>
</Envelope>
```

IDAPロールバック・レスポンス

ロールバック・リクエストに対するレスポンスでは、AQXmlRollbackResponseを使用します。

これには、[「デキュー・リクエストへのIDAPサーバー・レスポンス」](#)で説明したstatus_response要素が含まれます。

IDAP通知

クライアントが登録したイベントが発生した場合、REGISTERリクエストで指定したURLにAQXmlNotificationを使用してクライアントへの通知が送信されます。

次の要素が含まれています。

- notification_options

この要素には、子要素destinationおよびconsumer_nameがあります。destination要素では、イベントが発生した送信先キューまたはトピックを指定します。consumer_name要素では、イベントが発生したコンシューマ名を指定します。マルチ・コンシューマ・キューまたはトピックにのみ適用されます。

- message_set

この要素ではメッセージ・プロパティのセットを指定します。

エラー発生時のIDAPレスポンス

前述のいずれかのリクエストでエラーが発生した場合、FAULT要素が生成されます。

FAULT要素には、次の要素が含まれます。

- faultcode

この要素では、失敗を示すエラー・コードを指定します。

- faultstring

この要素は、クライアント・エラーまたはサーバー・エラーを示します。クライアント・エラーは、リクエストが無効であることを示します。サーバー・エラーは、Oracle Database Advanced Queuingサブレットが正しく設定されていないことを示します。

- detail

この要素には、[「デキュー・リクエストへのIDAPサーバー・レスポンス」](#)で説明したstatus_response要素が含まれます。

電子メールによるメッセージの通知

ここでは、データベースの電子メール通知の設定ステップを説明します。

1. Oracle Database Advanced Queuing管理者としてDBMS_AQELM.SET_MAILHOSTを起動し、SMTPメール・ホストを設定します。
2. Oracle Database Advanced Queuing管理者としてDBMS_AQELM.SET_MAILPORTを起動し、SMTPメール・ポートを設定します。特別な場合を除き、デフォルトの25に設定します。
3. DBMS_AQELM.SET_SENDFROMを起動してSendFromアドレスを設定します。
4. 設定後、Oracle Call Interface(OCI)またはPL/SQL APIを使用することで、電子メール通知の登録ができます。

7 Oracle Database Advanced Queuingのトラブルシューティング

次のトピックでは、Oracle Database Advanced Queuing (AQ)のトラブルシューティング方法を説明します。

- [Oracle Database Advanced Queuingの伝播に関する問題のデバッグ](#)
- [Oracle Database Advanced Queuingのエラー・メッセージ](#)

Oracle Database Advanced Queuingの伝播に関する問題のデバッグ

次のヒントは、伝播に関する問題のデバッグに役立ちます。ここでは、ソースおよびターゲット・データベースにキュー表およびキューを作成し、接続先データベースに対するデータベース・リンクを定義していると想定します。

表記法は、大カッコなしでエンティティの実際の名前が提供されるものと想定します。

関連項目:

[伝播の最適化](#)

デバッグを開始するには、次の手順を実行します。

1. 伝播スケジュールが作成され、ジョブ・キュー・プロセスが割り当てられたことを確認します。

DBA_QUEUE_SCHEDULESビューのエントリを検索し、スケジュールのステータスが有効であることを確認します。

SCHEDULE_DISABLEDがNに設定されている必要があります。表AQ\$_SCHEDULESでJOBNO(について0(ゼロ)以外のエントリがあることと、表JOB\$_にそのJOBNOを含むエントリがあることを確認します。

伝播が発生していることを確認するには、伝播済のメッセージの数(TOTAL_NUMBER)のDBA_QUEUE_SCHEDULESビューを監視します。

伝播が発生していない場合は、ビューでエラーをチェックします。また、DBA_QUEUE_SCHEDULESのNEXT_RUN_DATEとNEXT_RUN_TIMEをチェックし、エラーまたは設定方法に応じて後で伝播を実行するようにスケジュールされていないか確認します。

2. 接続先データベースへのデータベース・リンクが正しく設定されていることを確認します。キュー所有者がデータベース・リンクを使用できることを確認します。この場合は、次のように入力します。

```
select count (*) from table_name@dblink_name;
```

3. 2つ以上のジョブ・キュー・プロセスが実行中であることを確認します。
4. ソース・[キュー](#)のメッセージを次のようにして確認します。

```
select count (*) from AQ$<source_queue_table>
where q_name = 'source_queue_name';
```

5. 宛先キューのメッセージを次のようにして確認します。

```
select count (*) from AQ$<destination_queue_table>
where q_name = 'destination_queue_name';
```

6. 誰がジョブ・キュー・プロセスを使用しているかを確認します。

dba_jobs_runningを問い合わせ実行されているジョブを確認します。他のジョブが伝播ジョブに影響を及ぼしている可能性があります。

7. キュー表sys.aq\$prop_table_instnoがDBA_QUEUE_TABLESにあることを確認します。また、キューsys.aq\$prop_notify_queue_instnoがDBA_QUEUESにあり、エンキューおよびデキューに使用可能になっている必要があります。

Oracle Real Application Clusters(Oracle RAC)の場合は、このキュー表とキュー・ペアがシステム内の各Oracle RACノードにある必要があります。キュー表およびキュー・ペアはジョブ・キュー・プロセス間の通信に使用され、自動的に作成されます。

8. 宛先キューからメッセージをデキューするコンシューマが、伝播されたメッセージの受信者であることを確認します。

8.1形式のキューでは、次のように入力します。

```
select consumer_name, deq_txn_id, deq_time, deq_user_id,
       propagated_msgid from aq$<destination_queue_table>
       where queue = 'queue_name';
```

8.0形式のキューでは、次のように入力してキュー表の履歴列から同じ情報を取得できます。

```
select h.consumer, h.transaction_id, h.deq_time, h.deq_user,
       h.propagated_msgid from aq$<destination_queue_table> t, table(t.history) h
       where t.q_name = 'queue_name';
```

注意:



compatible が 8.0 に設定されているキュー表で作成されたキュー(このマニュアルでは 8.0 形式のキューと呼びます)は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になっています。したがって、新しいキューの作成には 8.1 以降の形式を使用し、既存の 8.0 形式のキューをなるべく早く移行することをお勧めします。

9. イベント24040、レベル10を使用して、[伝播](#)トレースを最高レベルでONにします。

伝播の発生時に、デバッグ情報がジョブ・キュー・トレース・ファイルに記録されます。トレース・ファイルで、エラーがないか、またメッセージが送信されたことを示す文を確認できます。

Oracle Database Advanced Queuingのエラー・メッセージ

Oracle Database Advanced Queuingのエラー・メッセージを次に示します。

ORA-1555

デキューにNEXT_MESSAGEナビゲーション・オプションを使用すると、このエラー・メッセージが表示される場合があります。

NEXT_MESSAGEでは、最初のデキュー・コール中に作成されたスナップショットが使用されます。その後で、UNDO情報が削除される場合があります。

これを解決するには、FIRST_MESSAGEオプションを使用してメッセージをデキューします。これによってカーソルが再実行され、新しいスナップショットが取得されます。FIRST_MESSAGEはNEXT_MESSAGEのように実行されないため、1つのメッセージにFIRST_MESSAGEを使用し、次の1000のメッセージにNEXT_MESSAGEを使用し、再びFIRST_MESSAGEを使用するというように、メッセージをバッチでデキューすることをお勧めします。

ORA-24033

メッセージが受信者を指定しないでマルチ・コンシューマ・キューにエンキューされ、そのキューにサブスクライバが指定されていない (またはそのメッセージに一致するルールベースのサブスクライバが存在しない) 場合、このエラーが発生します。これは、配信可能な受信者またはサブスクライバが存在しないために、そのメッセージが廃棄されることを示す警告です。

ORA-25237

サービス間(xa_start境界とxa_end境界の間など)でデキュー処理を続ける場合に、Oracle Database Advanced Queuingナビゲーション・オプションを使用するときは、FIRST_MESSAGEを使用してデキュー位置をリセットする必要があります。これは、xa_endが終了すると、XAがカーソル・フェッチ状態を取り消すためです。リセットしないと、ナビゲーション・オプションの指定順序が正しくないというエラー・メッセージが表示されます。

ORA-25307

メッセージ送信者に対してフロー制御が有効になっています。これは、送信者のメッセージのうち最も高速のサブスクライバがメッセージのエンキュー速度に追いつけないことを意味します。バッファ済メッセージング・アプリケーションは、このエラーを処理し、少し待ってからメッセージを再度エンキューする必要があります。

8 Oracle Database Advanced Queuing管理インタフェース

次のトピックでは、Oracle Database Advanced Queuing (AQ)の管理インタフェースについて説明します。

- [非シャード・キュー表の管理](#)
- [非シャード・キューの管理](#)
- [シャード・キューの管理](#)
- [変換の管理](#)
- [権限の付与および取消し](#)
- [サブスクライバの管理](#)
- [伝播の管理](#)
- [Oracle Database Advanced Queuingエージェントの管理](#)
- [LDAPサーバーへの別名の追加](#)
- [LDAPサーバーからの別名の削除](#)

関連項目:

- 各プログラム・インタフェースで使用可能な機能のリストは、[「Oracle Database Advanced Queuing: プログラム・インタフェース」](#)を参照してください。
- DBMS_AQADMパッケージの詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』を参照してください。

非シャード・キュー表の管理

次のトピックでは、非シャード・キュー表を管理する方法について説明します。

- [キュー表の作成](#)
- [キュー表の変更](#)
- [キュー表の削除](#)
- [キュー表のページ](#)
- [キュー表の移行](#)

キュー表の作成

DBMS_AQADM.CREATE_QUEUE_TABLEは、事前定義タイプのメッセージ用の非シャード・キュー表を作成します。

```
DBMS_AQADM.CREATE_QUEUE_TABLE(  
  queue_table      IN      VARCHAR2,  
  queue_payload_type IN    VARCHAR2,  
  storage_clause   IN      VARCHAR2          DEFAULT NULL,
```

```

sort_list          IN          VARCHAR2          DEFAULT NULL,
multiple_consumers IN          BOOLEAN              DEFAULT FALSE,
message_grouping  IN          BINARY_INTEGER      DEFAULT NONE,
comment           IN          VARCHAR2          DEFAULT NULL,
auto_commit       IN          BOOLEAN              DEFAULT TRUE,
primary_instance  IN          BINARY_INTEGER      DEFAULT 0,
secondary_instance IN        BINARY_INTEGER      DEFAULT 0,
compatible        IN          VARCHAR2          DEFAULT NULL,
secure            IN          BOOLEAN              DEFAULT FALSE
replication_mode  IN          BINARY_INTEGER      DEFAULT NONE);

```

次の必須およびオプションのパラメータがあります。

パラメータ	説明
queue_table	<p>この必須パラメータは、キュー表名を指定します。</p> <p>データベースの互換性が 10.0 である場合のみ、大文字と小文字が混在するキュー表の名前を指定できますが、名前を二重引用符で囲む必要があります。したがって、abc.efg は、スキーマが ABC で名前が EFG であることを示しますが、“abc”.“efg”はスキーマが abc で名前が efg であることを示します。</p> <p>12c リリース 2 (12.2)以降では、AQ キュー表名の最大長は 122 バイトです。長い名前のキュー表を作成しようとすると、エラーORA-24019 が発生します。</p>
queue_payload_type	<p>この必須パラメータは、ペイロード型を RAW またはオブジェクト型として指定します。詳細は、「ペイロード型」を参照してください。</p>
storage_clause	<p>このオプション・パラメータは、キュー表の表領域を指定します。詳細は、「STORAGE 句」を参照してください。</p>
sort_list	<p>このオプション・パラメータは、昇順でソート・キーとして使用される 1 つまたは 2 つの列を指定します。<i>sort_column1, sort_column2</i>という書式を使用します。詳細は、「ソート・キー」を参照してください。</p>
multiple_consumers	<p>このオプション・パラメータは、キュー表をシングル・コンシューマまたはマルチ・コンシューマとして指定します。デフォルトの FALSE は、表で作成されたキューにはメッセージごとに 1 つのコンシューマのみ指定できることを意味します。TRUE は、表で作成されたキューにはメッセージごとにマルチ・コンシューマを指定できることを意味します。</p>
message_grouping	<p>このオプション・パラメータは、メッセージがグループ化されるかどうかを指定します。デフォルトの NONE は、各メッセージが個別に処理されることを意味します。TRANSACTIONAL は、1 つのトランザクションでエンキューされるすべてのメッセージが同じグループの一部とみなされ、関連するメッセージのグループとしてデキューされることを意味します。</p>
comment	<p>このオプション・パラメータは、ユーザーが指定したキュー表の説明です。このユーザー・コメントは、キュー・カタログに追加されます。</p>
auto_commit	<p>TRUE の場合、現行のトランザクションがある場合は、CREATE_QUEUE_TABLE 操作が実行される前にコミット</p>

パラメータ	説明
	トされます。CREATE_QUEUE_TABLE 操作は、コールから戻ると持続されます。これはデフォルトです。FALSE の場合、この操作は現行トランザクションの一部で、コール元がコミットを入力した場合のみ持続されます。 注意: このパラメータは非推奨となっています。
primary_instance	このオプション・パラメータは、キュー表の 1 次所有者を指定します。キュー・モニターのスケジューリングとキュー表内のキューの伝播は、このインスタンスで実行されます。デフォルト値の 0 は、キュー・モニターのスケジューリングと伝播が使用可能な任意のインスタンスで実行されることを意味します。 compatible が 8.1 以上の場合のみ、このパラメータを指定および変更できます。
secondary_instance	このオプション・パラメータは、プライマリ・インスタンスが使用できない場合のキュー表の所有者を指定します。デフォルト値の 0 は、キュー表が使用可能な任意のインスタンスにフェイルオーバーされることを意味します。 primary_instance も指定され、compatible が 8.1 以上の場合のみ、このパラメータを指定して変更できます。
compatible	このオプション・パラメータは、そのキュー表が互換性を持つ最も低いデータベース・バージョンを指定します。可能な値は、8.0、8.1 および 10.0 です。データベースが 10.1 互換モードの場合、デフォルト値は 10.0 です。データベースが 8.1 互換または 9.2 互換モードの場合、デフォルト値は 8.1 です。データベースが 8.0 互換モードの場合、デフォルト値は 8.0 です。値 8.0 は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になりました。 互換性の詳細は、 「Oracle Database Advanced Queuing の互換性パラメータ」 を参照してください。
secure	キュー表を保護キューに使用する場合は、このオプション・パラメータを TRUE に設定する必要があります。保護キューとは、エンキューやデキューなどのキュー操作を実行できる 1 人以上のデータベース・ユーザーに対して、AQ エージェントを明示的に関連付ける必要があるキューのことです。保護キューの所有者は、キューですべてのキュー操作を実行できますが、その他のユーザーは、保護キュー・ユーザーとして構成されないかぎりキュー操作を実行できません。
replication_mode	将来の使用のために予約されています。キューがレプリケーション・モードで作成されている場合は DBMS_AQADM.REPLICATION_MODE、それ以外の場合は DBMS_AQADM.NONE です。デフォルトは DBMS_AQADM.NONE です。

ペイロード・タイプ

ペイロード型をオブジェクト型として指定するには、オブジェクト型を定義する必要があります。



注意:

オブジェクト型のシノニムを作成している場合、それを DBMS_AQADM.CREATE_QUEUE_TABLE で使用することはできません。エラーORA-24015 が発生します。

[CLOB](#)、[BLOB](#)および[BFILE](#)オブジェクトはOracle Database Advanced Queuing[メッセージ](#)で有効です。これらのオブジェクト型は、Oracle8/リリース8.1.x以降のOracleソフトウェアを使用してOracle Database Advanced Queuing[伝播](#)で伝播できます。[LOBのオブジェクト型をエンキュー](#)するには、まずLOB_attributeをEMPTY_BLOB()に設定した後に、エンキューを実行する必要があります。その後、キュー表のビューから生成されたLOBロケータを選択して標準のLOB操作を使用できます。

注意:



LOBを含むペイロードの場合、エンキューおよびデキュー操作を実行するには、キュー表に対する明示的な Select、Insert および Update 権限を付与する必要があります。

STORAGE句

storage_clause引数には、標準のCREATE TABLE storage_clause引数に使用できるテキストを入力できます。

表領域が指定されたキュー表では、[索引構成表](#)(IOT)および索引は、すべてその表領域に作成されます。現在、これらを異なる表領域間に分割することはできません。

注意:

11g リリース 2 (11.2)の qmon プロセスでは、デキュー-IOT、履歴 IOT およびタイム・マネージャ IOT の自動結合を実行します。手動で AQ の IOT を結合する必要はありません。ただし、パフォーマンスの低下が見られる場合、回避策として実行することが可能です。



ローカル管理表領域にキュー表を作成するか、2 つ以上の空きリスト・グループがあるキュー表を作成する場合、キュー・モニター・コーディネータはそのブロックのクリーン・アップをスキップします。これにより、時間の経過とともにパフォーマンスが低下する可能性があります。

デキュー-IOT を結合するには次を実行します。

```
ALTER TABLE AQ$_queue_table_1 COALESCE;
```

このコマンドは、キューの同時デキュー元およびエンキュー元がある場合に実行できますが、コマンドの実行中に、同時ユーザーのパフォーマンスが若干低下する可能性があります。

ソート・キー

sort_listパラメータは、メッセージがデキューされる順序を決定します。キュー表を作成した後は、メッセージのソート順序を変更することはできません。選択肢は次のとおりです。

- ENQ_TIME
- ENQ_TIME, PRIORITY
- PRIORITY
- PRIORITY, ENQ_TIME

- PRIORITY, COMMIT_TIME
- COMMIT_TIME

COMMIT_TIMEを指定すると、キュー表を使用する任意のキューが[コミット時間キュー](#)になり、Oracle Database Advanced Queuingは、トランザクションをコミットするときに、エンキューされたメッセージごとに[近似CSCN](#)を計算します。

COMMIT_TIMEをソート・キーとして指定する場合、次の値も指定する必要があります。

- multiple_consumers = TRUE
- message_grouping = TRANSACTIONAL
- compatible = 8.1以上

コミット時間順序付けは、トランザクションが相互に依存している場合、またはキュー内のメッセージを参照するときに一貫した結果が必要な場合に便利です。

その他の表およびビュー

表の作成時に、次のオブジェクトが作成されます。

- AQ\$_queue_table_name: Oracle Database Advanced Queuingアプリケーションで[キュー](#)・データの間合せに使用される読み込み専用のビュー
- AQ\$_queue_table_name_E: キュー表に対応付けられたデフォルトの[例外キュー](#)
- AQ\$_queue_table_name_I: マルチ・[コンシューマ](#)・キューの場合にデキュー操作に使用される索引または[索引構成表](#)(IOT)
- AQ\$_queue_table_name_T: キュー・モニター操作の索引
- AQ\$_queue_table_name_L: コミットされたデキュー操作のメッセージ識別子のキューの格納に使用されるデキュー・ログ表

次のオブジェクトは、8.1互換のマルチ・コンシューマ・キュー表でのみ作成されます。

- AQ\$_queue_table_name_S: サブスクライバに関する情報の格納に使用される表
- AQ\$_queue_table_name_H: デキュー履歴データの格納に使用される索引構成表(IOT)

注意:



Oracle Database Advanced Queuing では、これらの内部 AQ のキュー表に対するトリガーの使用はサポートされていません。

スキーマを指定しない場合、デフォルトのスキーマはそのユーザーのスキーマになります。

GLOBAL_TOPIC_ENABLED = TRUEの場合は、キュー表が作成されると、対応する[Lightweight Directory Access Protocol](#)(LDAP)エントリも作成されます。

キュー・タイプがANYDATAの場合、[バッファ済キュー](#)と2つの追加オブジェクトが作成されます。バッファ済キューには、取得プロセスによって作成された論理変更レコードが格納されます。論理変更レコードがキューに関連付けられたメモリー・バッファにステージングされた場合、通常はディスクに書き込まれません。

デキューされずに一定の期間バッファ内でステージングされているか、すべての取得済イベントを保持するのに十分な空きがメモリーにない場合、論理変更レコードは次の表に流出します。

- AQ\$_queue_table_name_P: メモリーから流出した取得済イベントの格納に使用される表
- AQ\$_queue_table_name_D: 伝播に関する情報の格納と、各イベントの処理に適したプロセスの適用に使用される表

関連項目:

- [デキュー・モード](#)
- [『Oracle Database SecureFilesおよびラージ・オブジェクト開発者ガイド』](#)

例

次の例では、SQL*Plusテスト環境を使用していることを前提にしています。[例8-1](#)では、この章にある他の例の準備としてユーザーを作成します。この例では、管理者権限を持つユーザーとして接続する必要があります。この章にある他の例のほとんどでは、ユーザーtest_admとして接続できます。いくつかの例は、DBMS_AQADMのEXECUTE権限を使用してtestとして実行する必要があります。

例8-1 AQ管理ユーザーの設定

```
CREATE USER test_adm IDENTIFIED BY test_adm DEFAULT TABLESPACE example;
GRANT DBA, CREATE ANY TYPE TO test_adm;
GRANT EXECUTE ON DBMS_AQADM TO test_adm;
GRANT aq_administrator_role TO test_adm;
BEGIN
  DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE (
    privilege      =>  'MANAGE_ANY',
    grantee        =>  'test_adm',
    admin_option   =>  FALSE);
END;
/
CREATE USER test IDENTIFIED BY test;
GRANT EXECUTE ON dbms_aq TO test;
```

例8-2 AQ管理のサンプル・タイプの設定

```
CREATE TYPE test.message_typ AS object(
  sender_id      NUMBER,
  subject        VARCHAR2(30),
  text           VARCHAR2(1000));
/
CREATE TYPE test.msg_table AS TABLE OF test.message_typ;
/
CREATE TYPE test.order_typ AS object(
  custno        NUMBER,
  item          VARCHAR2(30),
  description   VARCHAR2(1000));
/
CREATE TYPE test.lob_typ AS object(
  id            NUMBER,
  subject       VARCHAR2(100),
  data          BLOB,
  trailer       NUMBER);
/
```

例8-3 オブジェクト型メッセージのキュー表の作成

```

BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.obj_qtab',
    queue_payload_type  => 'test.message_typ');
END;
/

```

例8-4 RAW型メッセージのキュー表の作成

```

BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.raw_qtab',
    queue_payload_type  => 'RAW');
END;
/

```

例8-5 LOB型メッセージのキュー表の作成

```

BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.lob_qtab',
    queue_payload_type  => 'test.lob_typ');
END;
/

```

例8-6 XMLTypeメッセージのキュー表の作成

```

BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.xml_qtab',
    queue_payload_type  => 'SYS.XMLType',
    multiple_consumers  => TRUE,
    compatible          => '8.1',
    comment              => 'Overseas Shipping multiconsumer orders queue table');
END;
/

```

例8-7 グループ化メッセージのキュー表の作成

```

BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.group_qtab',
    queue_payload_type  => 'test.message_typ',
    message_grouping    => DBMS_AQADM.TRANSACTIONAL);
END;
/

```

例8-8 優先メッセージおよびマルチ・コンシューマのキュー表の作成

```

BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.priority_qtab',
    queue_payload_type  => 'test.order_typ',
    sort_list            => 'PRIORITY, ENQ_TIME',
    multiple_consumers  => TRUE);
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.multiconsumer_qtab',
    queue_payload_type  => 'test.message_typ',
    sort_list            => 'PRIORITY, ENQ_TIME',
    multiple_consumers  => TRUE);
END;

```


/

例8-9 コミット時間順序付けを使用したキュー表の作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.commit_time_qtab',
    queue_payload_type  => 'test.message_typ',
    sort_list           => 'COMMIT_TIME',
    multiple_consumers  => TRUE,
    message_grouping    => DBMS_AQADM.TRANSACTIONAL,
    compatible          => '10.0');
END;
/
```

例8-10 マルチ・コンシューマの8.1互換キュー表の作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.multiconsumer_81_qtab',
    queue_payload_type  => 'test.message_typ',
    multiple_consumers  => TRUE,
    compatible          => '8.1');
END;
/
```

例8-11 指定した表領域に対するキュー表の作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.example_qtab',
    queue_payload_type  => 'test.message_typ',
    storage_clause      => 'tablespace example');
END;
/
```

例8-12 空きリストまたは空きリスト・グループがあるキュー表の作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE_TABLE(
    queue_table          => 'test.freelist_qtab',
    queue_payload_type  => 'RAW',
    storage_clause      => 'STORAGE (FREELISTS 4 FREELIST GROUPS 2)',
    compatible          => '8.1');
END;
/
```

キュー表の変更

DBMS_AQADM.ALTER_QUEUE_TABLEは、非シャード・キュー表の既存のプロパティを変更します。

```
DBMS_AQADM.ALTER_QUEUE_TABLE (
  queue_table          IN  VARCHAR2,
  comment              IN  VARCHAR2          DEFAULT NULL,
  primary_instance    IN  BINARY_INTEGER DEFAULT NULL,
  secondary_instance  IN  BINARY_INTEGER DEFAULT NULL,
  replication_mode    IN  BINARY_INTEGER DEFAULT NULL);
```

パラメータ	説明
queue_table	この必須パラメータは、キュー表名を指定します。
comment	このオプション・パラメータは、ユーザーが指定したキュー表の説明です。このユーザー・コメントは、キュー・カタログに追加されます。
primary_instance	このオプション・パラメータは、キュー表の 1 次所有者を指定します。キュー・モニターのスケジューリングとキュー表内のキューの伝播は、このインスタンスで実行されます。 compatible が 8.1 以上の場合のみ、このパラメータを指定および変更できます。
secondary_instance	このオプション・パラメータは、プライマリ・インスタンスが使用できない場合のキュー表の所有者を指定します。 primary_instance も指定され、compatible が 8.1 以上の場合のみ、このパラメータを指定して変更できます。
replication_mode	将来の使用のために予約されています。キューがレプリケーション・モードで変更されている場合は DBMS_AQADM.REPLICATION_MODE、それ以外の場合は DBMS_AQADM.NONE です。デフォルト値は NULL です。

注意:



一般に、キュー表では DDL 文はサポートされていないため、キュー表が操作不可能になる場合があります。たとえば、キュー表に対して ALTER TABLE ...SHRINK 文を発行すると、内部エラーが発生し、その後キュー表を使用しようとしても、常にエラーになります。キュー表では DDL 文を使用しないことをお勧めします。

GLOBAL_TOPIC_ENABLED = TRUE の場合は、キュー表が変更されると、対応する LDAP エントリも変更されます。

例8-13 プライマリおよびセカンダリ・インスタンスの変更によるキュー表の変更

```
BEGIN
  DBMS_AQADM.ALTER_QUEUE_TABLE (
    queue_table      => 'test.obj_qtab',
    primary_instance => 3,
    secondary_instance => 2);
END;
/
```

例8-14 コメントの変更によるキュー表の変更

```
BEGIN
  DBMS_AQADM.ALTER_QUEUE_TABLE (
    queue_table      => 'test.obj_qtab',
    comment          => 'revised usage for queue table');
END;
/
```

キュー表の削除

DBMS_AQADM.DROP_QUEUE_TABLEは、既存の非シャード・キュー表を削除します。

```
DBMS_AQADM.DROP_QUEUE_TABLE (  
    queue_table      IN    VARCHAR2,  
    force            IN    BOOLEAN DEFAULT FALSE,
```

キュー表を削除する前に、そのキュー表内のすべてのキューを停止して削除する必要があります。forceがFALSEに設定されている場合、明示的に実行する必要があります。forceがTRUEに設定されている場合、キュー表内のすべてのキューと関連付けられた伝播スケジュールが自動的に削除されます。

GLOBAL_TOPIC_ENABLED = TRUEの場合は、キュー表が削除されると、対応するLDAPエントリも削除されます。

例8-15 キュー表の削除

```
BEGIN  
    DBMS_AQADM.DROP_QUEUE_TABLE (  
        queue_table      => 'test.obj_qtab');  
END;  
/
```

例8-16 forceオプションを使用したキューの削除

```
BEGIN  
    DBMS_AQADM.DROP_QUEUE_TABLE (  
        queue_table      => 'test.raw_qtab',  
        force            => TRUE);  
END;  
/
```

キュー表のパージ

DBMS_AQADM.PURGE_QUEUE_TABLEは、非シャード・キュー表からメッセージをパージします。

```
DBMS_AQADM.PURGE_QUEUE_TABLE (  
    queue_table      IN    VARCHAR2,  
    purge_condition  IN    VARCHAR2,  
    purge_options    IN    aq$_purge_options_t);
```

次のパラメータがあります。

パラメータ	説明
queue_table	この必須パラメータは、キュー表名を指定します。
purge_condition	パージ条件には、大文字と小文字を区別する SQL の WHERE 句を使用する必要があります。条件は aq\$queue_table_name ビューの列に基づいています。パージ条件の各列名には、「qtview」という接頭辞が付きます。 永続メッセージでサポートされるパージ条件はすべて、バッファ済メッセージでもサポートされます。 キュー表のすべてのキューをパージするには、purge_condition を NULL(引用符を使用し

パラメータ	説明
	ない)または' (一重引用符 2 つ)に設定します。
purge_options	<p>aq\$_purge_options_t 型には block パラメータが含まれます。block が TRUE の場合、キュー表をパーズする間、キュー表内のすべてのキューで排他的ロックが保持されます。これにより、キュー表のパーズ時にエンキュー実行者とデキュー実行者が同時にブロックされます。block が TRUE の場合、パーズ・コールは常に成功します。block のデフォルトは FALSE です。この場合、エンキュー実行者とデキュー実行者はブロックされませんが、同時実行回数が多いと、パーズがエラーによって失敗する可能性があります。</p> <p>aq\$_purge_options_t 型には delivery_mode パラメータも含まれます。デフォルトの PERSISTENT の場合は、永続メッセージのみがパーズされます。BUFFERED に設定されている場合は、バッファ済メッセージのみがパーズされます。PERSISTENT_OR_BUFFERED に設定されている場合は、両方がパーズされます。</p>

このプロシージャを実行すると、トレース・ファイルがudumpの宛先に生成されます。プロシージャの実行内容の詳細が示されます。このプロシージャはすべてのメッセージの処理後にコミットされます。

関連項目:

DBMS_AQADM.PURGE_QUEUE_TABLEの詳細は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_AQADMに関する項を参照してください。

注意:



[例 8-20](#) の consumer_name、[例 8-21](#) の sender_name など、一部のパーズ条件は、8.1 互換のキュー表でのみサポートされています。詳細については、[表 9-1](#) を参照してください。

例8-17 キュー表のすべてのメッセージのパーズ

```

DECLARE
po dbms_aqadm.aq$_purge_options_t;
BEGIN
  po.block := FALSE;
  DBMS_AQADM.PURGE_QUEUE_TABLE (
    queue_table    => 'test.obj_qtab',
    purge_condition => NULL,
    purge_options  => po);
END;
/

```

例8-18 指定したキューのすべてのメッセージのパーズ

```

DECLARE
po dbms_aqadm.aq$_purge_options_t;
BEGIN
  po.block := TRUE;
  DBMS_AQADM.PURGE_QUEUE_TABLE (
    queue_table    => 'test.obj_qtab',

```

```

    purge_condition => 'qtvview.queue = '' TEST.OBJ_QUEUE''',
    purge_options   => po);
END;
/

```

例8-19 指定したキューのすべてのPROCESSEDメッセージのパージ

```

DECLARE
po dbms_aqadm.aq$_purge_options_t;
BEGIN
    po.block := TRUE;
    DBMS_AQADM.PURGE_QUEUE_TABLE(
        queue_table      => 'test.obj_qtab',
        purge_condition => 'qtvview.queue = '' TEST.OBJ_QUEUE''
                           and qtvview.msg_state = '' PROCESSED''',
        purge_options    => po);
END;
/

```

例8-20 指定したキューと指定したコンシューマのすべてのメッセージのパージ

```

DECLARE
po dbms_aqadm.aq$_purge_options_t;
BEGIN
    po.block := TRUE;
    DBMS_AQADM.PURGE_QUEUE_TABLE(
        queue_table      => 'test.multiconsumer_81_qtab',
        purge_condition => 'qtvview.queue = '' TEST.MULTICONSUMER_81_QUEUE''
                           and qtvview.consumer_name = '' PAYROLL_APP''',
        purge_options    => po);
END;
/

```

例8-21 指定した送信者からのすべてのメッセージの削除

```

DECLARE
po dbms_aqadm.aq$_purge_options_t;
BEGIN
    po.block := TRUE;
    DBMS_AQADM.PURGE_QUEUE_TABLE(
        queue_table      => 'test.multiconsumer_81_qtab',
        purge_condition => 'qtvview.sender_name = '' TEST.OBJ_QUEUE''',
        purge_options    => po);
END;
/

```

キュー表の移行

DBMS_AQADM.MIGRATE_QUEUE_TABLEは、非シャード・キュー表を8.0、8.1または10.0から8.0、8.1または10.0に移行します。移行を実行できるのは、キュー表の所有者のみです。

```

DBMS_AQADM.MIGRATE_QUEUE_TABLE(
    queue_table  IN  VARCHAR2,
    compatible  IN  VARCHAR2);

```



注意:

このプロシージャでは、キュー表の所有者(通常のキュー・ユーザー)に DBMS_AQADM の EXECUTE 権限を付与する必要があります。通常のキュー・ユーザーに対してキューおよびキュー表の作成と削除、サブスクライバの追加と削除などを許可しない場合は、移行完了の直後に EXECUTE 権限を取り消す必要があります。

注意:



compatible が 8.0 に設定されているキュー表で作成されたキュー(このマニュアルでは 8.0 形式のキューと呼びます)は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になっています。したがって、新しいキューの作成には 8.1 以降の形式を使用し、既存の 8.0 形式のキューをなるべく早く移行することをお勧めします。

スキーマが下位リリースからエクスポートされたダンプのインポートによって作成された場合、またはスキーマに下位リリースからアップグレードされた Oracle Database Advanced Queuing キューがある場合は、DROP USER CASCADE を使用してスキーマを削除しようとすると、ORA-24005 エラーが発生します。このようなスキーマを削除するには、次の手順に従います。

1. イベント 10851 をレベル 1 に設定します。
2. AQ\$_queue_table_name_NR という形式のすべての表をスキーマから削除します。
3. イベント 10851 をオフにします。
4. スキーマを削除します。

例 8-22 8.1 互換から 10.0 互換へのキュー表のアップグレード

```
BEGIN
  DBMS_AQADM.MIGRATE_QUEUE_TABLE (
    queue_table => 'test.xml_qtab',
    compatible  => '10.0');
END;
/
```

非シャード・キューの管理

次のトピックでは、非シャード・キューを管理する方法について説明します。

注意:



シャード・キューの開始および停止では、非シャード・キューと同じ API を使用します。

- [キューの作成](#)
- [キューの変更](#)
- [キューの開始](#)
- [キューの停止](#)
- [キューの削除](#)

キューの作成

DBMS_AQADM.CREATE_QUEUEは非シャード・キューを作成します。

```
DBMS_AQADM.CREATE_QUEUE (
  queue_name      IN      VARCHAR2,
  queue_table     IN      VARCHAR2,
  queue_type      IN      BINARY_INTEGER DEFAULT NORMAL_QUEUE,
  max_retries     IN      NUMBER          DEFAULT NULL,
  retry_delay     IN      NUMBER          DEFAULT 0,
  retention_time  IN      NUMBER          DEFAULT 0,
  dependency_tracking IN  BOOLEAN        DEFAULT FALSE,
  comment        IN      VARCHAR2        DEFAULT NULL,
```

次のパラメータがあります。

パラメータ	説明
queue_name	<p>この必須パラメータは、新しいキューの名前を指定します。</p> <p>データベースの互換性が 10.0 である場合のみ、大文字と小文字が混在するキュー名を指定できますが、名前を二重引用符で囲む必要があります。したがって、abc.efg は、スキーマが ABC で名前が EFG であることを示しますが、“abc”.“efg”はスキーマが abc で名前が efg であることを示します。</p> <p>12c リリース 2 (12.2)以降では、ユーザーが生成するキュー名の最大長は 122 バイトです。長い名前のキューを作成しようとすると、エラーORA-24019 が発生します。[その他の表およびキュー]にリストされているキュー名など、Oracle Database Advanced Queuing で生成されたキュー名は、128 文字より長くできません。</p>
queue_table	<p>この必須パラメータは、キューが作成されるキュー表を指定します。</p>
queue_type	<p>このパラメータは、作成されるキューのタイプを指定します。デフォルトの NORMAL_QUEUE は、通常のキューを作成します。EXCEPTION_QUEUE は、例外キューを生成します。</p>
max_retries	<p>このパラメータは、メッセージに対する REMOVE モードのデキューの試行回数を制限します。max_retries の最大値は 2**31 -1 です。</p>
retry_delay	<p>このパラメータは、アプリケーションのロールバック後、スケジュールされたメッセージの再処理が行われるまでの時間を秒単位で指定します。デフォルトは 0(ゼロ)で、メッセージを最も迅速に取り出せます。このパラメータは、max_retries が 0(ゼロ)に設定されている場合は無効です。</p> <p>このパラメータは、シングル・コンシューマ・キューおよび 8.1 形式以上のマルチ・コンシューマ・キューではサポートされていますが、8.0 形式のマルチ・コンシューマ・キュー(Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨)ではサポートされていません。</p>

パラメータ	説明
retention_time	このパラメータは、キューからデキューされた後、メッセージがキュー表内に保持される秒数を指定します。retention_time が期限切れになると、メッセージはタイム・マネージャ・プロセスによって削除されます。INFINITE は、メッセージが永久に保持されることを意味します。デフォルトは 0(ゼロ)で、保持されません。
dependency_tracking	このパラメータは、将来の使用に備えて確保されています。FALSE がデフォルトです。このリリースでは、TRUE は使用できません。
comment	このオプション・パラメータは、ユーザーが指定したキューの説明です。このユーザー・コメントは、キュー・カタログに追加されます。

すべてのキュー名は、[スキーマ](#)内において一意である必要があります。キューは、CREATE_QUEUEで作成した後、START_QUEUEをコールすると有効になります。デフォルトでは、キューはエンキューおよびデキューともに使用禁止で作成されます。保存されているメッセージを参照するには、メッセージIDによってデキューするか、またはSQLを使用します。GLOBAL_TOPIC_ENABLED = TRUE の場合は、キューが作成されると、対応するLDAPエントリも作成されます。

次の例([例8-23](#)から[例8-30](#))は、[例8-1](#)から[例8-12](#)で作成したデータ構造を使用しています。

例8-23 オブジェクト型メッセージのキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.obj_queue',
    queue_table     => 'test.obj_qtab');
END;
/
```

例8-24 RAW型メッセージのキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.raw_queue',
    queue_table     => 'test.raw_qtab');
END;
/
```

例8-25 LOB型メッセージのキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.lob_queue',
    queue_table     => 'test.lob_qtab');
END;
/
```

例8-26 グループ化メッセージのキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.group_queue',
    queue_table     => 'test.group_qtab');
END;
/
```


例8-27 優先メッセージのキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.priority_queue',
    queue_table     => 'test.priority_qtab');
END;
/
```

例8-28 優先メッセージおよびマルチ・コンシューマのキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.multiconsumer_queue',
    queue_table     => 'test.multiconsumer_qtab');
END;
/
```

例8-29 伝播のデモ用のキューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.another_queue',
    queue_table     => 'test.multiconsumer_qtab');
END;
/
```

例8-30 マルチ・コンシューマの8.1形式キューの作成

```
BEGIN
  DBMS_AQADM.CREATE_QUEUE (
    queue_name      => 'test.multiconsumer_81_queue',
    queue_table     => 'test.multiconsumer_81_qtab');
END;
/
```

キューの変更

DBMS_AQADM.ALTER_QUEUEは、非シャード・キューの既存のプロパティを変更します。

```
DBMS_AQADM.ALTER_QUEUE (
  queue_name      IN   VARCHAR2,
  max_retries     IN   NUMBER   DEFAULT NULL,
  retry_delay     IN   NUMBER   DEFAULT NULL,
  retention_time  IN   NUMBER   DEFAULT NULL,
  comment         IN   VARCHAR2 DEFAULT NULL);
```

max_retries、comment、retry_delayおよびretention_timeのみ変更できます。保存されているメッセージを参照するには、メッセージIDによってデキューするか、またはSQLを使用します。GLOBAL_TOPIC_ENABLED = TRUEの場合は、キューが変更されると、対応するLDAPエントリも変更されます。

次の例は、保存時間を変更して、デキュー後1日間メッセージを保存します。

例8-31 保存時間の変更によるキューの変更

```
BEGIN
  DBMS_AQADM.ALTER_QUEUE (
    queue_name      => 'test.another_queue',
    retention_time  => 86400);
END;
```

キューの開始

DBMS_AQADM. START_QUEUEは、指定した非シャード・キューでのエンキューまたはデキューを有効にします。

```
DBMS_AQADM. START_QUEUE (
  queue_name      IN      VARCHAR2,
  enqueue         IN      BOOLEAN DEFAULT TRUE,
  dequeue         IN      BOOLEAN DEFAULT TRUE);
```

管理者は、キューを作成した後、START_QUEUEを使用してそのキューを有効にする必要があります。デフォルトでは、エンキューおよびデキューの両方が有効になります。例外キューでは、デキュー操作のみ許可されます。この操作は、コールが完了し、コールに[トランザクション](#)の特性がない場合にのみ有効になります。

例8-32 エンキューおよびデキューの両方を有効にしたキューの開始

```
BEGIN
  DBMS_AQADM. START_QUEUE (
    queue_name      => 'test.obj_queue');
END;
/
```

例8-33 デキューのみのキューの開始

```
BEGIN
  DBMS_AQADM. START_QUEUE (
    queue_name      => 'test.raw_queue',
    dequeue         => TRUE,
    enqueue         => FALSE);
END;
/
```

キューの停止

DBMS_AQADM. STOP_QUEUEは、指定した非シャード・キューでのエンキュー、デキューまたはその両方を無効にします。

```
DBMS_AQADM. STOP_QUEUE (
  queue_name      IN      VARCHAR2,
  enqueue         IN      BOOLEAN DEFAULT TRUE,
  dequeue         IN      BOOLEAN DEFAULT TRUE,
  wait            IN      BOOLEAN DEFAULT TRUE);
```

デフォルトでは、このコールによってエンキューおよびデキューの両方が無効になります。キューに対する未処理のトランザクションがある場合、キューは停止できません。この操作はコールが完了すると有効になり、トランザクションの特性はありません。

例8-34 キューの停止

```
BEGIN
  DBMS_AQADM. STOP_QUEUE (
    queue_name      => 'test.obj_queue');
END;
/
```

キューの削除

このプロシージャは、既存の非シャード・キューを削除します。あらかじめSTOP_QUEUEがコールされ、キューがエンキューおよびデ

キューの両方に対して無効にされていないかぎり、DROP_QUEUEは許可されません。すべてのキュー・データが、削除操作の一部として削除されます。

```
DBMS_AQADM.DROP_QUEUE(  
  queue_name      IN      VARCHAR2,
```

GLOBAL_TOPIC_ENABLED = TRUEの場合は、キューが削除されると、対応するLDAPエントリも削除されます。

例8-35 標準キューの削除

```
BEGIN  
  DBMS_AQADM.DROP_QUEUE(  
    queue_name      => 'test.obj_queue');  
END;  
/
```

シャード・キューの管理

次のトピックでは、シャード・キューを管理する方法について説明します。

注意:



シャード・キューの開始および停止では、非シャード・キューと同じ API を使用します。

- [シャード・キューの作成](#)
- [シャード・キューの削除](#)
- [シャード・キューの変更](#)
- [キューのパラメータの設定](#)
- [キューのパラメータの設定解除](#)
- [キューのパラメータの取得](#)
- [例外キューの作成](#)

シャード・キューの作成

CREATE_SHARDED_QUEUE APIはシャード・キューを作成します。

```
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);
```

次のパラメータがあります。

パラメータ	説明
-------	----

パラメータ	説明
queue_name	この必須パラメータは、新しいキューの名前を指定します。最大で 128 文字まで使用できます。
storage_clause	<p>記憶域パラメータは、キュー表の作成時に、CREATE TABLE 文に組み込まれます。storage_clause 引数には、標準の CREATE TABLE storage_clause 引数に使用できるテキストを入力できます。記憶域パラメータは次のパラメータの任意の組合せで構成できます: PCTFREE、PCTUSED、INITRANS、MAXTRANS、TABLESPACE、LOB および表の STORAGE 句。</p> <p>ここで表領域が指定されない場合は、キュー表とそのすべての関連オブジェクトが、デフォルトのユーザー表領域に作成されます。ここで表領域が指定されると、キュー表とそのすべての関連オブジェクトが、STORAGE 句で指定された表領域に作成されます。これらのパラメータの使用方法は、『Oracle Database SQL 言語リファレンス』を参照してください。</p>
multiple_consumers	FALSE の場合、各メッセージに対してコンシューマを 1 つのみ設定できます。これはデフォルトです。TRUE は、表で作成されたキューにはメッセージごとにマルチ・コンシューマを指定できることを意味します。
max_retries	このオプション・パラメータは、メッセージのデキューに失敗した後に再試行できる回数を制限します。max_retries の最大値は $2^{31} - 1$ です。再試行の制限を超えると、メッセージがキューからパージされます。デキューの実行後にアプリケーションがロールバックを発行するたびに、RETRY_COUNT が増加します。サーバー・プロセスの停止 (ALTER SYSTEM KILL SESSION) またはインスタンスの SHUTDOWN ABORT によって、デキュー・トランザクションが失敗した場合は、RETRY_COUNT は増加しません。
comment	このオプション・パラメータは、ユーザーが指定したキュー表の説明です。このユーザー・コメントは、キュー・カタログに追加されます。
queue_payload_type	ペイロードは、RAW、DBMS_AQADM.JMS_TYPE またはオブジェクト・タイプにすることができます。デフォルトは DBMS_AQADM.JMS_TYPE です。
queue_properties	<p>通常キューまたは例外キュー、再試行の遅延、保持時間、ソート・リスト、キャッシュ・ヒントなどのプロパティ。</p> <p>queue_properties の詳細は、『Oracle Database PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。</p>
replication_mode	将来の使用のために予約されています。キューがレプリケーション・モードで作成されている場合は DBMS_AQADM.REPLICATION_MODE、それ以外の場合は

パラメータ	説明
	DBMS_AQADM. NONE です。デフォルトは DBMS_AQADM. NONE です。

シャード・キューの削除

このプロシージャは、データベース・キューイング・システムから既存の共有されているキューを削除します。DROP_SHARDED_QUEUE をコールする前に、キューを停止する必要があります。forceをFALSEに設定した場合、ユーザーはDROP_SHARDED_QUEUEをコールする前にキューを明示的に停止する必要があります。forceをTRUEに設定した場合、キューは内部的に停止されてから削除されます。

構文

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

パラメータ

表8-1 CREATE_SHARDED_QUEUEプロシージャのパラメータ

パラメータ	説明
queue_name	この必須パラメータは、共有されているキューの名前を指定します。
force	キューが停止されていない場合でも、共有されているキューは削除されます。

シャード・キューの変更

このプロシージャを使用すると、ユーザーはシャード・キューのqueue_propertiesを変更できます。

構文

```
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);
```

パラメータ

表8-2 ALTER_SHARDED_QUEUEプロシージャのパラメータ

パラメータ	説明
queue_name	このパラメータは、シャード・キューの名前を指定します。最大で 128 文字まで使用できます。
max_retries	許可される最大試行数。

パラメータ	説明
comment	パラメータのコメント。
queue_properties	通常キューまたは例外キュー、再試行の遅延、保持時間、ソート・リスト、キャッシュ・ヒントなどのプロパティ。 queue_propertiesの詳細は、 『Oracle Database PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』 を参照してください。
replication_mode	将来の使用のために予約されています。キューがレプリケーション・モードで変更されている場合は DBMS_AQADM.REPLICATION_MODE、それ以外の場合は DBMS_AQADM.NONE です。デフォルトは NULL です。

キューのパラメータの設定

このプロシージャを使用すると、ユーザーはキュー・レベルまたはデータベース・レベルでシャード・キューに各種のパラメータを設定できます。データベース・レベルの場合、queue_nameはNULLである必要があります。キューによってデータベース・レベルのパラメータ値が上書きされます。

構文

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

パラメータ

表8-3 SET_QUEUE_PARAMETERプロシージャのパラメータ

パラメータ	説明
queue_name	シャード・キューの名前。
param_name	パラメータ名。
param_value	パラメータの値。

キューのパラメータの設定解除

このプロシージャを使用すると、ユーザーはキュー・レベルまたはデータベース・レベルでシャード・キューの各種のパラメータを設定解除できます。データベース・レベルの場合、queue_nameはNULLである必要があります。キューによってデータベース・レベルのパラメータ値が上書きされます。

構文

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

パラメータ

表8-4 UNSET_QUEUE_PARAMETERプロシージャのパラメータ

パラメータ	説明
queue_name	シャード・キューの名前。
param_name	パラメータ名。

キューのパラメータの取得

このプロシージャを使用すると、ユーザーはキュー・レベルまたはデータベース・レベルでシャード・キューの各種のパラメータを取得できます。データベース・レベルの場合、queue_nameはNULLである必要があります。キューによってデータベース・レベルのパラメータ値が上書きされます。

構文

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

パラメータ

表8-5 GET_QUEUE_PARAMETERプロシージャのパラメータ

パラメータ	説明
queue_name	シャード・キューの名前。
param_name	パラメータ名。
param_value	パラメータの値。

例外キューの作成

このプロシージャを使用すると、ユーザーはシャード・キューの例外キューを作成できます。

構文

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

パラメータ

表8-6 CREATE_EXCEPTION_QUEUEプロシージャのパラメータ

パラメータ	説明
-------	----

パラメータ	説明
sharded_queue_name	シャード・キューの名前。
exception_queue_name	例外キューの名前を指定します。

変換の管理

変換はメッセージのフォーマットを変更するため、あるアプリケーションによって作成されたメッセージを、別のアプリケーションで認識できます。永続メッセージとバッファ済メッセージの両方で変換を使用できます。次のトピックでは、キュー表を管理する方法について説明します。

- [変換の作成](#)
- [変換の変更](#)
- [変換の削除](#)

注意:



シャード・キューは変換をサポートしていません。

変換の作成

DBMS_TRANSFORM.CREATE_TRANSFORMATIONは、メッセージ・フォーマットの変換を作成します。

```
DBMS_TRANSFORM.CREATE_TRANSFORMATION (
  schema          VARCHAR2 (30),
  name            VARCHAR2 (30),
  from_schema     VARCHAR2 (30),
  from_type       VARCHAR2 (30),
  to_schema       VARCHAR2 (30),
  to_type         VARCHAR2 (30),
  transformation  VARCHAR2 (4000));
```

[変換](#)は、入力の型がfrom_typeで、to_type型のオブジェクトを返すSQLファンクションである必要があります。また、to_type型でfrom_typeを参照するSQL式も使用できます。from_typeに対するすべての参照は、source.user_dataという形式である必要があります。

この機能を使用するには、dbms_transformに対するEXECUTE権限が必要です。この権限は、AQ_ADMINISTRATOR_ROLEに含まれています。

また、変換のソース・タイプおよび宛先タイプであるユーザー定義型に対するEXECUTE権限、および変換ファンクションで使用するすべてのPL/SQLファンクションに対するEXECUTE権限も必要です。変換では、データベースの状態の書込み([DML](#)操作の実行)、カレント・トランザクションのコミットまたはロールバックはできません。

例8-36 変換の作成

```
BEGIN
  DBMS_TRANSFORM.CREATE_TRANSFORMATION (
    schema          => 'test',
```



```

name          => 'message_order_transform',
from_schema   => 'test',
from_type     => 'message_typ',
to_schema     => 'test',
to_type       => 'order_typ',
transformation => 'test.order_typ(
    source.user_data.sender_id,
    source.user_data.subject,
    source.user_data.text)');
END;
/

```

関連項目:

管理者およびユーザー・ロールの詳細は、[「Oracle Database Advanced Queuingのセキュリティ」](#)を参照してください。

変換の変更

DBMS_TRANSFORM.MODIFY_TRANSFORMATIONは、変換ファンクションを変更して、ターゲット型の各属性ごとに変換を指定します。

```

DBMS_TRANSFORM.MODIFY_TRANSFORMATION (
    schema          VARCHAR2 (30),
    name            VARCHAR2 (30),
    attribute_number INTEGER,
    transformation  VARCHAR2 (4000));

```

属性の番号が0(ゼロ)に指定された場合、変換式は単純にソース・タイプからターゲット型への変換を定義します。

from_typeに対するすべての参照は、source.user_dataという形式である必要があります。ソース・タイプの属性に対するすべての参照には、source.user_dataという接頭辞が付きます。

この機能を使用するには、dbms_transformに対するEXECUTE権限が必要です。また、変換のソース・タイプおよび宛先タイプであるユーザー定義型に対するEXECUTE権限、および変換ファンクションで使用するすべてのPL/SQLファンクションに対するEXECUTE権限も必要です。

変換の削除

DBMS_TRANSFORM.DROP_TRANSFORMATIONは変換を削除します。

```

DBMS_TRANSFORM.DROP_TRANSFORMATION (
    schema          VARCHAR2 (30),
    name            VARCHAR2 (30));

```

この機能を使用するには、dbms_transformに対するEXECUTE権限が必要です。また、変換のソース・タイプおよび宛先タイプであるユーザー定義型に対するEXECUTE権限、および変換ファンクションで使用するすべてのPL/SQLファンクションに対するEXECUTE権限も必要です。

権限の付与および取消し

次のトピックでは、権限を付与および取消しする方法について説明します。

- [Oracle Database Advanced Queuingシステム権限の付与](#)

- [Oracle Database Advanced Queuingシステム権限の取消し](#)
- [キュー権限の付与](#)
- [キュー権限の取消し](#)

Oracle Database Advanced Queuingシステム権限の付与

DBMS_AQADM.GRANT_SYSTEM_PRIVILEGEは、ユーザーおよびロールにOracle Database Advanced Queuingシステム権限を付与します。権限は、ENQUEUE_ANY、DEQUEUE_ANYおよびMANAGE_ANYです。最初は、SYSおよびSYSTEMのみがこのプロシージャを正常に使用できます。

```
DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE (
  privilege      IN   VARCHAR2,
  grantee        IN   VARCHAR2,
  admin_option   IN   BOOLEAN := FALSE);
```

ENQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューにメッセージをエンキューできます。DEQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューからメッセージをデキューできます。MANAGE_ANY権限を付与されたユーザーは、データベースのすべてのスキーマに対してDBMS_AQADMコールを実行できます。

注意:



Oracle Database 12c リリース 2 以降、SYS 以外のユーザーが MANAGE_ANY、ENQUEUE_ANY および DEQUEUE_ANY 権限で SYS が所有するキューにアクセスすることができなくなりました。

例8-37 AQシステム権限の付与

```
BEGIN
  DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE (
    privilege      => 'ENQUEUE_ANY',
    grantee        => 'test',
    admin_option   => FALSE);
  DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE (
    privilege      => 'DEQUEUE_ANY',
    grantee        => 'test',
    admin_option   => FALSE);
END;
```

Oracle Database Advanced Queuingシステム権限の取消し

DBMS_AQADM.REVOKE_SYSTEM_PRIVILEGEは、ユーザーおよびロールからOracle Database Advanced Queuingシステム権限を取り消します。権限は、ENQUEUE_ANY、DEQUEUE_ANYおよびMANAGE_ANYです。

```
DBMS_AQADM.REVOKE_SYSTEM_PRIVILEGE (
  privilege      IN   VARCHAR2,
  grantee        IN   VARCHAR2);
```

システム権限のADMINオプションを選択的に取り消すことはできません。

ENQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューにメッセージをエンキューできます。DEQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューからメッセージをデキューできます。MANAGE_ANY権限を付与されたユーザーは、データベースのすべてのスキーマに対してDBMS_AQADMコールを実行できます。

注意:



Oracle Database 12c リリース 2 以降、SYS 以外のユーザーが `MANAGE_ANY`、`ENQUEUE_ANY` および `DEQUEUE_ANY` 権限で SYS が所有するキューにアクセスすることができなくなりました。

例8-38 AQシステム権限の取消し

```
BEGIN
  DBMS_AQADM.REVOKE_SYSTEM_PRIVILEGE (
    privilege      => 'DEQUEUE_ANY',
    grantee        => 'test');
END;
/
```

キュー権限の付与

`DBMS_AQADM.GRANT_QUEUE_PRIVILEGE`は、ユーザーおよびロールにキューの権限を付与します。権限は、`ENQUEUE`、`DEQUEUE` または `ALL` です。初期設定では、キュー表の所有者のみがこのプロシージャを使用してそのキューの権限を付与できます。

```
DBMS_AQADM.GRANT_QUEUE_PRIVILEGE (
  privilege      IN   VARCHAR2,
  queue_name     IN   VARCHAR2,
  grantee       IN   VARCHAR2,
  grant_option   IN   BOOLEAN := FALSE);
```

注意:



このプロシージャでは、キュー表の所有者(通常のキュー・ユーザー)に `DBMS_AQADM` の `EXECUTE` 権限を付与する必要があります。通常のキュー・ユーザーに対してキューおよびキュー表の作成と削除、サブスクライバの追加と削除などを許可しない場合は、初期の `GRANT_QUEUE_PRIVILEGE` の完了直後に `EXECUTE` 権限を取り消す必要があります。

例8-39 キュー権限の付与

```
BEGIN
  DBMS_AQADM.GRANT_QUEUE_PRIVILEGE (
    privilege      => 'ALL',
    queue_name     => 'test.multiconsumer_81_queue',
    grantee       => 'test_adm',
    grant_option   => TRUE);
END;
/
```

キュー権限の取消し

`DBMS_AQADM.REVOKE_QUEUE_PRIVILEGE`は、ユーザーおよびロールからキューの権限を取り消します。この権限とは、`ENQUEUE` または `DEQUEUE` です。

```
DBMS_AQADM.REVOKE_QUEUE_PRIVILEGE (
  privilege      IN   VARCHAR2,
  queue_name     IN   VARCHAR2,
  grantee       IN   VARCHAR2);
```

権限を取り消すには、取消し実行者がその権限の付与者である必要があります。GRANTオプションによって伝播された権限は、付与者の権限が取り消されたときに取り消されます。

[例8-40](#)に示すように、権限受領者から特定のキューに関するデキュー権限を取り消して、エンキュー権限のみを付与しておくことができます。

例8-40 デキュー権限の取消し

```
BEGIN
  DBMS_AQADM.REVOKE_QUEUE_PRIVILEGE (
    privilege => 'DEQUEUE',
    queue_name => 'test.multiconsumer_81_queue',
    grantee => 'test_adm');
END;
```

サブスクライバの管理

次のトピックでは、サブスクライバを管理する方法について説明します。

- [サブスクライバの追加](#)
- [サブスクライバの変更](#)
- [サブスクライバの削除](#)

サブスクライバの追加

DBMS_AQADM.ADD_SUBSCRIBERは、キューにデフォルトのサブスクライバを追加します。

```
DBMS_AQADM.ADD_SUBSCRIBER (
  queue_name IN VARCHAR2,
  subscriber IN sys.aq$_agent,
  rule IN VARCHAR2 DEFAULT NULL,
  transformation IN VARCHAR2 DEFAULT NULL,
  queue_to_queue IN BOOLEAN DEFAULT FALSE,
  delivery_mode IN PLS_INTEGER DEFAULT PERSISTENT);
```

アプリケーションは、特定の受信者リストまたはデフォルトの[サブスクライバ](#)・リストに対してメッセージをエンキューできます。この操作は、マルチ・コンシューマが許可されているキューでのみ成功し、サブスクライバの合計は1024以下である必要があります。この操作はすぐに有効になり、この操作を含むトランザクションはコミットされます。このコール後に実行されたエンキュー要求は、新しいアクションを反映します。rule内のすべての文字列は、次のように、一重引用符で囲まれている必要があります。

```
rule => 'PRIORITY <= 3 AND CORRID = '' FROM JAPAN'''
```

ユーザー・データ・プロパティまたは属性は、オブジェクト・ペイロードのみに適用され、常に接頭辞としてtab.userdataを付ける必要があります。

GLOBAL_TOPIC_ENABLEDがTRUEの場合は、サブスクライバが作成されると、対応するLDAPエントリも作成されます。

デキューまたは伝播中に適用する変換の名前を指定します。変換は、DBMS_TRANSFORMパッケージを使用して作成します。

XMLType属性を持つペイロードを含むキューに対しては、XMLType.existsNode() メソッドおよびXMLType.extract() メソッドを含むルールを指定できます。

パラメータqueue_to_queueがTRUEに設定されている場合、追加されたサブスクライバは、キューからキューへのサブスクライバです。ソース・キューと宛先キューの間にキューからキューへの伝播が設定されている場合、キューからキューへのサブスクライバは、その伝播スケジュールによってメッセージを受信します。

delivery_modeパラメータがデフォルトのPERSISTENTの場合、サブスクライバは永続メッセージのみを受信します。BUFFEREDに設定されている場合、サブスクライバはバッファ済メッセージのみを受信します。PERSISTENT_OR_BUFFEREDに設定されている場合、サブスクライバは両方のタイプを受信します。ALTER_SUBSCRIBERを使用してこのパラメータを変更することはできません。

宛先キューがシングル・コンシューマ・キューの場合、エージェント名は、NULLである必要があります。

注意:



ADD_SUBSCRIBER はキューに対する管理操作です。Oracle Database AQ では、アプリケーションによる管理コールおよび操作コールの同時発行が可能です。ADD_SUBSCRIBER は、メッセージをエンキューまたはデキューしている保留中のコールが完了するまでブロックします。保留中のトランザクションの完了は待機しません。

関連項目:

- DBMS_TRANSFORMパッケージの詳細は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)を参照してください。
- [キューの伝播のスケジューリング](#)

例8-41 データベース・リンクの指定されたキューでのサブスクライバの追加

```
DECLARE
  subscriber      sys.aq$_agent;
BEGIN
  subscriber := sys.aq$_agent('subscriber1', 'test2.msg_queue2@london', null);
  DBMS_AQADM.ADD_SUBSCRIBER(
    queue_name     => 'test.multiconsumer_81_queue',
    subscriber     => subscriber);
END;
/
```

例8-42 データベース・リンクへのサブスクライバとしてのシングル・コンシューマ・キューの追加

```
DECLARE
  subscriber      sys.aq$_agent;
BEGIN
  subscriber := sys.aq$_agent('subscriber1', 'test2.msg_queue2@london', null);
  DBMS_AQADM.ADD_SUBSCRIBER(
    queue_name     => 'test.multiconsumer_81_queue',
    subscriber     => subscriber);
END;
/
```

例8-43 ルールを使用したサブスクライバの追加

```
DECLARE
  subscriber      sys.aq$_agent;
BEGIN
  subscriber := sys.aq$_agent('subscriber2', 'test2.msg_queue2@london', null);
  DBMS_AQADM.ADD_SUBSCRIBER(
    queue_name => 'test.multiconsumer_81_queue',
    subscriber => subscriber,
```

```

    rule      => 'priority < 2');
END;
/

```

例8-44 サブスクライバの追加と変換の指定

```

DECLARE
    subscriber      sys.aq$_agent;
BEGIN
    subscriber := sys.aq$_agent('subscriber3', 'test2.msg_queue2@london', null);
    DBMS_AQADM.ADD_SUBSCRIBER(
        queue_name    => 'test.multiconsumer_81_queue',
        subscriber    => subscriber,
        transformation => 'test.message_order_transform');
END;
/

```

例8-45 マルチ・コンシューマ・キューからシングル・コンシューマ・キューへの伝播

```

DECLARE
    subscriber      SYS.AQ$_AGENT;
BEGIN
    subscriber := SYS.AQ$_AGENT(NULL, 'test2.single_consumer__queue@london',
null);
    DBMS_AQADM.ADD_SUBSCRIBER(
        queue_name    => 'test.multiconsumer_81_queue',
        subscriber    => subscriber);
END;

```

サブスクライバの変更

DBMS_AQADM.ALTER_SUBSCRIBERは、指定したキューに対するサブスクライバの既存プロパティを変更します。

```

DBMS_AQADM.ALTER_SUBSCRIBER (
    queue_name      IN    VARCHAR2,
    subscriber      IN    sys.aq$_agent,
    rule            IN    VARCHAR2
    transformation  IN    VARCHAR2);

```

ルールまたは変換(あるいはその両方)を変更できます。これらの属性のいずれか1つのみを変更する場合は、他の属性の既存の値を変更コールに指定します。GLOBAL_TOPIC_ENABLED = TRUEの場合は、サブスクライバが変更されると、対応するLDAPエントリが作成されます。

例8-46 サブスクライバ・ルールの変更

```

DECLARE
    subscriber      sys.aq$_agent;
BEGIN
    subscriber := sys.aq$_agent('subscriber2', 'test2.msg_queue2@london', null);
    DBMS_AQADM.ALTER_SUBSCRIBER(
        queue_name => 'test.multiconsumer_81_queue',
        subscriber => subscriber,
        rule      => 'priority = 1');
END;
/

```

サブスクライバの削除

DBMS_AQADM.REMOVE_SUBSCRIBERは、デフォルトのサブスクライバをキューから削除します。

```
DBMS_AQADM.REMOVE_SUBSCRIBER (
  queue_name      IN      VARCHAR2,
  subscriber      IN      sys.aq$_agent);
```

この操作はすぐに有効になり、この操作を含むトランザクションはコミットされます。既存メッセージ内のこのサブスクライバに対するすべての参照は、操作の一部として削除されます。GLOBAL_TOPIC_ENABLED = TRUEの場合は、サブスクライバが削除されると、対応するLDAPエントリも削除されます。

そのコンシューマがデキューできるが保留しているメッセージがあるときに、REMOVE_SUBSCRIBERプロシージャを実行してもエラーにはなりません。このメッセージは、REMOVE_SUBSCRIBERプロシージャが完了すると自動的にデキュー不可になります。

注意:



REMOVE_SUBSCRIBER はキューに対する管理操作です。Oracle Database AQ では、アプリケーションによる管理コールおよび操作コールの同時発行が可能です。REMOVE_SUBSCRIBER は、メッセージをエンキューまたはデキューしている保留中のコールが完了するまでブロックします。保留中のトランザクションの完了は待機しません。

例8-47 サブスクライバの削除

```
DECLARE
  subscriber      sys.aq$_agent;
BEGIN
  subscriber := sys.aq$_agent ('subscriber2', 'test2.msg_queue2@london', null);
  DBMS_AQADM.REMOVE_SUBSCRIBER (
    queue_name => 'test.multiconsumer_81_queue',
    subscriber => subscriber);
END;
```

伝播の管理

あるキューに定義された伝播スケジュールは、そのキューの有効期間中いつでも変更または削除できます。

スケジュールを削除するかわりに一時的に使用不可にすることもできます。すべての管理コールは、スケジュールがアクティブかどうかに関係なく実行されます。スケジュールがアクティブの場合、コールが処理されるまでに数秒かかります。

次のトピックでは、伝播を管理する方法について説明します。

- [キューの伝播のスケジューリング](#)
- [伝播キュー・タイプの検証](#)
- [伝播スケジュールの変更](#)
- [伝播スケジュールの有効化](#)
- [伝播スケジュールの無効化](#)
- [キューの伝播スケジュールの解除](#)

キューの伝播のスケジューリング

DBMS_AQADM.SCHEDULE_PROPAGATIONは、メッセージの伝播をスケジュールします。

```
DBMS_AQADM.SCHEDULE_PROPAGATION (
  queue_name          IN VARCHAR2,
  destination         IN VARCHAR2 DEFAULT NULL,
  start_time          IN DATE       DEFAULT SYSDATE,
  duration             IN NUMBER     DEFAULT NULL,
  next_time            IN VARCHAR2  DEFAULT NULL,
  latency              IN NUMBER     DEFAULT 60,
  destination_queue   IN VARCHAR2  DEFAULT NULL);
```

宛先は、destinationパラメータのデータベース・リンクまたはdestination_queueパラメータのキュー名、あるいはその両方によって識別されます。データベース・リンクのみを指定すると、キューからdblinkへの伝播が行われます。別のデータベースにある複数のキューにメッセージを伝播する場合、すべての伝播が同じ頻度で行われます。

キュー表所有者のスキーマ内のプライベート・データベース・リンクが、パブリック・データベース・リンクと同じ名前の場合、AQは常にプライベート・データベース・リンクを使用します。

宛先キュー名を指定すると、キューからキューへの伝播が行われます。別のデータベース内の複数のキューにメッセージを伝播すると、キューからキューへの伝播により、各スケジュールを独立して構成できます。個別の伝播を有効化または無効化できます。

注意:



あるキューから別のデータベース内のキューへの伝播が必要な場合は、パラメータ destination および destination_queue を指定する必要があります。

キューからキューへの伝播モードは、宛先Oracle Real Application Clusters(Oracle RAC)システムに伝播する場合、透過的なフェイルオーバーをサポートします。キューからキューへの伝播では、Oracle RACでキューの所有者インスタンスが失敗した場合、データベース・リンクを再指定する必要がありません。

宛先にNULLを指定すると、メッセージを同じデータベース内の他のキューにも伝播できます。同じ宛先に複数の受信者を持つ場合、(キューが同じかどうかにかかわらず)メッセージは、すべての受信者に同時に伝播されます。

ソース・キューは、マルチ・コンシューマ向けのキュー表内に存在する必要があります。シングル・コンシューマ・キューを指定すると、エラーORA-24039が発生します。キューまたはデータベース・リンクを参照する場合、Oracle Database Advanced Queuingはシノニムの使用をサポートしていません。

伝播next_timeおよびdurationを指定すると、伝播は指定された期間、定期的に行われます。next_timeまたはdurationを指定せずに待機時間をゼロに指定すると、結果の伝播は永久に実行され、メッセージがキューに表れると伝播し、それ以外のときはアイドルになります。next_timeまたはdurationが指定されずにゼロ以外の待機時間が指定されると(デフォルト)、伝播スケジュールはイベントに基づきます。伝播する必要のあるメッセージがキューにあるときに実行されるよう、スケジュールされます。システム定義の期間、メッセージがないと、伝播する新規メッセージが表れるまでジョブは実行を停止します。ジョブが実行される時間は、準備の完了しているジョブの数やジョブ・キュー・プロセスの数などの他の要因に応じて変化します。

伝播は線形バックオフ・スキームを使用して、障害が発生したスケジュールからの伝播を再試行します。あるスケジュールで続けて障害が発生したときは、最初の再試行は30秒後、次の再試行は60秒後、3回目の再試行は120秒後、というように続きます。再試行時間が現行の伝播枠の期限切れ時刻を超える場合は、次の再試行は、次の伝播枠の開始時刻に行われます。最大16回の再試行が行われた後、そのスケジュールは自動的に使用不可能になります。

注意:



再試行が次の伝播ウィンドウに移動されると、常に移動されるようになり、指数バックオフ・スキームは再試行のスケジュールを管理しません。DBMS_AQADM.SCHEDULE_PROPAGATIONのnext_timeパラメータで指定された日付関数の結果、ウィンドウ間の間隔が短くなると、再試行の失敗数はすぐに16に達し、スケジュールが無効になります。

存在しないdestinationの値を指定すると、このプロシージャはエラーをスローせずに実行します。USER_QUEUE_SCHEDULESビューのLAST_ERROR_MSG列で、ランタイム伝播エラーを問い合わせることができます。

関連項目:

- ジョブ・キューおよびJnnnバックグラウンド・プロセスの詳細は、『[Oracle Database管理者ガイド](#)』のジョブ・キューの管理に関する項を参照してください。
- [Oracle Database Advanced Queuingへのインターネット・アクセス](#)
- [USER_QUEUE_SCHEDULES: ユーザー・スキーマの伝播スケジュール](#)

例8-48 同じデータベース内のキューへの伝播のスケジューリング

```
BEGIN
  DBMS_AQADM.SCHEDULE_PROPAGATION(
    queue_name => 'test.multiconsumer_queue');
END;
/
```

例8-49 他のデータベース内のキューへの伝播のスケジューリング

```
BEGIN
  DBMS_AQADM.SCHEDULE_PROPAGATION(
    queue_name => 'test.multiconsumer_queue',
    destination => 'another_db.world');
END;
/
```

例8-50 キューからキューへの伝播のスケジューリング

```
BEGIN
  DBMS_AQADM.SCHEDULE_PROPAGATION(
    queue_name => 'test.multiconsumer_queue',
    destination => 'another_db.world',
    destination_queue => 'target_queue');
END;
/
```

伝播キュー・タイプの検証

DBMS_AQADM.VERIFY_QUEUE_TYPESは、ソース・キューおよび宛先キューのタイプが同じであるかどうかを検証します。検証の結果は、ディクショナリ表SYS.AQ\$_MESSAGE_TYPES表に格納され、以前にこのコマンドから出力されたすべての結果は上書きされます。

```
DBMS_AQADM.VERIFY_QUEUE_TYPES(
```

```
src_queue_name  IN   VARCHAR2,
dest_queue_name IN   VARCHAR2,
destination     IN   VARCHAR2 DEFAULT NULL,
rc              OUT  BINARY_INTEGER);
```

ソースおよび宛先に同じ型のキューがない場合に指定された変換では、ソース・キュー・タイプを宛先キュー・タイプにマップする必要があります。

注意:

- SYS.AQ\$_MESSAGE_TYPES は、同じソース・キュー、宛先キューおよびデータベース・リンクに対して複数のエントリを持ちますが、変換は異なります。
- VERIFY_QUEUE_TYPES チェックは、AQ 伝播スケジュールごとに一度行われ、送信されるすべての伝播メッセージに対して行われるわけではありません。
- キューのペイロードが変更された場合は、ソース・キューおよび宛先キュー間の既存の伝播スケジュールを削除して再作成する必要があります。

[例8-51](#)には、同じタイプの2つのキューが含まれます。戻り値は次のとおりです。

```
VQT: new style queue
Compatible: 1
```

test.another_queueのかわりにtest.raw_queue(RAW型のキュー)を使用して同じ例を実行すると、この例は次の行を戻します。

```
VQT: new style queue
Compatible: 0
```

例8-51 キュー・タイプの検証

```
SET SERVEROUTPUT ON
DECLARE
rc      BINARY_INTEGER;
BEGIN
  DBMS_AQADM.VERIFY_QUEUE_TYPES(
    src_queue_name => 'test.multiconsumer_queue',
    dest_queue_name => 'test.another_queue',
    rc              => rc);
  DBMS_OUTPUT.PUT_LINE('Compatible: ' || rc);
END;
/
```

伝播スケジュールの変更

DBMS_AQADM.ALTER_PROPAGATION_SCHEDULEは、伝播スケジュールのパラメータを変更します。キューからキューへの伝播のdestination_queueパラメータは変更できません。

```
DBMS_AQADM.ALTER_PROPAGATION_SCHEDULE (
  queue_name      IN   VARCHAR2,
  destination     IN   VARCHAR2 DEFAULT NULL,
  duration        IN   NUMBER   DEFAULT NULL,
```

```
next_time      IN VARCHAR2 DEFAULT NULL,  
latency        IN NUMBER   DEFAULT 60,  
destination_queue IN VARCHAR2 DEFAULT NULL);
```

例8-52 同じデータベース内のキューへの伝播スケジュールの変更

```
BEGIN  
  DBMS_AQADM.ALTER_PROPAGATION_SCHEDULE(  
    queue_name => 'test.multiconsumer_queue',  
    duration   => '2000',  
    next_time  => 'SYSDATE + 3600/86400',  
    latency    => '32');  
END;  
/
```

例8-53 他のデータベース内のキューへの伝播スケジュールの変更

```
BEGIN  
  DBMS_AQADM.ALTER_PROPAGATION_SCHEDULE(  
    queue_name => 'test.multiconsumer_queue',  
    destination => 'another_db.world',  
    duration   => '2000',  
    next_time  => 'SYSDATE + 3600/86400',  
    latency    => '32');  
END;  
/
```

伝播スケジュールの有効化

DBMS_AQADM.ENABLE_PROPAGATION_SCHEDULEは、以前に無効化した伝播スケジュールを有効化します。

```
DBMS_AQADM.ENABLE_PROPAGATION_SCHEDULE(  
  queue_name      IN VARCHAR2,  
  destination     IN VARCHAR2 DEFAULT NULL,  
  destination_queue IN VARCHAR2 DEFAULT NULL);
```

例8-54 同じデータベース内のキューへの伝播の有効化

```
BEGIN  
  DBMS_AQADM.ENABLE_PROPAGATION_SCHEDULE(  
    queue_name => 'test.multiconsumer_queue');  
END;  
/
```

例8-55 他のデータベース内のキューへの伝播の有効化

```
BEGIN  
  DBMS_AQADM.ENABLE_PROPAGATION_SCHEDULE(  
    queue_name => 'test.multiconsumer_queue',  
    destination => 'another_db.world');  
END;  
/
```

伝播スケジュールの無効化

DBMS_AQADM.DISABLE_PROPAGATION_SCHEDULEは、以前に有効化した伝播スケジュールを無効化します。

```
DBMS_AQADM.DISABLE_PROPAGATION_SCHEDULE(  
  queue_name      IN VARCHAR2,
```

```
destination      IN   VARCHAR2 DEFAULT NULL,  
destination_queue IN   VARCHAR2 DEFAULT NULL);
```

例8-56 同じデータベース内のキューへの伝播の無効化

```
BEGIN  
  DBMS_AQADM.DISABLE_PROPAGATION_SCHEDULE(  
    queue_name => 'test.multiconsumer_queue');  
END;  
/
```

例8-57 他のデータベース内のキューへの伝播の無効化

```
BEGIN  
  DBMS_AQADM.DISABLE_PROPAGATION_SCHEDULE(  
    queue_name      => 'test.multiconsumer_queue',  
    destination     => 'another_db.world');  
END;  
/
```

キューの伝播スケジュールの解除

DBMS_AQADM.UNSCHEDULE_PROPAGATIONは、以前にスケジュールした、あるキューから宛先へのメッセージ伝播のスケジュールを解除します。宛先は、destinationパラメータの特定のデータベース・リンク、またはdestination_queueパラメータの名前によって識別されます。

```
DBMS_AQADM.UNSCHEDULE_PROPAGATION(  
  queue_name      IN   VARCHAR2,  
  destination     IN   VARCHAR2 DEFAULT NULL,  
  destination_queue IN VARCHAR2 DEFAULT NULL);
```

例8-58 同じデータベース内のキューへの伝播スケジュールの解除

```
BEGIN  
  DBMS_AQADM.UNSCHEDULE_PROPAGATION(  
    queue_name => 'test.multiconsumer_queue');  
END;  
/
```

例8-59 他のデータベース内のキューへの伝播スケジュールの解除

```
BEGIN  
  DBMS_AQADM.UNSCHEDULE_PROPAGATION(  
    queue_name      => 'test.multiconsumer_queue',  
    destination     => 'another_db.world');  
END;  
/
```

Oracle Database Advanced Queuingエージェントの管理

次のトピックでは、Oracle Database Advanced Queuingエージェントを管理する方法を説明します。

- [Oracle Database Advanced Queuingエージェントの作成](#)
- [Oracle Database Advanced Queuingエージェントの変更](#)
- [Oracle Database Advanced Queuingエージェントの削除](#)
- [データベース・アクセスの許可](#)

- [データベース・アクセスの禁止](#)

Oracle Database Advanced Queuingエージェントの作成

DBMS_AQADM.CREATE_AQ_AGENTは、HTTPプロトコルを使用して、Oracle Database Advanced Queuingインターネット・アクセスのためのエージェントを登録します。

```
DBMS_AQADM.CREATE_AQ_AGENT (  
  agent_name          IN VARCHAR2,  
  certificate_location IN VARCHAR2 DEFAULT NULL,  
  enable_http        IN BOOLEAN DEFAULT FALSE,  
  enable_anyp        IN BOOLEAN DEFAULT FALSE);
```

SYS.AQ\$INTERNET_USERSビューには、すべてのOracle Database Advanced Queuingインターネット・エージェントのリストが表示されます。エージェントが作成、変更または削除された場合、次の条件が満たされていれば、そのエージェントに対してLDAPエントリが作成されます。

- GLOBAL_TOPIC_ENABLED = TRUE
- certificate_locationを指定

Oracle Database Advanced Queuingエージェントの変更

DBMS_AQADM.ALTER_AQ_AGENTは、Oracle Databaseアドバンスド・キューイング・インターネット・アクセス用に登録されたエージェントを変更します。

```
DBMS_AQADM.ALTER_AQ_AGENT (  
  agent_name          IN VARCHAR2,  
  certificate_location IN VARCHAR2 DEFAULT NULL,  
  enable_http        IN BOOLEAN DEFAULT FALSE,  
  enable_anyp        IN BOOLEAN DEFAULT FALSE);
```

Oracle Database Advanced Queuingエージェントが作成、変更または削除された場合、次の条件が満たされていれば、そのエージェントに対してLDAPエントリが作成されます。

- GLOBAL_TOPIC_ENABLED = TRUE
- certificate_locationを指定

Oracle Database Advanced Queuingエージェントの削除

DBMS_AQADM.DROP_AQ_AGENTは、以前にOracle Database Advanced Queuingインターネット・アクセスに登録されたエージェントを削除します。

```
DBMS_AQADM.DROP_AQ_AGENT (  
  agent_name IN VARCHAR2);
```

Oracle Database Advanced Queuingエージェントが作成、変更または削除された場合、次の条件が満たされていれば、そのエージェントに対してLDAPエントリが作成されます。

- GLOBAL_TOPIC_ENABLED = TRUE
- certificate_locationを指定

データベース・アクセスの許可

DBMS_AQADM.ENABLE_DB_ACCESSは、Oracle Databaseアドバンスド・キューイング・インターネット・エージェントに特定のデー

データベース・ユーザーの権限を付与します。このエージェントは、CREATE_AQ_AGENTプロシージャを使用して事前に作成されていません。

```
DBMS_AQADM. ENABLE_DB_ACCESS (  
  agent_name          IN VARCHAR2,  
  db_username         IN VARCHAR2)
```

SYS.AQ\$INTERNET_USERSビューには、すべてのOracle Database Advanced Queuingインターネット・エージェント、および権限が付与されているデータベース・ユーザーの名前のリストが表示されます。

データベース・アクセスの禁止

DBMS_AQADM.DISABLE_DB_ACCESSは、Oracle Database Advanced Queuingインターネット・エージェントから特定のデータベース・ユーザーの権限を取り消します。これらの権限は、ENABLE_DB_ACCESSプロシージャを使用して事前にエージェントに付与されています。

```
DBMS_AQADM. DISABLE_DB_ACCESS (  
  agent_name          IN VARCHAR2,  
  db_username         IN VARCHAR2)
```

LDAPサーバーへの別名の追加

DBMS_AQADM.ADD_ALIAS_TO_LDAPは、LDAPサーバーに別名を追加します。

```
DBMS_AQADM. ADD_ALIAS_TO_LDAP (  
  alias              IN VARCHAR2,  
  obj_location       IN VARCHAR2);
```

このコールは、LDAPのOracle Database Advanced Queuingオブジェクトの別名および識別名を使用して、Oracle Database Advanced Queuingオブジェクトを指す別名を作成します。この別名は、データベース・サーバーの識別名の直下に配置されます。別名が指すオブジェクトは、キュー、エージェントまたは[ConnectionFactory](#)です。

LDAPサーバーからの別名の削除

DBMS_AQADM.DEL_ALIAS_FROM_LDAPは、LDAPサーバーから別名を削除します。

```
DBMS_AQADM. DEL_ALIAS_FROM_LDAP (  
  alias IN VARCHAR2);
```

このコールは、別名の名前を引数として使用し、LDAPサーバーの別名エントリを削除します。別名がLDAPディレクトリのデータベース・サーバー直下にあることが前提です。

9 Oracle Database Advanced Queuingおよび Messaging Gatewayのビュー

次のトピックでは、Oracle Database Advanced Queuing (AQ)管理インタフェースのビューおよびOracle Messaging Gateway (MGW)のビューについて説明します。

注意:



次の項で説明されていないすべてのビューは、[Oracle Database リファレンス](#)で説明されています。

Oracle AQのビュー

- [V\\$AQ_MESSAGE_CACHE_STAT](#): シャード・キューのメモリー管理
- [V\\$AQ_SHARDED_SUBSCRIBER_STAT](#): シャード・キュー・サブスクライバの統計
- [V\\$AQ_MESSAGE_CACHE_ADVICE](#): シミュレートされたメトリック
- [V\\$AQ_REMOTE_DEQUEUE_AFFINITY](#): デキュー・アフィニティ・インスタンス・リスト
- [DBA_QUEUE_TABLES](#): データベース内の全キュー表
- [USER_QUEUE_TABLES](#): ユーザー・スキーマのキュー表
- [ALL_QUEUE_TABLES](#): 現行ユーザーがアクセス可能なキュー表
- [DBA_QUEUES](#): データベース内の全キュー
- [USER_QUEUES](#): ユーザー・スキーマのキュー
- [ALL_QUEUES](#): ユーザーがなんらかの権限を持っているキュー
- [DBA_QUEUE_SCHEDULES](#): すべての伝播スケジュール
- [USER_QUEUE_SCHEDULES](#): ユーザー・スキーマの伝播スケジュール
- [QUEUE_PRIVILEGES](#): ユーザーがキュー権限を持っているキュー
- [AQ\\$<Queue_Table_Name>](#): キュー表内のメッセージ
- [AQ\\$<Queue_Table_Name_S>](#): キューのサブスクライバ
- [AQ\\$<Queue_Table_Name_R>](#): キューのサブスクライバおよびそのルール
- [DBA_QUEUE_SUBSCRIBERS](#): データベース内の全キューのサブスクライバ
- [USER_QUEUE_SUBSCRIBERS](#): ユーザー・スキーマのキューのサブスクライバ
- [ALL_QUEUE_SUBSCRIBERS](#): ユーザーがキュー権限を持っているキューのサブスクライバ
- [DBA_TRANSFORMATIONS](#): すべての変換
- [DBA_ATTRIBUTE_TRANSFORMATIONS](#): すべての変換ファンクション
- [USER_TRANSFORMATIONS](#): ユーザー変換
- [USER_ATTRIBUTE_TRANSFORMATIONS](#): ユーザー変換ファンクション

- [DBA_SUBSCR_REGISTRATIONS](#): すべてのサブスクリプション登録
- [USER_SUBSCR_REGISTRATIONS](#): ユーザー・サブスクリプション登録
- [AQ_INTERNET_USERS](#): インターネット・アクセスが登録されたOracle Database Advanced Queuingエージェント
- [V\\$AQ](#): データベース内の状態ごとのメッセージ数
- [V\\$BUFFERED_QUEUES](#): インスタンスにあるすべてのバッファ済キュー。
- [V\\$BUFFERED_SUBSCRIBERS](#): インスタンスにあるすべてのバッファ済キューのサブスクライバ
- [V\\$BUFFERED_PUBLISHERS](#): インスタンスにあるすべてのバッファ済パブリッシャ
- [V\\$PERSISTENT_QUEUES](#): インスタンスにあるすべてのアクティブな永続キュー
- [V\\$PERSISTENT_SUBSCRIBERS](#): インスタンスにある永続キューのすべてのアクティブなサブスクライバ
- [V\\$PERSISTENT_PUBLISHERS](#): インスタンスにある永続キューのすべてのアクティブなパブリッシャ
- [V\\$PROPAGATION_SENDER](#): 送信(ソース)側のバッファ・キュー伝播スケジュール
- [V\\$PROPAGATION_RECEIVER](#): 受信(宛先)側のバッファ・キュー伝播スケジュール
- [V\\$SUBSCR_REGISTRATION_STATS](#): 通知の診断性
- [V\\$METRICGROUP](#): メトリック・グループに関する情報
- [V\\$AQ_BACKGROUND_COORDINATOR](#): AQのマスター・バックグラウンド・コーディネータ・プロセス(AQPC)のパフォーマンス統計
- [V\\$AQ_JOB_COORDINATOR](#): 1コーディネータ当たりのパフォーマンス統計
- [V\\$AQ_SERVER_POOL](#): 全サーバーのパフォーマンス統計
- [V\\$AQ_CROSS_INSTANCE_JOBS](#): クロス・プロセス・ジョブの説明
- [V\\$AQ_IPC_ACTIVE_MSGS](#)
- [V\\$AQ_IPC_MSG_STATS](#)
- [V\\$AQ_IPC_PENDING_MSGS](#)
- [V\\$AQ_NONDUR_REGISTRATIONS](#): 非永続登録
- [V\\$AQ_NOTIFICATION_CLIENTS](#): セキュアなOCIクライアント接続
- [V\\$AQ_SUBSCRIBER_LOAD](#): 永続サブスクライバ
- [V\\$AQ_NONDUR_SUBSCRIBER](#): 非永続サブスクライバ
- [V\\$AQ_NONDUR_SUBSCRIBER_LWM](#): 非永続サブスクライバのLWM
- [V\\$AQ_MESSAGE_CACHE](#): パフォーマンス統計

Oracle Messaging Gatewayのビュー

- [MGW_GATEWAY](#): 構成およびステータスに関する情報
- [MGW_AGENT_OPTIONS](#): サプリメンタル・オプションおよびプロパティ
- [MGW_LINKS](#): メッセージ・システム・リンクの名前および型
- [MGW_MQSERIES_LINKS](#): WebSphere MQメッセージ・システム・リンク

- [MGW_TIBRV_LINKS: TIB/Rendezvousメッセージ・システム・リンク](#)
- [MGW_FOREIGN_QUEUES: 外部キュー](#)
- [MGW_JOBS: メッセージ・ゲートウェイ伝播ジョブ](#)
- [MGW_SUBSCRIBERS: サブスクライバに関する情報](#)
- [MGW_SCHEDULES: スケジュールに関する情報](#)

DBA_QUEUE_TABLES: データベース内の全キュー表

このビューには、キュー表の所有者インスタンスに関する情報が含まれます。

キュー表は、複数のキューを含むことがあります。この場合、キュー表内の各キューはキュー表と同じ所有者インスタンスを持ちます。[DBA_QUEUE_TABLES](#)の列は、[ALL_QUEUE_TABLES](#)の列と同じです。

関連項目:

DBA_QUEUE_TABLESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_QUEUE_TABLES: ユーザー・スキーマのキュー表

このビューは、ユーザー・スキーマ内のキュー表のみを表示する点を除いて、DBA_QUEUE_TABLESと同じです。

[USER_QUEUE_TABLES](#)には、OWNERの列は含まれていません。

関連項目:

USER_QUEUE_TABLESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

ALL_QUEUE_TABLES: 現行ユーザーがアクセス可能なキュー表

このビューは、現行のユーザーがアクセスできるキュー表を示します。

関連項目:

ALL_QUEUE_TABLESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

DBA_QUEUES: データベース内の全キュー

DBA_QUEUESビューは、データベースの各キューの操作特性を指定します。

列は、[ALL_QUEUES](#)の列と同じです。Oracle Database 12cリリース1 (12.1)には、データ型がVARCHAR2 (5) である新規列SHARDEDがあります。この列の値は、シャード・キューの場合TRUEであり、それ以外の場合はFALSEです。

関連項目:

DBA_QUEUESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_QUEUES: ユーザー・スキーマのキュー

USER_QUEUESビューは、ユーザー・スキーマ内のキューのみを表示する点を除いて、DBA_QUEUESと同じです。

Oracle Database 12cリリース1 (12.1)には、データ型がVARCHAR2 (5)である新規列SHARDEDがあります。この列の値は、シャード・キューの場合TRUEであり、それ以外の場合はFALSEです。

関連項目:

USER_QUEUESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

ALL_QUEUES: ユーザーがなんらかの権限を持っているキュー

ALL_QUEUESビューは、現行ユーザーにエンキューまたはデキュー権限が付与されているすべてのキューを示します。

MANAGE ANY QUEUE、ENQUEUE ANY QUEUEまたはDEQUEUE ANY QUEUEなどのアドバンスド・キューイング・システムの権限がユーザーに付与されている場合、このビューはデータベース内のすべてのキューを示します。Oracle Database 12cリリース1 (12.1)には、データ型がVARCHAR2 (5)である新規列SHARDEDがあります。この列の値は、シャード・キューの場合TRUEであり、それ以外の場合はFALSEです。

関連項目:

ALL_QUEUESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

DBA_QUEUE_SCHEDULES: すべての伝播スケジュール

DBA_QUEUE_SCHEDULESビューは、データベース内の現在のすべてのメッセージ伝播スケジュールを示します。

関連項目:

DBA_QUEUE_SCHEDULESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_QUEUE_SCHEDULES: ユーザー・スキーマの伝播スケジュール

USER_QUEUE_SCHEDULESビューは、ユーザー・スキーマ内のキュー・スケジュールのみを表示する点を除いて、DBA_QUEUE_SCHEDULESと同じです。

関連項目:

USER_QUEUE_SCHEDULESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

QUEUE_PRIVILEGES: ユーザーがキュー権限を持っているキュー

QUEUE_PRIVILEGESビューでは、ユーザーが権限付与者、権限受領者または所有者であるキューが表示されます。

また、キューで有効なロールがPUBLICに付与されているキューも表示されます。

関連項目:

QUEUE_PRIVILEGESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

AQ\$<Queue_Table_Name>: キュー表内のメッセージ

AQ\$<Queue_Table_Name>ビューは、メッセージ・データが格納されているキュー表を示します。

このビューはキュー表ごとに自動的に作成され、キュー・データの問合せに使用されます。デキュー履歴データ(時間、ユーザーIDおよびトランザクションID)は、シングル・コンシューマ・キューについてのみ有効です。

compatibleパラメータを8.1以上に設定して作成されたキュー表では、コンシューマによってデキューされなかったメッセージが「UNDELIVERABLE」と表示されます。これらのメッセージは、msgidでデキューできます。Oracle Database Advanced Queuingのキュー・プロセス・モニターが実行されている場合、メッセージは最終的に例外キューに移動されます。これらのメッセージは、通常のデキュー方法で例外キューからデキューできます。

compatibleパラメータなしで作成されたマルチ・コンシューマ・キュー表、またはcompatibleパラメータを8.0に設定して作成されたマルチ・コンシューマ・キュー表は、コンシューマごとのメッセージ状態を表示せず、全体的なメッセージの状態のみを表示します。

注意:



compatible が 8.0 に設定されているキュー表で作成されたキュー(このマニュアルでは 8.0 形式のキューと呼びます)は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になっています。したがって、新しいキューの作成には 8.1 以降の形式を使用し、既存の 8.0 形式のキューをなるべく早く移行することをお勧めします。

REMOVEモードでメッセージがデキューされると、メッセージをデキューしたコンシューマのDEQ_TIME、DEQ_USER_IDおよびDEQ_TXN_IDが更新されます。

MSGIDおよびORIGINAL_MSGIDを使用して、伝播されたメッセージを連鎖させることができます。メッセージ識別子がm1のメッセージがリモート・キューに伝播されると、m1はリモート・キューのORIGINAL_MSGID列に格納されます。

Oracle Database 10g以降は、AQ\$<Queue_Table_Name>にバッファ済メッセージが含まれます。MSG_STATEに、バッファ済メッセージが次のいずれかの値で示されます。

- IN MEMORY

ユーザーによってエンキューされたバッファ済メッセージ

- DEFERRED
取得プロセスによってエンキューされたバッファ済メッセージ
- SPILLED
ユーザーによってエンキューされ、ディスクに流出したバッファ済メッセージ
- DEFERRED SPILLED
取得によってエンキューされ、ディスクに流出したバッファ済メッセージ
- BUFFERED EXPIRED
期限切れのバッファ済メッセージ

JMSシャード・キューの場合、列RETRY_COUNT、EXCEPTION_QUEUE_OWNER、EXCEPTION_QUEUE、PROPAGATED_MSGID、SENDER_NAME、SENDER_ADDRESS、SENDER_PROTOCOL、ORIGINAL_MSGID、ORIGINAL_QUEUE_NAME、ORIGINAL_QUEUE_OWNER、EXPIRATION_REASONは常にNULLです。

JMSシャード・キューの場合、非永続サブスクライバはセッション固有であるため、このビューには永続サブスクライバのメッセージのみが表示されます。このビューには、インメモリ・シャード・キュー・メッセージ・キャッシュからのデータ(存在する場合)が表示され、存在しない場合、ディスクの値が表示されます。シャード・キューのAQ\$<queue_name>ビューから問い合わせるには、ユーザーは、次のいずれかである必要があります。

- ユーザーが所有者です
- ユーザーにキューに対するデキュー権限があります
- ユーザーにすべてのキューのデキュー権限があります

12cおよび今後のリリースのシャード・キューの場合、ビューは次の点で異なります。

- MSG_PRIORITYは、NUMBER (38)として定義されます
- キュー表のMSG_STATEにはBUFFERED_EXPIREDがないため、UNDELIVERABLEの最大長が、MSG_STATEの長さとしてされます。
- EXPIRATIONは、キュー表ではTIMESTAMP (6) WITH TIME ZONEとして定義されます。
- USER_DATA列は、USERDATA_RAWおよびUSERDATA_BLOBのデコードをUTL_RAW.CAST_TO_VARCHAR2とともに使用して定義されます。
- CONSUMER_NAMEは、VARCHAR2 (128)として定義されます

表9-1 AQ\$<Queue_Table_Name>ビュー

列	データ型	NULL	JMSシャード・キュー 12cリリース1 (12.1)	説明
QUEUE	VARCHAR2 (30)	-		キュー名
SHARD_ID	NUMBER	-		11g では N/A
SUBSHARD_ID	NUMBER	-		11g では N/A
MSG_ID	RAW (16)	NOT NULL		メッセージの一意識別子

列	データ型	NULL	JMSシャード・キュー 12cリリース1 (12.1)	説明
CORR_ID	VARCHAR2 (128)	-		ユーザー指定の相関識別子
MSG_PRIORITY	NUMBER	-	NUMBER (38)	メッセージの優先度
MSG_STATE	VARCHAR2 (16)	-		メッセージの状態。12c リリース 1 (12.1)のキュー表には、BUFFERED_EXPIRED はありません。このため、12c リリース 1 (12.1)では、UNDELIVERABLE の最大長が、MSG_STATE の長さとなります。
DELAY	DATE	-		待機状態のメッセージが準備完了になる日付書式の時間。ENQUEUE_TIME およびユーザーが DELAY を指定した場合と同じです。
DELAY_TIMESTAMP	TIMESTAMP	-		待機状態のメッセージが準備完了になるタイムスタンプ書式の時間。ENQUEUE_TIMESTAMP およびユーザーが DELAY を指定した場合と同じです。
EXPIRATION	NUMBER	-	TIMESTAMP (6) WITH TIME ZONE	メッセージが READY 状態になってから、期限切れになるまでの秒数
RETENTION_TIMESTAMP	TIMESTAMP (6)	-		11g では N/A
ENQ_TIME	DATE	-		エンキュー時刻
ENQ_TIMESTAMP	TIMESTAMP	-		エンキュー時刻
ENQ_USER_ID	NUMBER	-		エンキュー・ユーザーID
ENQ_USER_ID(10.1 の キュー表)	VARCHAR2 (30)	-		エンキュー・ユーザー名
ENQ_TXN_ID	VARCHAR2 (30)	-		エンキュー・トランザクション ID

列	データ型	NULL	JMSシャード・キュー 12cリリース1 (12.1)	説明
DEQ_TIME	DATE	-		デキュー時刻
DEQ_TIMESTAMP	TIMESTAMP	-		デキュー時刻
DEQ_USER_ID	NUMBER	-		デキュー・ユーザーID
DEQ_USER_ID(10.1の キュー表)	VARCHAR2 (30)	-		デキュー・ユーザー名
DEQ_TXN_ID	VARCHAR2 (30)	-		デキュー・トランザクション ID
RETRY_COUNT	NUMBER	-	NULL	再試行回数
EXCEPTION_QUEUE_OWNER	VARCHAR2 (30)	-	NULL	例外キューのスキーマ
EXCEPTION_QUEUE	VARCHAR2 (30)	-	NULL	例外キュー名
USER_DATA	-	-		ユーザー・データ。12c リリース 1 (12.1)では、USER_DATA 列は、USERDATA_RAW および USERDATA_BLOB のデコードを UTL_RAW.CAST_TO_VARCHAR2 とともに使用して定義されます。
SENDER_NAME	VARCHAR2 (30)	-	NULL	メッセージをエンキューしたエージェント名(8.1 互換のキュー表でのみ有効)
SENDER_ADDRESS	VARCHAR2 (1024)	-	NULL	最後に伝播を行ったソース・キュー名およびデータベース名(8.1 互換のキュー表でのみ有効)。ソース・キューがローカル・データベースにあるときは、データベース名は指定されません。
SENDER_PROTOCOL	NUMBER	-	NULL	送信者アドレスのプロトコル。今後の使用のために確保されています(8.1 互換のキュー表でのみ有効)。

列	データ型	NULL	JMSシャード・キュー 12cリリース1 (12.1)	説明
ORIGINAL_MSGID	RAW (16)	-	NULL	ソース・キューにおけるメッセージ ID(8.1 互換のキュー表でのみ有効)
CONSUMER_NAME	VARCHAR2 (30)	-	VARCHAR2 (128)	メッセージを受け取るエージェント名(8.1 互換のマルチ・コンシューマ・キュー表でのみ有効)
ADDRESS	VARCHAR2 (1024)	-		キュー名およびメッセージを受信するエージェントのデータベース・リンク名。アドレスがローカル・データベースに含まれている場合は、データベース・リンク名は指定されません。受信エージェントがキューのローカルである場合は、アドレスは NULL です(8.1 互換のマルチ・コンシューマ・キュー表でのみ有効)
PROTOCOL	NUMBER	-		メッセージを受け取るエージェント・アドレスのプロトコル(8.1 互換のキュー表でのみ有効)
PROPAGATED_MSGID	RAW (16)	-	NULL	メッセージを受け取るエージェントのキューにおけるメッセージ ID(8.1 互換のキュー表でのみ有効)
ORIGINAL_QUEUE_NAME	VARCHAR2 (30)	-	NULL	メッセージの送信元であるキューの名前
ORIGINAL_QUEUE_OWNER	VARCHAR2 (30)	-	NULL	メッセージの送信元であるキューの所有者
EXPIRATION_REASON	VARCHAR2 (19)	-	NULL	メッセージが例外キューに格納された理由。 TIME_EXPIRATION(指定の時間が経過したために期限切れになった)、MAX_RETRY_EXCEEDED(再試行回数が最大値を超えた)および PROPAGATION_FAILURE(伝播

列	データ型	NULL	JMSシャード・キュー 12cリリース1 (12.1) 説明
			の途中で配信不能な状態になった)のいずれかの値で示されます。

注意:



RETRY_COUNT が MAX_RETRIES より大きいメッセージは、例外キューに移動されます。サーバー・プロセスがインスタンスで停止した(ALTER SYSTEM KILL SESSION または SHUTDOWN ABORT など)ためにデキュー・トランザクションが失敗した場合、RETRY_COUNT は増分されません。

AQ\$<Queue_Table_Name_S>: キューのサブスクライバ

AQ\$<Queue_Table_Name_S>ビューは、指定されたキュー表にあるすべてのキューのサブスクライバに関する情報を示します。

ユーザーによってDBMS_AQADM.ADD_SUBSCRIBERを使用して作成されたサブスクライバと、ユーザー作成によるイベントを適用するために適用処理用に作成されたサブスクライバが示されます。また、作成時に変換が含まれていた場合は、[サブスクライバの変換](#)も表示されます。これは、キュー表の作成時に生成されます。

このビューの機能は、DBMS_AQADM.QUEUE_SUBSCRIBERS() プロシージャに相当します。これらのキューのサブスクライバを参照する場合、プロシージャではなくビューを使用することをお勧めします。このビューは、8.1互換のキュー表でのみ作成されます。

表9-2 AQ\$<Queue_Table_Name_S>ビュー

列	データ型	NULL	説明
QUEUE	VARCHAR2 (30)	NOT NULL	サブスクライバが定義されたキューの名前
NAME	VARCHAR2 (30)	-	エージェントの名前
ADDRESS	VARCHAR2 (1024)	-	エージェントのアドレス
PROTOCOL	NUMBER	-	エージェントのプロトコル
TRANSFORMATION	VARCHAR2 (61)	-	変換の名前(NULL の場合もあり)

AQ\$<Queue_Table_Name_R>: キューのサブスクライバおよびそのルール

AQ\$<Queue_Table_Name_R>ビューは、指定したキュー表にあるキューすべてのルールのベース・サブスクライバと、各サブスクライバが定義したルールのテキストを示します。

また、指定された場合は、サブスクライバの変換も表示されます。これは、キュー表の作成時に生成されます。

このビューは、8.1互換のキュー表でのみ作成されます。

表9-3 AQ\$<Queue_Table_Name_R>ビュー

列	データ型	NULL	説明
QUEUE	VARCHAR2 (30)	NOT NULL	サブスクライバが定義されたキューの名前
NAME	VARCHAR2 (30)	-	エージェントの名前
ADDRESS	VARCHAR2 (1024)	-	エージェントのアドレス
PROTOCOL	NUMBER	-	エージェントのプロトコル
RULE	CLOB	-	定義されたルールのテキスト
RULE_SET	VARCHAR2 (65)	-	一連の ルール
TRANSFORMATION	VARCHAR2 (61)	-	変換の名前(NULL の場合もあり)

AQ\$Queue_Name_R: マルチ・コンシューマ・キューのキュー・サブスクライバおよびそのルール

この表は、マルチ・コンシューマ・キューのキュー・サブスクライバおよびそのルールを示しています。

表9-4 AQ\$Queue_Name_R: マルチ・コンシューマ・キューのキュー・サブスクライバおよびそのルール

列	データ型	NULL
QUEUE	VARCHAR2 (30)	NOT NULL
NAME	VARCHAR2 (30)	-
ADDRESS	VARCHAR2 (1024)	-
PROTOCOL	NUMBER	-
RULE	CLOB	-
RULE_SET	VARCHAR2 (65)	-
TRANSFORMATION	VARCHAR2 (65)	-

AQ\$Queue_Name_S: マルチ・コンシューマ・キューのキュー・サブスクライバおよびそのルール

この表は、マルチ・コンシューマ・キューのキュー・サブスクライバおよびそのルールを示しています。

表9-5 AQ\$Queue_Name_S: マルチ・コンシューマ・キューのキュー・サブスクライバおよびそのルール

列	データ型	NULL
QUEUE	VARCHAR2 (30)	NOT NULL
NAME	VARCHAR2 (30)	-
ADDRESS	VARCHAR2 (1024)	-
PROTOCOL	NUMBER	-
TRANSFORMATION	VARCHAR2 (65)	-
QUEUE_TO_QUEUE	VARCHAR2 (5)	-

DBA_QUEUE_SUBSCRIBERS: データベース内の全キューのサブスクライバ

DBA_QUEUE_SUBSCRIBERSビューは、データベースのすべてのキューの全サブスクライバのリストを戻します。

列は、ALL_QUEUE_SUBSCRIBERSの列と同じです。

関連項目:

DBA_QUEUE_SUBSCRIBERSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_QUEUE_SUBSCRIBERS: ユーザー・スキーマのキューのサブスクライバ

USER_QUEUE_SUBSCRIBERSビューは、現行のユーザーのスキーマにあるキューのサブスクライバのリストを戻します。

OWNER列を含まない場合、列はALL_QUEUE_SUBSCRIBERSの列と同じです。

関連項目:

USER_QUEUE_SUBSCRIBERSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

ALL_QUEUE_SUBSCRIBERS: ユーザーがキュー権限を持っているキューのサブスクライバ

ALL_QUEUE_SUBSCRIBERSビューは、現行のユーザーがデキューする権限を持っているキューへのサブスクライバのリストを戻します。

関連項目:

ALL_QUEUE_SUBSCRIBERSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

DBA_TRANSFORMATIONS: すべての変換

DBA_TRANSFORMATIONSビューは、データベース内のすべての変換を示します。

これらの変換をエンキュー、デキュー、サブスクライブなどのアドバンスト・キューイング操作で指定すれば、メッセージ交換での変換が自動的に統合されます。このビューは、DBA権限を持つユーザーのみがアクセスできます。

関連項目:

DBA_TRANSFORMATIONSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

DBA_ATTRIBUTE_TRANSFORMATIONS: すべての変換ファンクション

DBA_ATTRIBUTE_TRANSFORMATIONSビューは、データベース内のすべての変換の変換ファンクションを示します。

関連項目:

DBA_ATTRIBUTE_TRANSFORMATIONSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_TRANSFORMATIONS: ユーザー変換

USER_TRANSFORMATIONSビューは、ユーザーが所有するすべての変換を示します。

変換の定義を参照するには、USER_ATTRIBUTE_TRANSFORMATIONSを問い合わせます。

関連項目:

USER_TRANSFORMATIONSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_ATTRIBUTE_TRANSFORMATIONS: ユーザー変換ファンクション

USER_ATTRIBUTE_TRANSFORMATIONSビューは、ユーザーのすべての変換の変換ファンクションを示します。

関連項目:

USER_ATTRIBUTE_TRANSFORMATIONSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

DBA_SUBSCR_REGISTRATIONS: すべてのサブスクリプション登録

DBA_SUBSCR_REGISTRATIONSビューは、データベース内のすべてのサブスクリプション登録をリストします。

関連項目:

DBA_SUBSCR_REGISTRATIONSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

USER_SUBSCR_REGISTRATIONS: ユーザー・サブスクリプション登録

USER_SUBSCR_REGISTRATIONSビューは、現行ユーザーのデータベース内のすべてのサブスクリプション登録をリストします。

列は、DBA_SUBSCR_REGISTRATIONSの列と同じです。

関連項目:

USER_SUBSCR_REGISTRATIONSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

AQ_INTERNET_USERS: インターネット・アクセスが登録されたOracle Database Advanced Queuingエージェント

AQ\$INTERNET_USERSビューは、Oracle Database Advanced Queuingへのインターネット・アクセスに登録しているエージェントに関する情報を示します。各インターネット・エージェントがマップされているデータベース・ユーザーのリストも示されます。

表9-6 AQ\$INTERNET_USERSビュー

列	データ型	NULL	説明
AGENT_NAME	VARCHAR2 (30)	-	Oracle Database Advanced Queuing インターネット・エージェントの名前
DB_USERNAME	VARCHAR2 (30)	-	このインターネット・エージェントのマップ先のデータベース・ユーザーの名前

列	データ型	NULL	説明
HTTP_ENABLED	VARCHAR2 (4)	-	このエージェントが Oracle Database Advanced Queuing への HTTP アクセスを許可されている場合は YES、許可されていない場合は NO
FTP_ENABLED	VARCHAR2 (4)	-	このエージェントが Oracle Database Advanced Queuing への FTP アクセスを許可されているかどうか(このリリースでは常に NO)

V\$AQ: データベース内の状態ごとのメッセージ数

V\$AQビューは、データベース全体における状態ごとのメッセージ数に関する情報を示します。

Oracle Real Application Clusters環境では、各インスタンスはそれぞれのOracle Database Advanced Queuing 統計情報を固有のシステム・グローバル領域(SGA)に所有し、他のインスタンスによって収集された統計については認識しません。ただし、あるインスタンスでGV\$AQビューに対して問い合わせると、その他のすべてのインスタンスからそれぞれのOracle Database Advanced Queuing統計情報が問合せ元のインスタンスに集計されます。

関連項目:

V\$AQの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$BUFFERED_QUEUES: インスタンスにあるすべてのバッファ済キュー

V\$BUFFERED_QUEUESビューは、インスタンスにあるすべてのバッファ済キューに関する情報を示します。1つのキューに対して1行ずつ示されます。

関連項目:

V\$BUFFERED_QUEUESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$BUFFERED_SUBSCRIBERS: インスタンスにあるすべてのバッファ済キューのサブスクライバ

V\$BUFFERED_SUBSCRIBERSビューは、インスタンスにあるすべてのバッファ済キューのサブスクライバに関する情報を示します。サブスクライバ、キュー各1つに対して1行ずつ示されます。

関連項目:

V\$BUFFERED_SUBSCRIBERSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$BUFFERED_PUBLISHERS: インスタンスにあるすべてのバッファ済パブリッシャ

V\$BUFFERED_PUBLISHERSビューは、インスタンスにあるすべてのバッファ済パブリッシャに関する情報を示します。

送信者、キュー各1つに対して1行ずつ示されます。データベース(またはOracle RAC環境のインスタンス)を再起動すると、値はゼロにリセットされます。

関連項目:

V\$BUFFERED_PUBLISHERSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$PERSISTENT_QUEUES: インスタンスにあるすべてのアクティブな永続キュー

V\$PERSISTENT_QUEUESビューは、インスタンスにあるすべてのアクティブな永続キューに関する情報を示します。

1つのキューに対して1行ずつ示されます。データベース(またはOracle RAC環境のインスタンス)を再起動すると、行は削除されます。

関連項目:

V\$PERSISTENT_QUEUESの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$PERSISTENT_QMN_CACHE: 永続キューのバックグラウンド・タスクのパフォーマンス統計

V\$PERSISTENT_QMN_CACHEビューは、データベースのすべてのキュー表に関連するすべてのバックグラウンド・アクティビティに関する詳細な統計を示します。

1つのキュー表に対して1行ずつ示されます。データベース(またはOracle RAC環境のインスタンス)を再起動すると、値はリセットされます。

関連項目:

V\$PERSISTENT_QMN_CACHEの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$PERSISTENT_SUBSCRIBERS: インスタンスにある永続キューのすべてのアクティブなサブスクライバ

V\$PERSISTENT_SUBSCRIBERSビューは、データベースにある永続キューのすべてのアクティブなサブスクライバに関する情報を示します。

インスタンス、キュー、サブスクライバ各1つに対して1行ずつ示されます。データベース(またはOracle RAC環境のインスタンス)を再起動すると、行は削除されます。

関連項目:

[V\\$PERSISTENT_SUBSCRIBERS](#)の詳細は、『[Oracle Databaseリファレンス](#)』を参照してください。

V\$PERSISTENT_PUBLISHERS: インスタンスにある永続キューのすべてのアクティブなパブリッシャ

V\$PERSISTENT_PUBLISHERSビューは、データベースにある永続キューのすべてのアクティブなパブリッシャに関する情報を示します。

インスタンス、キュー、パブリッシャ各1つに対して1行ずつ示されます。データベース(またはOracle RAC環境のインスタンス)を再起動すると、行は削除されます。

関連項目:

V\$PERSISTENT_PUBLISHERSの詳細は、『[Oracle Databaseリファレンス](#)』を参照してください。

V\$PROPAGATION_SENDER: 送信(ソース)側のバッファ・キュー伝播スケジュール

V\$PROPAGATION_SENDERビューは、送信(ソース)側のバッファ・キュー伝播スケジュールに関する情報を示します。

データベース(またはOracle Real Application Clusters(Oracle RAC)インスタンス)を再起動する、伝播が別のインスタンスに移行する、またはスケジュールが解除された伝播を試行すると、値はゼロにリセットされます。

関連項目:

V\$PROPAGATION_SENDERの詳細は、『[Oracle Databaseリファレンス](#)』を参照してください。

V\$PROPAGATION_RECEIVER: 受信(宛先)側のバッファ・キュー伝播スケジュール

V\$PROPAGATION_RECEIVERビューは、受信(宛先)側のバッファ・キュー伝播スケジュールに関する情報を示します。

データベース(またはOracle Real Application Clusters(Oracle RAC)インスタンス)を再起動する、伝播が別のインスタンスに移行する、またはスケジュールが解除された伝播を試行すると、値はゼロにリセットされます。

関連項目:

V\$PROPAGATION_RECEIVERの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$SUBSCR_REGISTRATION_STATS: 通知の診断性

V\$SUBSCR_REGISTRATION_STATSビューは、通知の診断性に関する情報を示します。

関連項目:

V\$SUBSCR_REGISTRATION_STATSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$METRICGROUP: メトリック・グループに関する情報

V\$METRICGROUPビューは、ストリームの4つの主要コンポーネント、取得、伝播、適用およびキューそれぞれのメトリック・グループに関する情報を示します。

関連項目:

V\$METRICGROUPの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$AQ_MESSAGE_CACHE_STAT: シャード・キューのメモリー管理

V\$AQ_MESSAGE_CACHE_STATビューは、システム・グローバル領域(SGA)にあるstreams_pool内のシャード・キューのメモリー管理に関する統計を示します。シャード・キューは、サブシャード単位でstreams_poolを使用します。したがって、このビューの列は、キューに関係なくサブシャード・レベルでの統計を示しています。このビューは、すべてのシャード・キューにまたがる統計を示します。

表9-7 V\$AQ_MESSAGE_CACHE_STATビュー

列	データ型	説明
INST_ID	NUMBER	シャード・キューのインスタンス ID。
NUM_EVICTED	NUMBER	すべてのシャード・キューで削除されたサブシャードの数
NUM_PREFETCHED	NUMBER	AQ バックグラウンド・プロセスによってプリフェッチされたサブシャードの数
NUM_UNEVICTION	NUMBER	フォアグラウンド・プロセス(デキュー・プロセスなど)によって削除解除されたサブシャードの数
NUM_UNCACHED	NUMBER	アンキャッシュとして格納されているサブシャードの数

列	データ型	説明
NUM_TRACKED	NUMBER	デキュー・レートをアクティブに追跡しているサブシャードの数
NUM_CACHED	NUMBER	メモリーにキャッシュされて格納されているサブシャードの数
MAX_SUBSH_SIZE	NUMBER	サブシャード当たりのメッセージ数から見た、これまでの最大サブシャード・サイズ
MIN_SUBSH_SIZE	NUMBER	サブシャード当たりのメッセージ数から見た、これまでの最小サブシャード・サイズ
MEAN_SUBSH_SIZE	NUMBER	サブシャード当たりのメッセージ数から見た、これまでの平均サブシャード・サイズ
AVG_EVICTION_RATE	NUMBER	1 秒あたりに削除されたサブシャードの平均数
AVG_LOAD_RATE	NUMBER	1 秒あたりにプリフェッチまたは削除解除されたサブシャードの平均数
AVG_EVICTION_TIME	NUMBER	1 つのサブシャードの削除に要する平均時間(ミリ秒)
AVG_LOAD_TIME	NUMBER	1 つのサブシャードの削除解除に要する平均時間(ミリ秒)
AVG_MISS_RATIO	NUMBER	フォアグラウンドの削除解除数とバックグラウンドのプリフェッチ数の平均比率
AVG_THRASH_RATIO	NUMBER	デキューが試行されることなくバックグラウンドでプリフェッチされたサブシャード数とプリフェッチされたサブシャードの総数の平均比率
CON_ID	NUMBER	<p>データが関係するコンテナの ID。使用可能な値は次のとおりです。</p> <ul style="list-style-type: none"> ● 0: この値は、CDB 全体に関連するデータを含む行に使用される。この値は、非 CDB 内の行にも使用される。 ● 1: この値は、ルートのみに関連するデータを含む行に使用される ● n: n は、データを含む行に適用されるコンテナ ID



注意:

前述の一部の列は、シャード・キュー・メモリー・アドバイザによって分析中に使用されます。

V\$AQ_SHARDED_SUBSCRIBER_STAT: シャード・キュー・サブスクライバの統計

V\$AQ_SHARDED_SUBSCRIBER_STATビューは、シャード・キューのサブスクライバに関する統計情報を示します。この統計はメモリー・アドバイザによって使用されます。

表9-8 V\$AQ_SHARDED_SUBSCRIBER_STATビュー

列	データ型	説明
INST_ID	NUMBER	インスタンス識別子
QUEUE_ID	NUMBER	キュー識別子
SUBSCRIBER_ID	NUMBER	サブスクライバ識別子
SHARD_ID	NUMBER	シャード識別子
PRIORITY	NUMBER	シャードの優先度の値
DEQUEUE_SUBSHARD	NUMBER	このシャード内の最後の既知のデキュー・ポジション
ENQUEUED_MSGS	NUMBER	エンキュー済メッセージの数
DEQUEUED_MSGS	NUMBER	デキュー済メッセージの数
ELAPSED_DEQUEUE_TIME	NUMBER	デキューの実行に要した時間(秒)
CPU_DEQUEUE_TIME	NUMBER	デキューの実行に要した実際の CPU 時間(秒)
DEQUEUE_RATE	NUMBER	1 秒当たりにデキューされたメッセージ数
TIME_SINCE_LAST_DEQUEUE	NUMBER	最後のデキュー・アクティビティからの時間(秒)
ESTD_TIME_TO_DRAIN	NUMBER	現在のエンキューとデキューのレートにおけるシャード排出の推定時間(秒)。エンキュー・レートがデキュー・レートよりも大きい場合は null。

列	データ型	説明
ESTD_TIME_TO_DRAIN_NO_ENQ	NUMBER	新規エンキューがない場合のシャード排出の推定時間(秒)
CON_ID	NUMBER	データが関係するコンテナの ID。

V\$AQ_MESSAGE_CACHE_ADVICE: シミュレートされたメトリック

V\$AQ_MESSAGE_CACHE_ADVICEビューは、可能となるメッセージ・キャッシュ・サイズの範囲についてのシミュレートされたメトリックを示します。このビューは、次のようなメトリックの形式で情報を提供することにより、キャッシュ・サイズの決定に役立ちます。

表9-9 V\$AQ_MESSAGE_CACHE_ADVICEビュー

列	データ型	説明
INST_ID	NUMBER	インスタンス ID
SIZE_FOR_ESTIMATE	NUMBER	シミュレーションするキャッシュ・サイズ(MB)
SIZE_FACTOR	NUMBER	現行のキャッシュ・サイズに対するサイズ要因(%)
ESTD_SIZE_TYPE	VARCHAR2	この列は、次の値のいずれかを持ちます。 <ul style="list-style-type: none"> ● MINIMUM - このキャッシュ・サイズは、インメモリーのすべてのデキューが必要(アンキャッシュなし) ● CURRENT - メッセージ・キャッシュの現在のサイズ ● MAXIMUM - このキャッシュ・サイズは、ゼロ削除が必要 ● Null - それ以外の場合
ESTD_CACHED_SUBSHARDS	NUMBER	このサイズでキャッシュされたサブシャードの推定数
ESTD_UNCACHED_SUBSHARDS	NUMBER	このサイズでアンキャッシュされたサブシャードの推定数
ESTD_EVICTIONS	NUMBER	このサイズで削除されたサブシャードの推定数
ESTD_EVICTION_RATE	NUMBER	1 分あたりに削除されるサブシャードの推定数

列	データ型	説明
ESTD_FG_UNEVICTIONS	NUMBER	フォアグラウンド・プロセスによって削除解除されたサブシャードの推定数
ESTD_FG_UNEVICTION_RATE	NUMBER	フォアグラウンド・プロセスによって削除解除されるサブシャードの推定数
ESTD_BG_UNEVICTIONS	NUMBER	バックグラウンド・プロセスによって削除解除されたサブシャードの推定数
ESTD_BG_UNEVICTION_RATE	NUMBER	バックグラウンド・プロセスによって削除解除されるサブシャードの推定数
ESTD_BG_PROCESSES	NUMBER	このサイズに必要なバックグラウンド・プロセスの推定数
TOTAL_ENQUEUE_RATE	NUMBER	1 秒あたりにエンキューされる、シミュレートされたメッセージ数
TOTAL_DEQUEUE_RATE	NUMBER	1 秒あたりにデキューされる、シミュレートされたメッセージ数
AVG_SUBSHARD_SIZE	NUMBER	キャッシュされたサブシャード当たりの、シミュレートされた平均メッセージ数
AVG_SUBSHARD_MEMORY	NUMBER	キャッシュされたサブシャード当たりの、シミュレートされた平均メモリ(MB)
AVG_EVICTION_TIME	NUMBER	キャッシュされたサブシャードの削除にかかるシミュレートされた平均時間(ミリ秒)
AVG_UNEVICTION_TIME	NUMBER	キャッシュされたサブシャードの削除解除にかかるシミュレートされた平均時間(ミリ秒)
FLAGS	NUMBER	将来の内部使用のために予約済
SIMULATION_TIME	NUMBER	シミュレートされた時間(分)
CON_ID	NUMBER	データが関係するコンテナの ID。

V\$AQ_REMOTE_DEQUEUE_AFFINITY: デキュー・アフィニティ・インスタンス・リスト

V\$AQ_REMOTE_DEQUEUE_AFFINITYビューは、シャードの所有者のインスタンスからローカルにはデキューされないサブスクライバのデキュー・アフィニティ・インスタンスを示します。これらのサブスクライバのためにインスタンス間のメッセージ転送が使用されます。

表9-10 V\$AQ_REMOTE_DEQUEUE_AFFINITYビュー

列	データ型	説明
QUEUE_ID	NUMBER	キュー識別子
QUEUE_SCHEMA	VARCHAR2	キューのスキーマ名
QUEUE_NAME	VARCHAR2	キュー名
SUBSCRIBER_ID	NUMBER	サブスクライバ識別子
SHARD_ID	NUMBER	SOURCE_INSTANCE から INST_ID に転送されるサブスクライバのシャード識別子
SOURCE_INSTANCE	NUMBER	そこからシャードが転送される所有者のインスタンス
INST_ID	NUMBER	シャードのサブスクライバのデキュー・インスタンス ID。サブスクライバのシャードが転送される宛先インスタンス

関連項目:

V\$AQ_REMOTE_DEQUEUE_AFFINITYの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$AQ_BACKGROUND_COORDINATOR: AQのマスター・バックグラウンド・コーディネータ・プロセス(AQPC)のパフォーマンス統計

V\$AQ_BACKGROUND_COORDINATORビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。

このビューは、Oracle Database Advanced Queuingのマスター・バックグラウンド・コーディネータ・プロセス(AQPC)のパフォーマンス統計情報を示します。

関連項目:

V\$AQ_BACKGROUND_COORDINATORの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$AQ_JOB_COORDINATOR: 1コーディネータ当たりのパフォーマンス統計

V\$AQ_JOB_COORDINATORビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。

このビューには、AQのマスター・コーディネータによって制御されている各AQコーディネータについて、1コーディネータ当たりのパフォーマンス統計が示されます。

関連項目:



V\$AQ_JOB_COORDINATORの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_SERVER_POOL: 全サーバーのパフォーマンス統計

V\$AQ_SERVER_POOLビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。このビューには、プール内のすべてのサーバーのパフォーマンス統計が示されます。

関連項目:



V\$AQ_SERVER_POOLの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_CROSS_INSTANCE_JOBS: クロス・プロセス・ジョブの説明

V\$AQ_CROSS_INSTANCE_JOBSビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。このビューには、各クロス・プロセス・ジョブが示されます。

各ジョブは、サブスクリバのセットに対して、ソース・インスタンスから宛先インスタンスハシャードのメッセージを転送します。

関連項目:



V\$AQ_CROSS_INSTANCE_JOBSの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_IPC_ACTIVE_MSGS

V\$AQ_IPC_ACTIVE_MSGSは、スレーブによって処理される長く優先順位の高いメッセージおよびマスターによって処理される短いメッセージに関する情報を示します。

関連項目:

V\$AQ_IPC_ACTIVE_MSGSの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_IPC_MSG_STATS

V\$AQ_IPC_MSG_STATSは、累積コール、平均保留/処理時間および最後の失敗など、各メッセージ・クラスの累積統計を示します。

関連項目:

V\$AQ_IPC_MSG_STATSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$AQ_IPC_PENDING_MSGS


V\$AQ_IPC_PENDING_MSGSは、ローカル・マスター・コンテキストに表示される保留メッセージに関する情報が表示されません。

関連項目:

V\$AQ_IPC_PENDING_MSGSの詳細は、[『Oracle Databaseリファレンス』](#)を参照してください。

V\$AQ_NONDUR_REGISTRATIONS: 非永続登録

V\$AQ_NONDUR_REGISTRATIONSビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。このビューは、非永続サブスクリプションに関する情報を表示します。




関連項目:

V\$AQ_NONDUR_REGISTRATIONSの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_NOTIFICATION_CLIENTS: セキュアなOCIクライアント接続

V\$AQ_NOTIFICATION_CLIENTSビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。このビューは、セキュアなOCIクライアント接続のパフォーマンス統計情報を示します。



関連項目:

V\$AQ_NOTIFICATION_CLIENTSの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_SUBSCRIBER_LOAD: 永続サブスクライバ

V\$AQ_SUBSCRIBER_LOADビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。このビューは、Oracle RAC環境における各インスタンスの待機時間によって、シャード・キューのすべてのサブスクライバの負荷を示します。

待機時間は、対応する各インスタンスにおいて、そのサブスクライバのすべてのメッセージが排出されるまでに必要となる、現在時刻からの予想時間(秒)を表します。待機時間の計算では、過去のエンキューとデキューの速度および履歴に基づく将来のエンキューとデキューの速度が考慮されます。

関連項目:



V\$AQ_SUBSCRIBER_LOADの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_NONDUR_SUBSCRIBER: 非永続サブスクライバ

V\$AQ_NONDUR_SUBSCRIBERビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。

V\$AQ_NONDUR_SUBSCRIBERは、シャード・キューの非永続サブスクライバに関する情報を示します。

関連項目:



V\$AQ_NONDUR_SUBSCRIBERの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_NONDUR_SUBSCRIBER_LWM: 非永続サブスクライバのLWM

V\$AQ_NONDUR_SUBSCRIBER_LWMビューは、Oracle Database 12cリリース1 (12.1)以降で適用可能です。非永続サブスクライバのLWMは、シャード、優先順位およびLWM(サブシャード)の組合せです。

関連項目:



V\$AQ_NONDUR_SUBSCRIBER_LWMの詳細は、[『Oracle Database リファレンス』](#)を参照してください。

V\$AQ_MESSAGE_CACHE: パフォーマンス統計

V\$AQ_MESSAGE_CACHEビューは、インスタンスのサブシャード・レベルで、シャード化されたキューのメッセージ・キャッシュに関するパフォーマンス統計情報を示します。

関連項目:



MGW_GATEWAY: 構成およびステータスに関する情報

このビューは、メッセージ・ゲートウェイの構成およびステータスに関する情報をリストします。

表9-11 MGW_GATEWAYビューのプロパティ

名前	型	説明
AGENT_DATABASE	VARCHAR2	メッセージ・ゲートウェイ・エージェントが使用するデータベース接続文字列。NULL は、ローカル接続を使用することを示します。
AGENT_INSTANCE	NUMBER	メッセージ・ゲートウェイ・エージェントが現在実行されているデータベース・インスタンス。エージェントが実行されていない場合、NULL です。
AGENT_JOB	NUMBER	[非推奨] メッセージ・ゲートウェイ・エージェントのプロセスの起動に使用するキューされたジョブのジョブ番号。ジョブ番号は、メッセージ・ゲートウェイの開始時に設定され、停止時に削除されます。
AGENT_NAME	VARCHAR2	メッセージ・ゲートウェイ・エージェントの名前。
AGENT_PING	VARCHAR2	ゲートウェイ・エージェントの ping ステータス。値は次のとおりです。 <ul style="list-style-type: none"> ● NULL: ping は実行されていません。 ● REACHABLE: ping は正常に終了しました。 ● UNREACHABLE: ping は失敗しました。 <p>AGENT_PING は、メッセージ・ゲートウェイ・エージェントへの接続を試行します。ping が失敗した場合、多少の遅延(最大 5 秒)が発生します。AGENT_STATUS が NOT_STARTED または START_SCHEDULED の場合、ping は実行されません。</p>
AGENT_START_TIME	TIMESTAMP	現在実行されているメッセージ・ゲートウェイ・エージェントが起動された時間。エージェントが実行されていない場合、NULL です。
AGENT_STATUS	VARCHAR2	メッセージ・ゲートウェイ・エージェントのステータス。値は次のとおりです。 <ul style="list-style-type: none"> ● NOT_STARTED: メッセージ・ゲートウェイ・エージェントは起動されていません。 ● START_SCHEDULED: メッセージ・ゲートウェイ・エージェントを起動するようにスケジュールされています。DBMS_MGWADM. STARTUP を使用して

名前	型	説明
		<p>メッセージ・ゲートウェイが起動されましたが、メッセージ・ゲートウェイ・エージェントを起動するキューされたジョブが実行されていません。</p> <ul style="list-style-type: none"> ● STARTING: Messaging Gateway エージェントを起動中です。DBMS_MGWADM. STARTUP を使用してメッセージ・ゲートウェイが起動されて、キューされたジョブが実行され、メッセージ・ゲートウェイ・エージェントが起動中です。 ● INITIALIZING: メッセージ・ゲートウェイ・エージェントが起動され、初期化中です。 ● RUNNING: メッセージ・ゲートウェイ・エージェントは実行中です。 ● SHUTTING_DOWN: メッセージ・ゲートウェイ・エージェントは停止中です。 ● BROKEN: 予期しない状態が発生し、メッセージ・ゲートウェイ・エージェントの起動が阻止されました。DBMS_MGWADM. CLEANUP_GATEWAY をコールして、エージェントを起動する必要があります。
AGENT_USER	VARCHAR2	メッセージ・ゲートウェイ・エージェントがデータベースへの接続に使用するデータベース・ユーザー名
COMMENTS	VARCHAR2	エージェントに関するコメント。
CONNTYPE	VARCHAR2	<p>エージェントが使用する接続タイプ。</p> <ul style="list-style-type: none"> ● JDBC_OCI: JDBC OCI ドライバを使用する場合 ● JDBC_THIN: JDBC thin ドライバを使用する場合
INITFILE	VARCHAR2	エージェントが使用するメッセージ・ゲートウェイ初期化ファイルの名前。NULL は、デフォルトの初期化ファイルを使用することを示します。
LAST_ERROR_DATE	DATE	メッセージ・ゲートウェイ・エージェントの最終エラーの日付。メッセージ・ゲートウェイが開始されると最後のエラー情報が削除されます。メッセージ・ゲートウェイ・エージェントが起動に失敗するか、異常な条件のために終了すると、設定されます。
LAST_ERROR_MSG	VARCHAR2	メッセージ・ゲートウェイ・エージェントの最終エラーに関するメッセージ。
LAST_ERROR_TIME	VARCHAR2	メッセージ・ゲートウェイ・エージェントの最終エラーの時間。

名前	型	説明
MAX_CONNECTIONS	NUMBER	[非推奨] Oracle データベースへのメッセージ・コネクションの最大数。
MAX_MEMORY	NUMBER	メッセージ・ゲートウェイ・エージェントにより使用される最大ヒープ・サイズ(MB)。
MAX_THREADS	NUMBER	メッセージ・ゲートウェイ・エージェントにより作成されるメッセージ・スレッドの最大数。
SERVICE	VARCHAR2	エージェントにより使用される Oracle Scheduler ジョブ・クラスに関連付けられたデータベース・サービスの名前。

MGW_AGENT_OPTIONS: サプリメンタル・オプションおよびプロパティ

このビューは、メッセージ・ゲートウェイ・エージェントのサプリメント・オプションおよびプロパティに関する情報をリストします。

表9-12 MGW_AGENT_OPTIONSビュー

列	型	説明
AGENT_NAME	VARCHAR2	メッセージ・ゲートウェイ・エージェントの名前。
ENCRYPTED	VARCHAR2	値を暗号化して格納するかどうかを示します。 <ul style="list-style-type: none"> ● TRUE: 値を暗号化して格納する場合 ● FALSE: 値をクリアテキストとして格納する場合
NAME	VARCHAR2	オプションの名前。
TYPE	VARCHAR2	オプションタイプまたは使用方法。オプションを使用して Java システム・プロパティを設定する場合、JAVA_SYSTEM_PROP です。
VALUE	VARCHAR2	オプションの値。値が暗号化されたフォームで格納される場合、<<ENCRYPTED>>です。

MGW_LINKS: メッセージ・システム・リンクの名前および型

このビューは、現在定義されているメッセージ・システム・リンクの名前および型をリストします。

表9-13 MGW_LINKSビューのプロパティ

名前	型	説明
AGENT_NAME	VARCHAR2	このリンクの伝播ジョブを処理するメッセージ・ゲートウェイ・エージェントの名前

名前	型	説明
LINK_COMMENT	VARCHAR2	リンクに関するユーザー・コメント
LINK_NAME	VARCHAR2	メッセージ・システム・リンクの名前
LINK_TYPE	VARCHAR2	メッセージ・システム・リンクのタイプ。値は次のとおりです。 <ul style="list-style-type: none"> ● MQSERIES: WebSphere MQ リンクの場合 ● TIBRV: TIB/Rendezvous リンクの場合

MGW_MQSERIES_LINKS: WebSphere MQメッセージ・システム・リンク

このビューは、WebSphere MQメッセージ・システム・リンクに関する情報をリストします。このビューには、リンク作成時に指定された大部分のメッセージ・システムのプロパティが表示されます。

表9-14 MGW_MQSERIES_LINKSビューのプロパティ

名前	型	説明
AGENT_NAME	VARCHAR2	このリンクの伝播ジョブを処理するメッセージ・ゲートウェイ・エージェントの名前
CHANNEL	VARCHAR2	接続チャンネル
HOSTNAME	VARCHAR2	WebSphere MQ ホストの名前
INBOUND_LOG_QUEUE	VARCHAR2	インバウンド伝播のログ・キュー
INTERFACE_TYPE	VARCHAR2	メッセージ・インタフェース・タイプ。値は次のとおりです。 <ul style="list-style-type: none"> ● BASE_JAVA: WebSphere MQ Base Java インタフェース ● JMS_CONNECTION: WebSphere MQ JMS に統一されたドメインに依存しないコネクションの場合 ● JMS_QUEUE_CONNECTION: WebSphere MQ JMS キュー・コネクションの場合 ● JMS_TOPIC_CONNECTION: WebSphere MQ JMS トピック・コネクションの場合

名前	型	説明
LINK_COMMENT	VARCHAR2	リンクに関するユーザー・コメント
LINK_NAME	VARCHAR2	メッセージ・システム・リンクの名前
MAX_CONNECTIONS	NUMBER	メッセージ・コネクションの最大数
OPTIONS	SYS. MGW_PROPERT TIES	リンク・オプション
OUTBOUND_LOG_QUEUE	VARCHAR2	発信伝播のログ・キュー
PORT	NUMBER	ポート番号
QUEUE_MANAGER	VARCHAR2	WebSphere キュー・マネージャの名前

MGW_TIBRV_LINKS: TIB/Rendezvousメッセージ・システム・リンク

このビューは、TIB/Rendezvousメッセージ・システム・リンクに関する情報をリストします。このビューには、リンク作成時に指定された大部分のメッセージ・システムのプロパティが表示されます。

表9-15 MGW_TIBRV_LINKSビューのプロパティ

プロパティ名	型	説明
AGENT_NAME	VARCHAR2	このリンクの伝播ジョブを処理するメッセージ・ゲートウェイ・エージェントの名前
CM_LEDGER	VARCHAR2	TIB/Rendezvous CM ledger のファイル名
CM_NAME	VARCHAR2	TIB/Rendezvous CM ledger のファイル名
DAEMON	VARCHAR2	RDV トランスポート用の TIB/Rendezvous デモン・パラメータ
LINK_COMMENT	VARCHAR2	リンクに関するユーザー・コメント
LINK_NAME	VARCHAR2	メッセージ・システム・リンクの名前
NETWORK	VARCHAR2	RDV トランスポート用の TIB/Rendezvous ネットワーク・パラメータ
OPTIONS	SYS. MGW_PROPERT IES	リンク・オプション
SERVICE	VARCHAR2	RDV トランスポート用の TIB/Rendezvous サービス・パラメータ

MGW_FOREIGN_QUEUES: 外部キュー

このビューは、外部キューに関する情報をリストします。このビューには、キュー登録時に指定された大部分のキューのプロパティが表示されます。

表9-16 MGW_FOREIGN_QUEUESビューのプロパティ

名前	型	説明
DOMAIN	VARCHAR2	キュー・ドメイン・タイプ。値は次のとおりです。 <ul style="list-style-type: none">● NULL: メッセージ・システムに基づいてキュー・ドメイン・タイプを自動的に決定● QUEUE: キュー(Point-to-Point)モデルの場合● TOPIC: トピック(パブリッシュ・サブスクライブ)モデルの場合
LINK_NAME	VARCHAR2	メッセージ・システム・リンクの名前
NAME	VARCHAR2	登録キューの名前
OPTIONS	SYS. MGW_PROPERTIES	オプションのキュー・プロパティ
PROVIDER_QUEUE	VARCHAR2	メッセージ・プロバイダ(ネイティブ)のキュー名
QUEUE_COMMENT	VARCHAR2	外部キューに対するユーザー・コメント

MGW_JOBS: メッセージ・ゲートウェイ伝播ジョブ

このビューは、メッセージ・ゲートウェイ伝播ジョブに関する情報をリストします。このビューには、伝播ジョブの作成時に指定された大部分のジョブのプロパティが、その他のステータスおよび統計情報とともに表示されます。

表9-17 MGW_JOBSビュー

列	型	説明
AGENT_NAME	VARCHAR2	このジョブを処理するメッセージ・ゲートウェイ・エージェントの名前
COMMENTS	VARCHAR2	伝播ジョブに関するコメント
DESTINATION	VARCHAR2	メッセージの伝播先の宛先キュー
ENABLED	VARCHAR2	ジョブが有効化されているかどうかを示します。値は次のとおりです。

列	型	説明
		<ul style="list-style-type: none"> ● TRUE: ジョブが有効化されている場合 ● FALSE: ジョブが有効化されていない場合
EXCEPTION_QUEUE	VARCHAR2	伝播のロギングの目的で使用される例外キュー
EXCEPTIONQ_MSGS	NUMBER	オプションタイプまたは使用方法。オプションを使用して Java システム・プロパティを設定する場合、JAVA_SYSTEM_PROP です。
FAILURES	NUMBER	最後にエージェントが起動されてから例外キューに移動されたメッセージの数
JOB_NAME	VARCHAR2	伝播ジョブの名前
LAST_ERROR_MSG	VARCHAR2	伝播の最終エラーに関するメッセージ
LAST_ERROR_DATE	DATE	伝播の最終エラーの日付
LAST_ERROR_TIME	VARCHAR2	伝播の最終エラーの時間
LINK_NAME	VARCHAR2	このジョブで使用されるメッセージ・ゲートウェイ・リンクの名前
OPTIONS	SYS.MGW_PROPERTIES	ジョブ・オプション
POLL_INTERVAL	INTEGER	伝播のポーリング間隔(秒)
PROPAGATED_MSGS	NUMBER	最後にエージェントが起動されてから例外キューに伝播されたメッセージの数
PROP_STYLE	VARCHAR2	メッセージの伝播スタイル。値は次のとおりです。 <ul style="list-style-type: none"> ● NATIVE: ネイティブなメッセージ伝播の場合 ● JMS: JMS メッセージ伝播の場合
PROPAGATION_TYPE	VARCHAR2	伝播型。値は次のとおりです。 <ul style="list-style-type: none"> ● OUTBOUND は、Oracle Database AQ から Oracle 以外への伝播用です。 ● INBOUND は、Oracle 以外から Oracle Database AQ への伝播用です。

列	型	説明
RULE	VARCHAR2	伝播ソースで使用されるサブスクリプション・ルール
SOURCE	VARCHAR2	メッセージの伝播元のソース・キュー
STATUS	VARCHAR2	<p>ジョブ・ステータス。値は次のとおりです。</p> <ul style="list-style-type: none"> ● READY: 伝播のジョブが準備完了であることを示します。実際にメッセージを伝播する前に、ジョブが有効化され、メッセージ・ゲートウェイ・エージェントが実行中である必要があります。 ● RETRY: ジョブへのメッセージの伝播中にエージェントでエラーが発生し、操作を再試行することを示します。 ● FAILED: ジョブが失敗し、エージェントがメッセージの伝播を停止したことを示します。通常、これはリカバリ不可能なエラーが発生または伝播障害制限に達したためです。エージェントでメッセージの伝播を実行する前に、ジョブをリセットする必要があります。ジョブはエージェントを起動すると自動的にリセットされ、DBMS_MGWADM. RESET_JOBを使用すると手動でリセットできます。 ● DELETE_PENDING: ジョブの削除が保留中であることを示します。これは、DBMS_MGWADM. REMOVE_JOB がコールされたが、このジョブに関連する特定のクリーン・アップ・タスクが未完了であることを表します。 ● SUBSCRIBER_DELETE_PENDING: ジョブに関連付けられたサブスクライバの削除が保留中であることを示します。これは、DBMS_MGWADM. REMOVE_SUBSCRIBER がコールされたが、特定のクリーン・アップ・タスクが未完了であることを表します。
TRANSFORMATION	VARCHAR2	メッセージ変換に使用される変換

MGW_SUBSCRIBERS: サブスクライバに関する情報

このビューは、メッセージ・ゲートウェイ・サブスクライバの構成およびステータスに関する情報をリストします。このビューには、サブスクライバの追加時に指定された大部分のサブスクライバのプロパティが、その他のステータスおよび統計情報とともに表示されます。

表9-18 MGW_SUBSCRIBERSビューのプロパティ

名前	型	説明
DESTINATION	VARCHAR2	メッセージの伝播先の宛先キュー
EXCEPTIONQ_MSGS	NUMBER	最後にエージェントが起動されてから伝播例外キューに移動されたメッセージ

名前	型	説明
		の数
EXCEPTION_QUEUE	VARCHAR2	ロギングの目的で使用される例外キュー
FAILURES	NUMBER	伝播の失敗数
LAST_ERROR_DATE	DATE	伝播の最終エラーの日付
LAST_ERROR_MSG	VARCHAR2	伝播の最終エラーに関するメッセージ
LAST_ERROR_TIME	VARCHAR2	伝播の最終エラーの時間
OPTIONS	SYS. MGW_PROPERTIES	サブスクライバ・オプション
PROP_STYLE	VARCHAR2	メッセージの伝播スタイル。値は次のとおりです。 <ul style="list-style-type: none"> ● NATIVE: ネイティブなメッセージ伝播の場合 ● JMS: JMS メッセージ伝播の場合
PROPAGATED_MSGS	NUMBER	最後にエージェントが起動されてから宛先キューに伝播されたメッセージの数
PROPAGATION_TYPE	VARCHAR2	伝播型。値は次のとおりです。 <ul style="list-style-type: none"> ● OUTBOUND は、Oracle Database AQ から Oracle 以外への伝播用です。 ● INBOUND は、Oracle 以外から Oracle Database AQ への伝播用です。
QUEUE_NAME	VARCHAR2	サブスクリプション・ソース・キュー
RULE	VARCHAR2	サブスクリプション・ルール
STATUS	VARCHAR2	サブスクライバ・ステータス。値は次のとおりです。 <ul style="list-style-type: none"> ● ENABLED: サブスクライバが有効化されている場合。 ● DELETE_PENDING: 通常、DBMS_MGWADM.REMOVE_SUBSCRIBER がコールされたが、このサブスクライバに関連する特定のクリーン・アップ・タスクが未完了で、サブスクライバの削除が保留中であることを表しま

名前	型	説明
		す。
SUBSCRIBER_ID	VARCHAR2	伝播サブスクライバの識別子
TRANSFORMATION	VARCHAR2	メッセージ変換に使用される変換

MGW_SCHEDULES: スケジュールに関する情報

このビューは、メッセージ・ゲートウェイ・スケジュールの構成およびステータスに関する情報をリストします。このビューには、スケジュールの作成時に指定された大部分のスケジュールのプロパティが、その他のステータス情報とともに表示されます。

表9-19 MGW_SCHEDULESビューのプロパティ

名前	型	説明
DESTINATION	VARCHAR2	伝播の宛先
LATENCY	NUMBER	伝播枠の待機時間(秒)
NEXT_TIME	VARCHAR2	将来の使用のために予約済
PROPAGATION_TYPE	VARCHAR2	伝播型。値は次のとおりです。 <ul style="list-style-type: none"> ● OUTBOUND は、Oracle Database AQ から Oracle 以外への伝播用です。 ● INBOUND は、Oracle 以外から Oracle Database AQ への伝播用です。
PROPAGATION_WINDOW	NUMBER	将来の使用のために予約済
SCHEDULE_DISABLED	VARCHAR2	スケジュールが使用不可かどうかを示します。Y は、スケジュールが使用不可であることを示します。N は、スケジュールが使用可能であることを示します。
SCHEDULE_ID	VARCHAR2	伝播スケジュールの識別子
SOURCE	VARCHAR2	伝播ソース
START_DATE	DATE	将来の使用のために予約済
START_TIME	VARCHAR2	将来の使用のために予約済

10 PL/SQLを使用したOracle Database Advanced Queuingの操作

次のトピックでは、Oracle Database Advanced Queuing (AQ)のPL/SQL操作インターフェースについて説明します。

- [保護キューの使用方法](#)
- [メッセージのエンキュー](#)
- [メッセージの配列のエンキュー](#)
- [1つ以上のキューのリスニング](#)
- [メッセージのデキュー](#)
- [メッセージの配列のデキュー](#)
- [通知の登録](#)
- [サブスクリバの通知の転送](#)
- [エージェントのLDAPサーバーへの追加](#)
- [エージェントのLDAPサーバーからの削除](#)

関連項目:

- 各プログラム・インターフェースで使用可能な機能のリストは、[「Oracle Database Advanced Queuing: プログラム・インターフェース」](#)を参照してください。
- PL/SQLインターフェースの詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のDBMS_AQに関する項を参照してください。
- Javaインターフェースの詳細は、[Oracle Databaseアドバンスド・キューイングJava APIリファレンス](#)を参照してください
- 『[Oracle Call Interfaceプログラマーズ・ガイド](#)』の「その他のOCIリレーショナル関数」に関する項
- [Oracle Call Interface \(OCI\)](#)の詳細は、『[Oracle Call Interfaceプログラマーズ・ガイド](#)』の「OCIプログラミングの高度なトピック」に関する項を参照してください。

保護キューの使用方法

保護キューの場合は、messages_propertiesパラメータにsender_idを指定する必要があります。

sender_idの詳細は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のMESSAGE_PROPERTIES_T型に関する項を参照してください。

保護キューを使用する場合の要件は、次のとおりです。

- DBMS_AQADM.CREATE_AQ_AGENTを使用して、有効なOracle Database Advanced Queuingエージェントを作成済である必要があります。
- sender_idを、保護キューに対してエンキュー権限があるデータベース・ユーザーにマップします。これには、DBMS_AQADM.ENABLE_DB_ACCESSを使用します。

関連項目:

- [Oracle Database Advanced Queuingエージェントの作成](#)
- [データベース・アクセスの許可](#)

メッセージのエンキュー

このプロシージャは、指定したキューにメッセージを追加します。

```
DBMS_AQ. ENQUEUE (  
  queue_name          IN      VARCHAR2,  
  enqueue_options    IN      enqueue_options_t,  
  message_properties  IN      message_properties_t,  
  payload             IN      " type_name",  
  msgid              OUT     RAW);
```

メッセージのエンキュー後は、[メッセージ](#)・ペイロードを更新できません。メッセージ・ペイロードを変更する場合は、メッセージをデキューして新規メッセージをエンキューする必要があります。

RAW型のペイロードを格納するために、Oracle Database Advanced Queuingでは、ペイロード・リポジトリとしてLOB列を持つ[キュー表](#)が作成されます。ペイロードの最大サイズは、Oracle Database Advanced Queuingへのアクセスに使用するプログラム・インタフェースによって決まります。PL/SQL、Javaおよびプリコンパイラの場合は32K、OCIの場合は4Gです。

メッセージが[受信者](#)を指定しないでマルチ・コンシューマ・キューにエンキューされ、その[キュー](#)にサブスクライバが指定されていない(またはそのメッセージに一致するルールベースのサブスクライバが存在しない)場合、ORA-24033エラーになります。これは、配信可能な受信者またはサブスクライバが存在しないために、そのメッセージが廃棄されることを示す警告です。

複数のメッセージが同時にエンキューされると、すべてのenq_timeが同一になります。この場合、メッセージのデキュー順序は、同じenq_timeを持つ各メッセージに対して定量的に増加する変数step_noによって決まります。同じセッション中にエンキューされる2つのメッセージについて、enq_timeとstep_noの両方が同一になることはありません。

エンキュー・オプション

enqueue_optionsパラメータでは、エンキュー操作に使用可能なオプションを指定します。次の属性が含まれます。

- visibility

visibility属性では、エンキュー・リクエストのトランザクション動作を指定します。ON_COMMIT(デフォルト)を指定すると、エンキューは現行のトランザクションの一部になります。IMMEDIATEを指定すると、エンキュー操作は操作の最後にコミットする自律型トランザクションになります。

[LOB](#)ロケータを使用するときは、IMMEDIATEオプションを使用しないでください。LOBロケータは、トランザクションが継続している間のみ有効です。IMMEDIATEオプションを使用すると、トランザクションが自動的にコミットされるため、ロケータは有効になりません。

バッファ済メッセージを使用するには、visibility属性をIMMEDIATEに設定する必要があります。

- relative_msgid

relative_msgid属性では、順序逸脱操作で参照されるメッセージのメッセージIDを指定します。このパラメータは、sequence_deviationでBEFORE属性を指定しないと無視されます。

- sequence_deviation

sequence_deviation属性では、すでにキューにある他のメッセージを基準にしてメッセージのデキュー時期を指定します。BEFOREを指定すると、メッセージはrelative_msgidで指定したメッセージの前に配置されます。TOPを指定すると、メッセージは他のすべてのメッセージの前に配置されます。

メッセージにsequence_deviationを指定すると、そのメッセージに指定できる遅延および優先順位の値が制限されます。遅延の値は、このメッセージより後にエンキューされるメッセージの遅延以下に設定する必要があります。優先順位は、このメッセージより後にエンキューされるメッセージの優先順位以上に設定する必要があります。

注意:



Oracle Database Advanced Queuing 10g リリース 1 (10.1)より前のリリースでは、message_grouping が TRANSACTIONAL に設定されている場合、sequence_deviation 属性は無効です。

順序逸脱機能は、Oracle Database Advanced Queuing 10g リリース 2 (10.2)では非推奨になりました。

- transformation

transformation属性では、メッセージのエンキュー前に適用する変換を指定します。変換関クションの戻り型は、キューの型と一致する必要があります。

- delivery_mode

delivery_mode属性のデフォルトはPERSISTENTで、メッセージは永続メッセージとしてエンキューされます。BUFFEREDに設定すると、メッセージはバッファ済メッセージとしてエンキューされます。NULL値は設定できません。

メッセージ・プロパティ

message_propertiesパラメータには、Oracle Database Advanced Queuingでメッセージを個別に管理するために使用される情報が含まれます。次の属性が含まれます。

- priority

priority属性では、メッセージの優先順位を指定します。これには、負数も含めたあらゆる数値を使用できます。数値が小さいほど高い優先度を示します。

- delay

delay属性では、メッセージがWAITING状態になっている秒数を指定します。この秒数をすぎると、メッセージはREADY状態になり、デキュー可能になります。NO_DELAYを指定すると、メッセージは即時にデキュー可能になります。msgidを指定してデキューすると、delay指定がオーバーライドされます。

注意:



バッファ済メッセージでは delay はサポートされません。

- expiration

expiration属性では、メッセージがREADY状態になった後でデキュー可能な秒数を指定します。期限切れになる前にデキューされない場合、メッセージはEXPIRED状態で例外キューに移されます。NEVERを指定すると、メッセージは期限

切れになりません。

注意:



メッセージの遅延および期限切れの処理は、キュー・モニター(QMN)のバックグラウンド・プロセスによって行われます。Oracle Database Advanced Queuing の遅延および期限切れの機能を使用するときは、データベースの QMN プロセスを開始する必要があります。

- correlation

correlation属性は、エンキュー時にメッセージのプロデューサが指定する識別子です。

- attempts

attempts属性は、メッセージをデキューするために実行された試行回数を指定します。このパラメータを、エンキュー時刻に設定することはできません。

- recipient_list

recipient_listパラメータは、マルチ・コンシューマに対応するキューにのみ有効です。デフォルトの受信者はキューのサブスクライバです。

- exception_queue

exception_queue属性では、正常に処理できない場合にメッセージが移されるキューの名前を指定します。指定した例外キューが移動時に存在しない場合、そのメッセージは、キュー表に関連付けられているデフォルトの例外キューに移動され、アラート・ログに警告ログが記録されます。

- delivery_mode

エンキュー時にメッセージ・プロパティで指定したdelivery_modeの値は無視されます。エンキュー・オプションで指定した値を使用して、メッセージの配信モードが設定されます。エンキュー・オプションで配信モードを指定しなければ、デフォルトでPERSISTENTに設定されます。

- enqueue_time

enqueue_time属性では、メッセージがエンキューされた時刻を指定します。この値は常に協定世界時(UTC)で指定され、システムによって決定されるため、ユーザーがエンキュー時に設定することはできません。

注意:



システム・クロックの季節変動情報(標準時と夏時間間の切替えなど)は各キュー表とともに格納されるため、季節変動は自動的に enqueue_time に反映されます。他のなんらかの理由でシステム・クロックに変更があった場合は、Oracle Database Advanced Queuing 用のデータベースを再起動して変更後の時刻を取得する必要があります。

- state

state属性では、デキュー時のメッセージの状態を指定します。このパラメータを、エンキュー時刻に設定することはできません。

- sender_id

sender_id属性は、メッセージ・プロデューサがエンキュー時に指定したaq\$_agent型の識別子です。

- original_msgid

original_msgid属性は、Oracle Database AQでメッセージの伝播に使用されます。

- transaction_group

transaction_group属性では、メッセージのトランザクション・グループを指定します。この属性は、DBMS_AQ.DEQUEUE_ARRAYでのみ設定されます。DBMS_AQ.ENQUEUEまたはDBMS_AQ.ENQUEUE_ARRAYを介してメッセージのトランザクション・グループを設定する場合、この属性は使用できません。

- user_property

user_property属性はオプションです。ペイロードの追加情報の格納に使用します。

次のトピックの例には、[「Oracle Database Advanced Queuing管理インタフェース」](#)の例と同じユーザー、メッセージ型、キュー表およびキューを使用しています。まだこれらの構造をテスト環境で作成していない場合は、次の例を実行する必要があります。

- [例8-1](#)
- [例8-2](#)
- [例8-3](#)
- [例8-5](#)
- [例8-7](#)
- [例8-8](#)
- [例8-23](#)
- [例8-25](#)
- [例8-26](#)
- [例8-27](#)
- [例8-28](#)
- [例8-36](#)

[例8-1](#)では、管理権限を持つユーザーとして接続する必要があります。前述のリストの他の例では、ユーザーtest_admとして接続できます。キューの作成後に、[「キューの開始」](#)に示すようにキューを開始する必要があります。特に明記しないかぎり、すべての例は通常のキュー・ユーザー'test'として接続して実行できます。

LOB型メッセージのエンキュー

[例10-3](#)では、[例8-1](#)で作成したtest.lob_typeメッセージ・ペイロード・オブジェクト型を使用して、プロシージャblobenqueue()を作成します。エンキュー時に、LOB属性がEMPTY_BLOBに設定されます。エンキューの完了後、トランザクションがコミットされる前に、test.lob_qtabキュー表のuser_data列からLOB属性が選択されます。LOBデータは、LOBインタフェース(OCIおよびPL/SQLの両方を介して使用可能)を使用してキューに書き込まれます。実際のエンキュー操作は、[を参照してください](#)。

デキュー時には、メッセージ・ペイロードにLOBロケータが含まれます。デキュー後、トランザクションがコミットされる前に、このLOBロケータを使用してLOBデータを読み取ることができます。[例10-14](#)を参照してください。

シングル・コンシューマ・キューへの複数メッセージのエンキュー

[例10-5](#)では、6つのメッセージをtest.obj_queue1にエンキューしています。これらのメッセージは、[例10-17](#)でデキューされます。

マルチ・コンシューマ・キューへの複数メッセージのエンキュー

[例10-6](#)では、ユーザー' test_adm'として接続し、キュー test.multiconsumer_queue にサブスクライバ RED および GREEN を追加する必要があります。この2つのサブスクライバは [例10-7](#) で必要になります。

[例10-7](#)では、送信者001からの複数のメッセージをエンキューします。MESSAGE 1はすべてのキュー・サブスクライバを対象としています。MESSAGE 2はREDおよびBLUEを対象としています。これらのメッセージは、[例10-17](#)でデキューされます。

グループ化されたメッセージのエンキュー

[例10-8](#)では、3つのメッセージ・グループをエンキューし、各グループにはメッセージが3つずつ含まれます。これらのメッセージは、[例10-16](#)でデキューされます。

遅延および期限切れのメッセージのエンキュー

[例10-9](#)のアプリケーションでは、メッセージは現時点の1週間前から3週間後までの期間内にデキューされる必要があります。期限切れは最も早いデキュー時刻から計算されるため、この場合は期限切れを2週間に設定する必要があります。

例10-1 キュー名とペイロードを指定したメッセージのエンキュー

```
DECLARE
  enqueue_options    DBMS_AQ.enqueue_options_t;
  message_properties DBMS_AQ.message_properties_t;
  message_handle     RAW(16);
  message            test.message_typ;
BEGIN
  message := test.message_typ(001, 'TEST MESSAGE', 'First message to obj_queue');
  DBMS_AQ.ENQUEUE (
    queue_name       => 'test.obj_queue',
    enqueue_options  => enqueue_options,
    message_properties => message_properties,
    payload          => message,
    msgid            => message_handle);
  COMMIT;
END;
/
```

例10-2 優先順位を指定したメッセージのエンキュー

```
DECLARE
  enqueue_options    DBMS_AQ.enqueue_options_t;
  message_properties DBMS_AQ.message_properties_t;
  message_handle     RAW(16);
  message            test.order_typ;
BEGIN
  message := test.order_typ(002, 'PRIORITY MESSAGE', 'priority 30');
  message_properties.priority := 30;
  DBMS_AQ.ENQUEUE (
    queue_name       => 'test.priority_queue',
    enqueue_options  => enqueue_options,
    message_properties => message_properties,
    payload          => message,
    msgid            => message_handle);
  COMMIT;
END;
/
```

例10-3 LOB型メッセージのエンキュー・プロシージャの作成

```
CREATE OR REPLACE PROCEDURE blobenqueue(msgno IN NUMBER) AS
  enq_userdata      test.lob_typ;
```



```

enq_msgid          RAW(16);
enqueue_options    DBMS_AQ.enqueue_options_t;
message_properties DBMS_AQ.message_properties_t;
lob_loc            BLOB;
buffer             RAW(4096);
BEGIN
buffer             := HEXTORAW(RPAD(' FF', 4096, ' FF'));
enq_userdata       := test.lob_typ(msgno, 'Large Lob data', EMPTY_BLOB(), msgno);
DBMS_AQ.ENQUEUE (
    queue_name      => 'test.lob_queue',
    enqueue_options => enqueue_options,
    message_properties => message_properties,
    payload         => enq_userdata,
    msgid           => enq_msgid);
SELECT t.user_data.data INTO lob_loc
FROM lob_qtab t
WHERE t.msgid = enq_msgid;
DBMS_LOB.WRITE(lob_loc, 2000, 1, buffer );
COMMIT;
END;
/

```

例10-4 LOB型メッセージのエンキュー

```

BEGIN
FOR i IN 1..5 LOOP
    blobenqueue(i);
END LOOP;
END;
/

```

例10-5 複数メッセージのエンキュー

```

SET SERVEROUTPUT ON
DECLARE
enqueue_options    DBMS_AQ.enqueue_options_t;
message_properties DBMS_AQ.message_properties_t;
message_handle     RAW(16);
message            test.message_typ;
BEGIN
message := test.message_typ(001, 'ORANGE', 'ORANGE enqueued first. ');
DBMS_AQ.ENQUEUE (
    queue_name      => 'test.obj_queue',
    enqueue_options => enqueue_options,
    message_properties => message_properties,
    payload         => message,
    msgid           => message_handle);
message := test.message_typ(001, 'ORANGE', 'ORANGE also enqueued second. ');
DBMS_AQ.ENQUEUE (
    queue_name      => 'test.obj_queue',
    enqueue_options => enqueue_options,
    message_properties => message_properties,
    payload         => message,
    msgid           => message_handle);
message := test.message_typ(001, 'YELLOW', 'YELLOW enqueued third. ');
DBMS_AQ.ENQUEUE (
    queue_name      => 'test.obj_queue',
    enqueue_options => enqueue_options,
    message_properties => message_properties,
    payload         => message,

```

```

        msgid                => message_handle);
message := test.message_typ(001, 'VIOLET', 'VIOLET enqueued fourth. ');
DBMS_AQ.ENQUEUE (
    queue_name                => 'test.obj_queue',
    enqueue_options           => enqueue_options,
    message_properties        => message_properties,
    payload                   => message,
    msgid                    => message_handle);
message := test.message_typ(001, 'PURPLE', 'PURPLE enqueued fifth. ');
DBMS_AQ.ENQUEUE (
    queue_name                => 'test.obj_queue',
    enqueue_options           => enqueue_options,
    message_properties        => message_properties,
    payload                   => message,
    msgid                    => message_handle);
message := test.message_typ(001, 'PINK', 'PINK enqueued sixth. ');
DBMS_AQ.ENQUEUE (
    queue_name                => 'test.obj_queue',
    enqueue_options           => enqueue_options,
    message_properties        => message_properties,
    payload                   => message,
    msgid                    => message_handle);
COMMIT;
END;
/

```

例10-6 サブスクライバREDおよびGREENの追加

```

DECLARE
    subscriber                sys.aq$_agent;
BEGIN
    subscriber                := sys.aq$_agent('RED', NULL, NULL);
    DBMS_AQADM.ADD_SUBSCRIBER(
        queue_name            => 'test.multiconsumer_queue',
        subscriber            => subscriber);

    subscriber                := sys.aq$_agent('GREEN', NULL, NULL);
    DBMS_AQADM.ADD_SUBSCRIBER(
        queue_name            => 'test.multiconsumer_queue',
        subscriber            => subscriber);
END;
/

```

例10-7 マルチ・コンシューマ・キューへの複数メッセージのエンキュー

```

DECLARE
    enqueue_options           DBMS_AQ.enqueue_options_t;
    message_properties        DBMS_AQ.message_properties_t;
    recipients                DBMS_AQ.aq$_recipient_list_t;
    message_handle            RAW(16);
    message                   test.message_typ;
BEGIN
    message := test.message_typ(001, 'MESSAGE 1', 'For queue subscribers');
    DBMS_AQ.ENQUEUE (
        queue_name            => 'test.multiconsumer_queue',
        enqueue_options       => enqueue_options,
        message_properties    => message_properties,
        payload               => message,
        msgid                 => message_handle);

```

```

message := test.message_typ(001, 'MESSAGE 2', 'For two recipients');
recipients(1) := sys.aq$_agent('RED', NULL, NULL);
recipients(2) := sys.aq$_agent('BLUE', NULL, NULL);
message_properties.recipient_list := recipients;
DBMS_AQ.ENQUEUE (
    queue_name          => 'test.multiconsumer_queue',
    enqueue_options     => enqueue_options,
    message_properties  => message_properties,
    payload             => message,
    msgid               => message_handle);
COMMIT;
END;
/

```

例10-8 グループ化されたメッセージのエンキュー

```

DECLARE
    enqueue_options     DBMS_AQ.enqueue_options_t;
    message_properties  DBMS_AQ.message_properties_t;
    message_handle      RAW(16);
    message             test.message_typ;
BEGIN
    FOR groupno in 1..3 LOOP
        FOR msgno in 1..3 LOOP
            message := test.message_typ(
                001,
                'GROUP ' || groupno,
                'Message ' || msgno || ' in group ' || groupno);
            DBMS_AQ.ENQUEUE (
                queue_name          => 'test.group_queue',
                enqueue_options     => enqueue_options,
                message_properties  => message_properties,
                payload             => message,
                msgid               => message_handle);
        END LOOP;
    COMMIT;
    END LOOP;
END;
/

```

例10-9 遅延と期限切れを指定したメッセージのエンキュー

```

DECLARE
    enqueue_options     DBMS_AQ.enqueue_options_t;
    message_properties  DBMS_AQ.message_properties_t;
    message_handle      RAW(16);
    message             test.message_typ;
BEGIN
    message := test.message_typ(001, 'DELAYED', 'Message is delayed one week. ');
    message_properties.delay       := 7*24*60*60;
    message_properties.expiration := 2*7*24*60*60;
    DBMS_AQ.ENQUEUE (
        queue_name          => 'test.obj_queue',
        enqueue_options     => enqueue_options,
        message_properties  => message_properties,
        payload             => message,
        msgid               => message_handle);
    COMMIT;
END;
/

```

例10-10 変換を指定したメッセージのエンキュー

```
DECLARE
  enqueue_options    DBMS_AQ.enqueue_options_t;
  message_properties DBMS_AQ.message_properties_t;
  message_handle     RAW(16);
  message            test.message_typ;
BEGIN
  message := test.message_typ(001, 'NORMAL MESSAGE', 'enqueued to obj_queue');
  enqueue_options.transformation := 'message_order_transform';
  DBMS_AQ.ENQUEUE (
    queue_name          => 'test.priority_queue',
    enqueue_options    => enqueue_options,
    message_properties => message_properties,
    payload            => message,
    msgid              => message_handle);
  COMMIT;
END;
/
```

メッセージの配列のエンキュー

ENQUEUE_ARRAY関数を使用して、対応する配列のメッセージ・プロパティを使用するペイロードの配列をエンキューします。

```
DBMS_AQ.ENQUEUE_ARRAY (
  queue_name          IN  VARCHAR2,
  enqueue_options    IN  enqueue_options_t,
  array_size         IN  PLS_INTEGER,
  message_properties_array IN message_properties_array_t,
  payload_array      IN  VARRAY,
  msgid_array        OUT msgid_array_t)
RETURN PLS_INTEGER;
```

結果には、エンキューされたメッセージのメッセージ識別子の配列が出力されます。関数はエンキューに成功したメッセージの数を返します。

バッファ済メッセージの場合、配列のエンキューはサポートされていませんが、array_sizeを1に設定してDBMS_AQ.ENQUEUE_ARRAY()を使用すると、バッファ済メッセージをエンキューできます。

message_properties_arrayパラメータは、メッセージ・プロパティの配列です。このレコードには、ペイロードの配列の各要素に対応する要素が必要です。配列内のすべてのメッセージの配信モードは同じです。

ペイロード構造は、VARRAYまたはネストした表のいずれかです。メッセージIDは、DBMS_AQ.msgid_array_t型のRAW(16)のエントリの配列に戻されます。

リレーショナルDB環境における配列操作と同様に、すべての状況に対応可能な1つの最適な配列サイズを指定することはできません。アプリケーション開発者は、様々な配列サイズをテストして、各アプリケーションにとって最適な値を決定する必要があります。

関連項目:

- [エンキュー・オプション](#)
- [メッセージ・プロパティ](#)

例10-11 メッセージの配列のエンキュー

```
DECLARE
  enqueue_options      DBMS_AQ.enqueue_options_t;
  msg_prop_array       DBMS_AQ.message_properties_array_t;
  msg_prop              DBMS_AQ.message_properties_t;
  payload_array        test.msg_table;
  msgid_array          DBMS_AQ.msgid_array_t;
  retval               PLS_INTEGER;
BEGIN
  payload_array := msg_table(
    message_typ(001, 'MESSAGE 1', 'array enqueued to obj_queue'),
    message_typ(001, 'MESSAGE 2', 'array enqueued to obj_queue'));
  msg_prop_array := DBMS_AQ.message_properties_array_t(msg_prop, msg_prop);

  retval := DBMS_AQ.ENQUEUE_ARRAY(
    queue_name           => 'test.obj_queue',
    enqueue_options     => enqueue_options,
    array_size          => 2,
    message_properties_array => msg_prop_array,
    payload_array       => payload_array,
    msgid_array         => msgid_array);

  COMMIT;
END;/
```

1つ以上のキューのリスニング

このプロシージャでは、監視するキューを指定します(複数可)。

```
DBMS_AQ.LISTEN(
  agent_list           IN    aq$agent_list_t,
  wait                 IN    BINARY_INTEGER DEFAULT FOREVER,
  listen_delivery_mode IN    PLS_INTEGER DEFAULT PERSISTENT,
  agent                OUT   sys.aq$agent
  message_delivery_mode OUT   PLS_INTEGER);

TYPE aq$agent_list_t IS TABLE OF aq$agent INDEXED BY BINARY_INTEGER;
```

このコールの引数に、エージェント・リストがあります。各エージェントは、名前、アドレスおよびプロトコルの一意の組合せで識別されます。

監視対象のキューは、リストされている各エージェントのアドレス・フィールドで指定されます。エージェントには、監視対象の各キューに対するデキュー権限が必要です。マルチ・コンシューマ・キューを監視する場合はエージェント名を指定する必要がありますが、シングル・コンシューマ・キューの場合、エージェント名を指定する必要はありません。アドレスでサポートされているのは、ローカル・キューのみです。プロトコルは、将来の使用に備えて確保されています。

注意:



マルチ・コンシューマ・キューのリスニングは Java [API](#) ではサポートされていません。

`listen_delivery_mode`パラメータでは、エージェントの対象メッセージの型を指定します。デフォルトのPERSISTENTの場合、エージェントには永続メッセージのみが通知されます。BUFFEREDに設定すると、エージェントにはバッファ済メッセージのみが通知されます。PERSISTENT_OR_BUFFEREDに設定すると、エージェントには両方の型が通知されます。

このコールは、リストにあるエージェントによって処理可能なメッセージがある場合にエージェントとメッセージ型を戻すブロック・コールです。処理可能なエージェントが複数ある場合、最初にリストされたエージェントのみが戻されます。待機期限切れになった時点で1つもメッセージがない場合は、エラーになります。

Listenコールから正常に戻っても、指定されたキューの1つにリストされたいずれかのエージェントのメッセージがあることを意味しているにすぎません。対象となっているエージェントは、関連メッセージを継続してデキューする必要があります。

注意:



非永続キューには LISTEN はコールできません。

test.obj_queueとtest.priority_queueの両方にメッセージ(それぞれ[例10-1](#)および[例10-2](#)でエンキュー)が含まれていますが、[例10-12](#)では次の値のみが戻されます。

```
Message in Queue: "TEST"."OBJ_QUEUE"
```

test_agent_listでエージェントの順序が保持される場合、test.priority_queueはtest.obj_queueの前に表示されるため、この例では次のように戻されます。

```
Message in Queue: "TEST"."PRIORITY_QUEUE"
```

関連項目:

[AQエージェント型](#)

例10-12 シングル・コンシューマ・キューのリスニング(タイムアウト0(ゼロ))

```
SET SERVEROUTPUT ON
DECLARE
  agent          sys.aq$_agent;
  test_agent_list DBMS_AQ.aq$_agent_list_t;
BEGIN
  test_agent_list(1) := sys.aq$_agent(NULL, 'test.obj_queue', NULL);
  test_agent_list(2) := sys.aq$_agent(NULL, 'test.priority_queue', NULL);
  DBMS_AQ.LISTEN(
    agent_list => test_agent_list,
    wait       => 0,
    agent      => agent);
  DBMS_OUTPUT.PUT_LINE('Message in Queue: ' || agent.address);
END;
```

メッセージのデキュー

このプロシージャでは、メッセージを指定のキューからデキューします。

```
DBMS_AQ.DEQUEUE (
  queue_name          IN      VARCHAR2,
  dequeue_options     IN      dequeue_options_t,
  message_properties  OUT     message_properties_t,
  payload             OUT     "type_name",
```

デキューの対象として、永続メッセージのみ、バッファ・メッセージのみ、またはその両方のいずれかを選択できます。次に示すデキュー・オプションのリストのdelivery_modeを参照してください。

関連項目:

[「メッセージ・プロパティ」](#)

デキュー・オプション

dequeue_optionsパラメータでは、デキュー操作に使用可能なオプションを指定します。次の属性が含まれます。

- consumer_name

コンシューマは、DBMS_AQADM.ADD_SUBSCRIBERプロシージャのAQ\$_AGENT型またはメッセージ・プロパティの受信者リストに使用した名前を指定して、キューからメッセージをデキューできます。値を指定すると、consumer_nameと一致するメッセージのみがアクセスされます。キューがマルチ・コンシューマ用に設定されていない場合は、このフィールドをNULL(デフォルト)に設定する必要があります。

- dequeue_mode

dequeue_mode属性では、デキュー関連のロック動作を指定します。BROWSEを指定すると、メッセージはロックを取得せずにデキューされます。LOCKEDを指定すると、メッセージはデキューされ、トランザクションの間中は書き込みロックが継続します。REMOVEを指定すると、メッセージはデキューされて削除されます(デフォルト)。メッセージは保存プロパティに基づいてキュー表に保存されます。REMOVE_NO_DATAを指定すると、メッセージは更新済または削除済としてマークされます。

- navigation

navigation属性では、デキュー対象メッセージの位置を指定します。FIRST_MESSAGEを指定すると、使用可能で検索基準と一致する最初のメッセージがデキューされます。NEXT_MESSAGEを指定すると、検索基準と一致する次に使用可能なメッセージがデキューされます(デフォルト)。前のメッセージがメッセージ・グループに属する場合は、そのメッセージ・グループに属するメッセージの中から、検索基準に一致する次に使用可能なメッセージがデキューされます。

NEXT_TRANSACTIONを指定すると、現行のトランザクション・グループ内のメッセージがスキップされ、次のトランザクション・グループの最初のメッセージがデキューされます。この設定を使用できるのは、キューでメッセージのグループ化が可能な場合のみです。

- visibility

visibility属性では、新規メッセージのデキュー時期を指定します。ON_COMMITを指定すると、デキューは現行のトランザクションの一部になります(デフォルト)。IMMEDIATEを指定すると、デキュー操作は操作の最後にコミットする自律型トランザクションとなります。visibility属性は、BROWSEデキュー・モードでは無視されます。

配信モードDBMS_AQ.BUFFEREDまたはDBMS_AQ.PERSISTENT_OR_BUFFEREDでメッセージをデキューする場合は、visibilityを常にIMMEDIATEに設定する必要があります。

- wait

wait属性では、検索基準と一致する使用可能なメッセージがない場合の待機時間を指定します。数値を指定すると、操作はその秒数だけ待機します。FOREVERを指定すると、操作は永続的に待機します(デフォルト)。NO_WAITを指定すると、操作は待機しません。

- msgid

msgid属性では、デキュー対象メッセージのメッセージ識別子を指定します。msgidを指定しないかぎり、READY状態のメッセージのみがデキューされます。

- correlation

correlation属性では、デキュー対象メッセージの相関識別子を指定します。一連のデキュー・コール間では、FIRST_MESSAGEナビゲーション・オプションを指定しないと相関識別子を変更できません。

相関識別子は、Oracle Database Advanced Queuingでは解釈されない、アプリケーション定義の識別子です。パーセント記号やアンダースコアなど、特殊なパターン・マッチング文字を使用できます。複数のメッセージがパターンと一致する場合、デキュー順序は不定になり、キューのソート順序は無視されます。



注意:

バッファ済メッセージの場合、デキュー・オプション correlation および deq_condition の両方がサポートされますが、索引を作成してこれらの問合せを最適化することはできません。

- deq_condition

deq_condition属性は、SQL問合せのWHERE句に似たブール式です。このブール式には、メッセージ・プロパティの条件、ユーザー・データ・プロパティ(オブジェクト・ペイロードのみ)およびPL/SQLまたはSQL関数を含めることができます。

メッセージ・ペイロード(オブジェクト・ペイロード)のデキュー条件を指定するには、句にオブジェクト・タイプの属性を使用します。ペイロードが格納されたキュー・テーブルの特定の列を指定するには、各属性の前に修飾子として tab.user_dataを追加する必要があります。

deq_condition属性は4000文字以内です。複数のメッセージがデキュー条件を満たす場合、デキュー順序は不定になり、キューのソート順は考慮されません。

- transformation

transformation属性では、メッセージがデキューされてからコール側に戻されるまでに適用される変換を指定します。

- delivery_mode

delivery_mode属性では、デキューするメッセージの型を指定します。DBMS_AQ.PERSISTENTに設定すると、永続メッセージのみがデキューされます。DBMS_AQ.BUFFEREDに設定すると、バッファ済メッセージのみがデキューされます。

デフォルトのDBMS_AQ.PERSISTENT_OR_BUFFEREDに設定すると、永続メッセージとバッファ済メッセージの両方がデキューされます。デキューされたメッセージのメッセージ・プロパティのdelivery_mode属性は、そのメッセージがバッファ済メッセージであったか永続メッセージであったかを示します。

デキューの順序は、デキュー・オプションのメッセージ識別子および相関識別子でオーバーライドされないかぎり、キュー表の作成時に指定されたソート順によって決定されます。

キュー操作には、データベース読み込み一貫性メカニズムが適用されます。たとえば、トランザクションによる参照が始まってからエンキューされたメッセージを、BROWSEコールでは参照できません。

[コミット時間キュー](#)は、[近似CSCN](#)に基づいて確定的な順序を確立できないかぎり、BROWSEまたはDEQUEUEのコールではメッセージを参照できません。

dequeue_conditionsパラメータのnavigation属性がNEXT_MESSAGE (デフォルト)の場合、後続のデキューでは、最初のデキューで取得されたスナップショットに基づいて、キューからメッセージが取り出されます。したがって、最初のデキュー・コマンド以降にエンキューされたメッセージは、キューに残っているメッセージがすべて処理された後にのみ処理されます。すべてのメッセージがエ

ンキュー済の場合、またはキューに優先順位ベースの順序が設定されていない場合、これは問題ではありません。ただし、アプリケーションでキューにある最も優先順位の高いメッセージを処理する必要がある場合は、FIRST_MESSAGEナビゲーション・オプションを使用する必要があります。

注意:



同時にエンキューされるメッセージがある場合には、FIRST_MESSAGE ナビゲーション・オプションを使用すると効率的です。FIRST_MESSAGE オプションを指定しないと、Oracle Database Advanced Queuing では最初のデキュー・コマンドと同様のスナップショットが継続的に生成されるため、パフォーマンスの低下につながります。FIRST_MESSAGE オプションを指定すると、Oracle Database Advanced Queuing では、すべてのデキュー・コマンドに新規のスナップショットが使用されます。

メッセージをグループ化できるキューに同一トランザクションのメッセージがエンキューされると、グループが形成されます。トランザクションに1つのメッセージしかエンキューされていない場合でも、事実上1つのグループが形成されます。1つのトランザクションでグループ化できるメッセージの数に、上限はありません。

メッセージをグループ化できないキューでは、LOCKEDまたはREMOVEモードのデキューによって1つのメッセージのみがロックされます。それに対して、グループの一部であるメッセージをデキューする操作では、グループ全体がロックされます。これは、グループのメッセージ全体を1つの単位として処理する必要がある場合に有効です。

グループ内のすべてのメッセージがデキューされている場合、そのデキューはグループ内のすべてのメッセージが処理済であることを示すエラーを戻します。その後、アプリケーションはNEXT_TRANSACTIONを使用して、次の使用可能なグループからのメッセージのデキューを開始します。使用可能なグループがない場合は、dequeue_optionsのwait属性で指定した期間が終了するとデキューがタイムアウトになります。

一般的には、メッセージのコンシューマはデキュー・インタフェースによってメッセージにアクセスすると考えられます。処理済または処理が終了していないメッセージは、メッセージIDまたはSELECTコマンドを使用して参照できます。

[例10-13](#)では、[例10-1](#)でエンキューされたメッセージが戻されます。戻り値は次のとおりです。

```
From Sender No. 1
Subject: TEST MESSAGE
Text: First message to obj_queue
```

関連項目:

- [デキュー・モード](#)

LOB型メッセージのデキュー

[例10-14](#)では、[Example 10-4](#)でエンキューされたLOBタイプ・メッセージをデキューするために、blobdequeue() プロシージャを作成します。実際のデキューは[例10-15](#)で示されています。戻り値は次のとおりです。

```
Amount of data read: 2000
Amount of data read: 2000
Amount of data read: 2000
Amount of data read: 2000
Amount of data read: 2000
```

グループ化されたメッセージのデキュー

[例10-16](#)を実行すると、[例10-8](#)でエンキューした、グループ化されたメッセージをデキューできます。戻り値は次のとおりです。

```
GROUP 1: Message 1 in group 1
GROUP 1: Message 2 in group 1
GROUP 1: Message 3 in group 1
Finished GROUP 1
GROUP 2: Message 1 in group 2
GROUP 2: Message 2 in group 2
GROUP 2: Message 3 in group 2
Finished GROUP 2
GROUP 3: Message 1 in group 3
GROUP 3: Message 2 in group 3
GROUP 3: Message 3 in group 3
Finished GROUP 3
No more messages
```

マルチ・コンシューマ・キューからのデキュー

[例10-17](#)を実行すると、[例10-7](#)でREDについてエンキューしたメッセージをデキューできます。REDをGREENに変更し、さらにBLUEに変更すると、それぞれのメッセージのデキューにも使用できます。例の出力は、それぞれ異なります。

REDはマルチ・コンシューマ・キューのサブスクライバで、MESSAGE 2の指定の受信者でもあるため、両方のメッセージを取得します。

```
Message: MESSAGE 1 .. For queue subscribers
Message: MESSAGE 2 .. For two recipients
No more messages for RED
```

GREENはサブスクライバにすぎないため、受信者が指定されていないキュー内のメッセージ(この場合はMESSAGE 1)のみを取得します。

```
Message: MESSAGE 1 .. For queue subscribers
No more messages for GREEN
```

BLUEはキューのサブスクライバではありませんが、MESSAGE 2の宛先に指定されています。

```
Message: MESSAGE 2 .. For two recipients
No more messages for BLUE
```

[例10-18](#)では、[例10-5](#)でエンキューしたメッセージをPINKが見つかるまで参照し、PINKを削除します。この例は、次の行を戻します。

```
Browsed Message Text: ORANGE enqueued first.
Browsed Message Text: ORANGE also enqueued second.
Browsed Message Text: YELLOW enqueued third.
Browsed Message Text: VIOLET enqueued fourth.
Browsed Message Text: PURPLE enqueued fifth.
Browsed Message Text: PINK enqueued sixth.
Removed Message Text: PINK enqueued sixth.
```

デキュー・モード

[例10-19](#)では、[例10-5](#)でエンキューしたメッセージをPURPLEが見つかるまでロック・モードでプレビューし、PURPLEを削除します。この例は、次の行を戻します。

```
Locked Message Text: ORANGE enqueued first.
Locked Message Text: ORANGE also enqueued second.
Locked Message Text: YELLOW enqueued third.
```

```
Locked Message Text: VIOLET enqueued fourth.  
Locked Message Text: PURPLE enqueued fifth.  
Removed Message Text: PURPLE enqueued fifth.
```

例10-13 オブジェクト型メッセージのデキュー

```
SET SERVEROUTPUT ON  
DECLARE  
dequeue_options    DBMS_AQ.dequeue_options_t;  
message_properties DBMS_AQ.message_properties_t;  
message_handle     RAW(16);  
message            test.message_typ;  
BEGIN  
    dequeue_options.navigation := DBMS_AQ.FIRST_MESSAGE;  
    DBMS_AQ.DEQUEUE(  
        queue_name      => 'test.obj_queue',  
        dequeue_options => dequeue_options,  
        message_properties => message_properties,  
        payload         => message,  
        msgid           => message_handle);  
    DBMS_OUTPUT.PUT_LINE('From Sender No.' || message.sender_id);  
    DBMS_OUTPUT.PUT_LINE('Subject: ' || message.subject);  
    DBMS_OUTPUT.PUT_LINE('Text: ' || message.text);  
    COMMIT;  
END;  
/
```

例10-14 LOB型メッセージのデキュー・プロシージャの作成

```
CREATE OR REPLACE PROCEDURE blobdequeue(msgno IN NUMBER) AS  
dequeue_options    DBMS_AQ.dequeue_options_t;  
message_properties DBMS_AQ.message_properties_t;  
msgid              RAW(16);  
payload            test.lob_typ;  
lob_loc            BLOB;  
amount             BINARY_INTEGER;  
buffer             RAW(4096);  
BEGIN  
    DBMS_AQ.DEQUEUE(  
        queue_name      => 'test.lob_queue',  
        dequeue_options => dequeue_options,  
        message_properties => message_properties,  
        payload         => payload,  
        msgid           => msgid);  
    lob_loc             := payload.data;  
    amount              := 2000;  
    DBMS_LOB.READ(lob_loc, amount, 1, buffer);  
    DBMS_OUTPUT.PUT_LINE('Amount of data read: ' || amount);  
    COMMIT;  
END;  
/
```

例10-15 LOB型メッセージのデキュー

```
BEGIN  
    FOR i IN 1..5 LOOP  
        blobdequeue(i);  
    END LOOP;  
END;  
/
```

例10-16 グループ化されたメッセージのデキュー

```
SET SERVEROUTPUT ON
DECLARE
  dequeue_options      DBMS_AQ.dequeue_options_t;
  message_properties   DBMS_AQ.message_properties_t;
  message_handle       RAW(16);
  message              test.message_typ;
  no_messages          exception;
  end_of_group         exception;
  PRAGMA EXCEPTION_INIT (no_messages, -25228);
  PRAGMA EXCEPTION_INIT (end_of_group, -25235);
BEGIN
  dequeue_options.wait      := DBMS_AQ.NO_WAIT;
  dequeue_options.navigation := DBMS_AQ.FIRST_MESSAGE;
  LOOP
    BEGIN
      DBMS_AQ.DEQUEUE(
        queue_name      => 'test.group_queue',
        dequeue_options => dequeue_options,
        message_properties => message_properties,
        payload         => message,
        msgid           => message_handle);
      DBMS_OUTPUT.PUT_LINE(message.subject || ': ' || message.text);
      dequeue_options.navigation := DBMS_AQ.NEXT_MESSAGE;
    EXCEPTION
      WHEN end_of_group THEN
        DBMS_OUTPUT.PUT_LINE('Finished ' || message.subject);
        COMMIT;
        dequeue_options.navigation := DBMS_AQ.NEXT_TRANSACTION;
    END;
  END LOOP;
EXCEPTION
  WHEN no_messages THEN
    DBMS_OUTPUT.PUT_LINE('No more messages');
END;
/
```

例10-17 マルチ・コンシューマ・キューからのREDのメッセージのデキュー

```
SET SERVEROUTPUT ON
DECLARE
  dequeue_options      DBMS_AQ.dequeue_options_t;
  message_properties   DBMS_AQ.message_properties_t;
  message_handle       RAW(16);
  message              test.message_typ;
  no_messages          exception;
  PRAGMA EXCEPTION_INIT (no_messages, -25228);
BEGIN
  dequeue_options.wait      := DBMS_AQ.NO_WAIT;
  dequeue_options.consumer_name := 'RED';
  dequeue_options.navigation := DBMS_AQ.FIRST_MESSAGE;
  LOOP
    BEGIN
      DBMS_AQ.DEQUEUE(
        queue_name      => 'test.multiconsumer_queue',
        dequeue_options => dequeue_options,
        message_properties => message_properties,
        payload         => message,
        msgid           => message_handle);
```

```

DBMS_OUTPUT.PUT_LINE(' Message: ' || message.subject || ' .. ' || message.text );
dequeue_options.navigation := DBMS_AQ.NEXT_MESSAGE;
END;
END LOOP;
EXCEPTION
  WHEN no_messages THEN
    DBMS_OUTPUT.PUT_LINE (' No more messages for RED ');
COMMIT;
END;
/

```

例10-18 参照モードによるデキューと指定のメッセージの削除

```

SET SERVEROUTPUT ON
DECLARE
  dequeue_options      DBMS_AQ.dequeue_options_t;
  message_properties   DBMS_AQ.message_properties_t;
  message_handle       RAW(16);
  message              test.message_typ;
BEGIN
  dequeue_options.dequeue_mode := DBMS_AQ.BROWSE;
  LOOP
    DBMS_AQ.DEQUEUE(
      queue_name          => 'test.obj_queue',
      dequeue_options     => dequeue_options,
      message_properties  => message_properties,
      payload             => message,
      msgid               => message_handle);
    DBMS_OUTPUT.PUT_LINE (' Browsed Message Text: ' || message.text);
    EXIT WHEN message.subject = 'PINK';
  END LOOP;
  dequeue_options.dequeue_mode := DBMS_AQ.REMOVE;
  dequeue_options.msgid        := message_handle;
  DBMS_AQ.DEQUEUE(
    queue_name          => 'test.obj_queue',
    dequeue_options     => dequeue_options,
    message_properties  => message_properties,
    payload             => message,
    msgid               => message_handle);
  DBMS_OUTPUT.PUT_LINE(' Removed Message Text: ' || message.text);
  COMMIT;
END;
/

```

例10-19 ロックモードによるデキューと指定のメッセージの削除

```

SET SERVEROUTPUT ON
DECLARE
  dequeue_options      DBMS_AQ.dequeue_options_t;
  message_properties   DBMS_AQ.message_properties_t;
  message_handle       RAW(16);
  message              test.message_typ;
BEGIN
  dequeue_options.dequeue_mode := DBMS_AQ.LOCKED;
  LOOP
    DBMS_AQ.dequeue(
      queue_name          => 'test.obj_queue',
      dequeue_options     => dequeue_options,
      message_properties  => message_properties,
      payload             => message,

```

```

        msgid                => message_handle);
    DBMS_OUTPUT.PUT_LINE(' Locked Message Text: ' || message.text);
    EXIT WHEN message.subject = 'PURPLE';
END LOOP;
dequeue_options.dequeue_mode := DBMS_AQ.REMOVE;
dequeue_options.msgid        := message_handle;
DBMS_AQ.DEQUEUE(
    queue_name                => 'test.obj_queue',
    dequeue_options           => dequeue_options,
    message_properties         => message_properties,
    payload                   => message,
    msgid                     => message_handle);
DBMS_OUTPUT.PUT_LINE(' Removed Message Text: ' || message.text);
COMMIT;
END;
/

```

メッセージの配列のデキュー

DEQUEUE_ARRAY関数を使用して、ペイロードの配列および対応する配列のメッセージ・プロパティをデキューします。

```

DBMS_AQ.DEQUEUE_ARRAY(
    queue_name                IN      VARCHAR2,
    dequeue_options           IN      dequeue_options_t,
    array_size                 IN      PLS_INTEGER,
    message_properties_array  OUT     message_properties_array_t,
    payload_array             OUT     VARRAY,
    msgid_array               OUT     msgid_array_t)
RETURN PLS_INTEGER;

```

結果には、デキューされたメッセージのペイロードの配列、メッセージIDおよびメッセージ・プロパティが出力されます。また、正常にデキューされたメッセージの数も戻されます。

バッファ済メッセージの場合、配列のデキューはサポートされていませんが、array_sizeを1に設定してDBMS_AQ.DEQUEUE_ARRAY()を使用すると、バッファ済メッセージをデキューできます。

ペイロード構造は、VARRAYまたはネストした表のいずれかです。メッセージ識別子は、DBMS_AQ.msgid_array_t型のRAW(16)のエントリの配列に戻されます。メッセージ・プロパティは、DBMS_AQ.message_properties_array_t型の配列に戻されます。

リレーショナルDB環境における配列操作と同様に、すべての状況に対応可能な1つの最適な配列サイズを指定することはできません。アプリケーション開発者は、様々な配列サイズをテストして、各アプリケーションにとって最適な値を決定する必要があります。

DBMS_AQ.DEQUEUEで使用可能なデキュー・オプションはすべて、DBMS_AQ.DEQUEUE_ARRAYでも使用できます。デキューの対象として、永続メッセージのみ、バッファ・メッセージのみ、またはその両方のいずれかを選択できます。また、dequeue_optionsのnavigation属性には、DBMS_AQ.DEQUEUE_ARRAYに固有の2つのオプションが用意されています。

メッセージをデキューするときは、トランザクション・グループのすべてのメッセージを1つのコールでデキューできます。また、複数のトランザクション・グループのメッセージをデキューすることもできます。いずれの方法も、次のナビゲーション・メソッドを使用して指定できます。

- NEXT_MESSAGE_ONE_GROUP
- FIRST_MESSAGE_ONE_GROUP
- NEXT_MESSAGE_MULTI_GROUP
- FIRST_MESSAGE_MULTI_GROUP

ナビゲーション・メソッドNEXT_MESSAGE_ONE_GROUPは、検索基準と一致したメッセージを次の使用可能なトランザクション・グループから配列にデキューします。ナビゲーション・メソッドFIRST_MESSAGE_ONE_GROUPは、位置をキューの先頭にリセットし、1つのトランザクション・グループから使用可能かつ検索基準と一致するすべてのメッセージをデキューします。

デキューされるメッセージの数は、配列サイズの制限によって決まります。トランザクション・グループ内のメッセージ数がarray_sizeを超えている場合は、複数のDEQUEUE_ARRAYをコールして、そのトランザクション・グループのすべてのメッセージをデキューする必要があります。

ナビゲーション・メソッドNEXT_MESSAGE_MULTI_GROUPおよびFIRST_MESSAGE_MULTI_GROUPは、それぞれのカウンターパートONE_GROUPと同様に機能しますが、1つのトランザクション・グループには制限されません。配列にデキューされる各メッセージには、一連のメッセージ・プロパティが対応付けられています。メッセージ・プロパティtransaction_groupによって、同じトランザクション・グループに属するメッセージが特定されます。

[例10-20](#)では、[例10-11](#)でエンキューされたメッセージがデキューされます。戻り値は次のとおりです。

```
Number of messages dequeued: 2
```

関連項目:

[メッセージのデキュー](#)

例10-20 メッセージの配列のデキュー

```
SET SERVEROUTPUT ON
DECLARE
  dequeue_options      DBMS_AQ.dequeue_options_t;
  msg_prop_array       DBMS_AQ.message_properties_array_t :=
                        DBMS_AQ.message_properties_array_t();
  payload_array        test.msg_table;
  msgid_array          DBMS_AQ.msgid_array_t;
  retval               PLS_INTEGER;
BEGIN
  retval := DBMS_AQ.DEQUEUE_ARRAY(
    queue_name          => 'test.obj_queue',
    dequeue_options    => dequeue_options,
    array_size         => 2,
    message_properties_array => msg_prop_array,
    payload_array      => payload_array,
    msgid_array        => msgid_array);
  DBMS_OUTPUT.PUT_LINE('Number of messages dequeued: ' || retval);
END;/
```

通知の登録

このプロシージャでは、メッセージ通知に使用する電子メール・アドレス、ユーザー定義PL/SQLプロシージャまたはHTTP URLを登録します。

```
DBMS_AQ.REGISTER(
  reg_list      IN SYS.AQ$REG_INFO_LIST,
  reg_count     IN NUMBER);
```

注意:



12cリリース2 (12.2)以降では、ユーザーが生成するキュー名の最大長は 122 バイトです。[「キューの作成」](#)を参照してください。

reg_listパラメータは、SYS.AQ\$_REG_INFOオブジェクトのリストです。通知のサービス品質は、SYS.AQ\$_REG_INFOの qosflags属性を使用して指定できます。

reg_countパラメータでは、reg_listのエントリ数を指定します。各サブスクリプションには固有のreg_listエントリが必要です。一度に、複数のサブスクリプションを登録できます。

PL/SQL通知の受信時には、コールバックの起動に使用されるOracle Database Advanced Queuingメッセージ・プロパティ記述子に通知メッセージのdelivery_modeがDBMS_AQ.PERSISTENTまたはDBMS_AQ.BUFFEREDとして指定されます。

電子メールによる通知を登録する場合は、データベースで電子メール通知の送信に使用されるSMTPサーバーのホスト名およびポート名を設定する必要があります。必要な場合は、データベースのsentfromフィールドを使用して、発信元の電子メールアドレスを設定します。この機能を使用するには、Java対応のデータベースが必要です。

HTTP通知を登録した場合は、データベースでHTTP通知の転送に使用されるプロキシ・サーバーのホスト名とポート番号、および非プロキシ・ドメインのリストを設定できます。

ジョブ・キュー・プロセスにより処理される通知は、内部キューSYS.AQ_SRVNTFN_TABLE_Qに格納されます。通知に失敗すると、Oracle Database Advanced Queuingは失敗した通知を最大でMAX_RETRIES回まで再試行します。

注意:



SYS.AQ_SRVNTFN_TABLE_QのMAX_RETRIESおよびRETRY_DELAYプロパティは変更できます。新規の設定はすべての通知に適用されます。

関連項目:

- SYS.AQ\$_REG_INFOオブジェクトの詳細は、[「AQ登録情報型」](#)を参照してください。
- メッセージ・プロパティ記述子の詳細は、[「AQ通知記述子型」](#)を参照してください。

例10-21 通知の登録

```
DECLARE
  reginfo          sys.aq$_reg_info;
  reg_list         sys.aq$_reg_info_list;
BEGIN
  reginfo := sys.aq$_reg_info(
    'test.obj_queue',
    DBMS_AQ.NAMESPACE_ANONYMOUS,
    'http://www.company.com:8080',
    HEXTORAW('FF'));
  reg_list := sys.aq$_reg_info_list(reginfo);
  DBMS_AQ.REGISTER(
    reg_list => reg_list,
    reg_count => 1);
COMMIT;
```



```
END;  
/
```

通知の登録解除

このプロシージャでは、メッセージ通知に使用する電子メール・アドレス、ユーザー定義PL/SQLプロシージャまたはHTTP URLの登録を解除します。

```
DBMS_AQ.UNREGISTER(  
  reg_list      IN SYS.AQ$_REG_INFO_LIST,  
  reg_count     IN NUMBER);
```

サブスクライバの通知の転送

このプロシージャでは、匿名サブスクリプションのリストに通知を転送し、サブスクリプションに登録する全クライアントが永続メッセージの通知を取得できるようにします。

```
DBMS_AQ.POST(  
  post_list     IN SYS.AQ$_POST_INFO_LIST,  
  post_count    IN NUMBER);
```

この機能は、バッファ済メッセージについてはサポートされません。

countパラメータでは、post_listのエントリ数を指定します。転送済の各サブスクリプションは、post_list内に固有のエントリを持つ必要があります。一度に、複数のサブスクリプションを転送できます。

post_listパラメータでは、転送する匿名サブスクリプションのリストを指定します。次の3つの属性があります。

- name
name属性では、転送する匿名サブスクリプションの名前を指定します。
- namespace
namespace属性では、サブスクリプションの名前空間を指定します。DBMS_AQ.POSTを介して他のアプリケーションから通知を受信するには、namespaceをDBMS_AQ.NAMESPACE_ANONYMOUSに設定する必要があります。
- payload
payload属性では、匿名サブスクリプションに転送するペイロードを指定します。このコールに関連付けるペイロードがない可能性もあります。

このコールは、最も効果的な転送を保証します。通知は、登録されたクライアントに対して最大で1回行われます。このコールは、主に軽量な通知に使用されます。アプリケーションがより確実な保証を必要とする場合は、キューにエンキューできます。

例10-22 オブジェクト型メッセージの転送

```
DECLARE  
  postinfo      sys.aq$_post_info;  
  post_list     sys.aq$_post_info_list;  
BEGIN  
  postinfo := sys.aq$_post_info(' test.obj_queue', 0, HEXTORAW(' FF' ));  
  post_list := sys.aq$_post_info_list(postinfo);  
  DBMS_AQ.POST(  
    post_list     => post_list,  
    post_count    => 1);  
  COMMIT;  
END;  
/
```

エージェントのLDAPサーバーへの追加

このプロシージャは、LDAPサーバーにOracle Databaseアドバンスド・キューイング・エージェントのエントリを作成します。

```
DBMS_AQ. BIND_AGENT (  
  agent          IN SYS. AQ$_AGENT,  
  certificate    IN VARCHAR2 default NULL);
```

agentパラメータは、[Lightweight Directory Access Protocol](#) (LDAP)サーバーに登録されるOracle Database Advanced Queuingエージェントを指定します。

certificateパラメータでは、このエージェントにデジタル証明書(usercertificate属性)を使用するLDAP内のOrganizationalPersonエントリの位置(LDAP識別名)を指定します。たとえば、cn=OE、cn=ACME、cn=comは、指定のエージェントに証明書を使用するOrganizationalPerson OEの識別名です。エージェントがデジタル証明を持たない場合、このパラメータはデフォルトでNULLに設定されます。

関連項目:

[「AQエージェント型」](#)

エージェントのLDAPサーバーからの削除

このプロシージャでは、LDAPサーバーからOracle Database Advanced Queuingエージェントのエントリを削除します。

```
DBMS_AQ. UNBIND_AGENT (  
  agent          IN SYS. AQ$_AGENT);
```

11 Oracle Java Message Serviceの概要

次のトピックでは、Oracle Database Advanced Queuing (AQ)へのOracle [Java Message Service](#) (JMS)インタフェースについて説明します。

- [JMSおよびOracle JMSの一般的な機能](#)
- [JMSでの構造化ペイロード/メッセージの型](#)
- [JMSのバッファ済メッセージ](#)
- [JMS Point-to-Pointモデル機能](#)
- [JMSパブリッシュ・サブスクライブ・モデル機能](#)
- [JMSメッセージ・プロデューサの機能](#)
- [JMSメッセージ・コンシューマ機能](#)
- [JMS伝播](#)
- [JMS AQのメッセージ変換](#)
- [JMSストリーミング](#)
- [Java EEの準拠](#)

JMSおよびOracle JMSの一般的な機能

この項の内容は次のとおりです。

- [JMSコネクションおよびセッション](#)
- [JMS宛先](#)
- [JMSでのシステム・レベルのアクセス制御](#)
- [JMSでの宛先レベルのアクセス制御](#)
- [JMSでの保存およびメッセージ履歴](#)
- [JMSでのOracle Real Application Clustersのサポート](#)
- [JMSでの統計ビューのサポート](#)

JMSコネクションおよびセッション

この項の内容は次のとおりです。

- [ConnectionFactoryオブジェクト](#)
- [AQjmsFactoryを使用したConnectionFactoryオブジェクトの取得](#)
- [JNDIを使用したConnectionFactoryオブジェクトの検索](#)
- [JMSコネクション](#)
- [JMSセッション](#)

ConnectionFactoryオブジェクト

ConnectionFactoryは、管理者によって定義された接続構成パラメータの集合をカプセル化します。クライアントはこれを使用してJMSプロバイダとの接続を確立します。この場合、Oracle Databaseの一部であるOracle JMSがJMSプロバイダです。

ConnectionFactoryオブジェクトには、次の3種類があります。

- ConnectionFactory
- QueueConnectionFactory
- TopicConnectionFactory

AQjmsFactoryを使用したConnectionFactoryオブジェクトの取得

AQjmsFactoryクラスを使用して、ConnectionFactory、QueueConnectionFactoryまたはTopicConnectionFactoryオブジェクトに対するハンドルを取得できます。

Point-to-Point操作とパブリッシュ・サブスクライブ操作の両方をサポートしているConnectionFactoryを取得するには、AQjmsFactory.getConnectionFactory()を使用します。QueueConnectionFactoryを取得するには、AQjmsFactory.getQueueConnectionFactory()を使用します。TopicConnectionFactoryを取得するには、AQjmsFactory.getTopicConnectionFactory()を使用します。

ConnectionFactory、QueueConnectionFactoryまたはTopicConnectionFactoryは、ホスト名、ポート番号、SIDドライバまたはJDBC URLおよびプロパティを使用して作成できます。

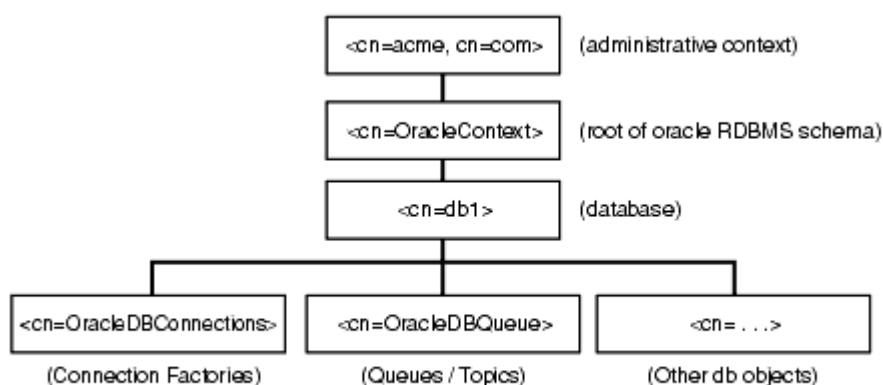
JNDIを使用したConnectionFactoryオブジェクトの検索

JMS管理者は、[Lightweight Directory Access Protocol](#)(LDAP)サーバーにConnectionFactoryオブジェクトを登録できます。JMSで[Java Naming and Directory Interface](#)(JNDI)の検索を使用可能にするには、次の設定が必要です。

1. データベースの登録

Oracle Databaseサーバーのインストール時に、データベースをLDAPサーバーに登録する必要があります。これは、[Database Configuration Assistant](#) (DBCA)を使用して実行できます。[図11-1](#)に、LDAPサーバー内のOracle Database Advanced Queuingエントリの構造を示します。ConnectionFactory情報は<cn=OracleDBConnections>の下に格納され、トピックおよびキューは<cn=OracleDBQueues>の下に格納されます。

図11-1 LDAPサーバー内のOracle Database Advanced Queuingエントリの構造



2. パラメータGLOBAL_TOPIC_ENABLEDを設定します。

データベースのGLOBAL_TOPIC_ENABLEDシステム・パラメータをTRUEに設定する必要があります。これによって、Oracle Database Advanced Queuingで作成されたすべてのキューおよびトピックが自動的にLDAPサーバーに登録されます。このパラメータは、ALTER SYSTEM SET GLOBAL_TOPIC_ENABLED = TRUEを使用して設定できます。

3. ConnectionFactoryオブジェクトを登録します

データベースを設定してLDAPサーバーが使用できるようになると、JMS管理者は `AQjmsFactory.registerConnectionFactory()` を使用して、LDAPに `ConnectionFactory`、`QueueConnectionFactory` および `TopicConnectionFactory` の各オブジェクトを登録できます。

登録には次のいずれかの方法を使用します。

- 直接LDAPサーバーに接続

LDAPにコネクション・ファクトリを登録するには、ユーザーに `GLOBAL_AQ_USER_ROLE` が必要です。

LDAPに直接接続するには、`registerConnectionFactory` メソッドのパラメータに、LDAPコンテキスト、`ConnectionFactory` の名前、`QueueConnectionFactory` の名前、`TopicConnectionFactory` の名前、ホスト名、データベースSID、ポート番号、[JDBCドライバ](#) (ThinまたはOCI8) およびファクトリ・タイプ (キューまたはトピック) を含める必要があります。

- データベース・サーバーを介してLDAPに接続

ユーザーはOracle Databaseにログインしてから、データベースによるLDAPエントリの更新ができます。ユーザーが、データベースにログインしてこの操作を実行するには、`AQ_ADMINISTRATOR_ROLE` を取得する必要があります。

データベースを介してLDAPに接続するには、`registerConnectionFactory` メソッドのパラメータに、`AQ_ADMINISTRATOR_ROLE` を持つユーザーへのJDBCコネクション、`ConnectionFactory` の名前、`QueueConnectionFactory` の名前、`TopicConnectionFactory` の名前、ホスト名、データベースSID、ポート番号、JDBCドライバ (ThinまたはOCI8) およびファクトリ・タイプ (キューまたはトピック) を含める必要があります。

JMSコネクション

JMSコネクションは、クライアントとJMSプロバイダ間のアクティブなコネクションを表します。JMSコネクションによって、次に示すいくつかの重要なサービスが実行されます。

- JMSプロバイダとのオープン・コネクションまたはコネクション・プールのいずれかをカプセル化します。
- クライアントとプロバイダのサービス・デーモン間のオープンなTCP/IPソケットを表します。
- コネクション確立時に、クライアントを認証するための構造を提供します。
- Sessionsを作成します
- コネクション・メタデータを提供します。
- オプションの `ExceptionHandler` をサポートします。

データベースへのJMSコネクションは、`createConnection()`、`createQueueConnectionFactory()` または `createTopicConnectionFactory()` を起動し、`ConnectionFactory` オブジェクト、`QueueConnectionFactory` オブジェクトまたは `TopicConnectionFactory` オブジェクトにそれぞれパラメータ `username` および `password` を渡して作成できます。

次に、`Connection` オブジェクトでサポートされているメソッドのいくつかを示します。

- `start()`

このメソッドは、着信メッセージの配信を開始または再開します。

- `stop()`

このメソッドは、着信メッセージの配信を一時停止します。`Connection` オブジェクトが停止すると、すべてのメッセージ・

コンシューマへの配信は禁止されます。また、[同期](#)受信のブロックおよびメッセージは、メッセージ・リスナーに配信されません。

- `close()`

このメソッドは、[JMS](#)セッションをクローズして、関連付けられたすべてのリソースを解放します。

- `createSession(true, 0)`

このメソッドは、JMS Connectionインスタンスを使用してJMS Sessionを作成します。

- `createQueueSession(true, 0)`

このメソッドはQueueSessionを作成します。

- `createTopicSession(true, 0)`

このメソッドはTopicSessionを作成します。

- `setExceptionHandler(ExceptionListener)`

このメソッドは、Connectionの例外リスナーを設定します。問題が非同期でクライアントに通知されます。メッセージを処理するのみのConnectionの場合、その失敗を確認する他の方法はありません。

- `getExceptionHandler()`

このメソッドは、このConnectionのExceptionHandlerを取得します。

JMSクライアントは、通常1つのConnection、1つのSession、複数のMessageProducerおよびMessageConsumerオブジェクトを作成します。現在のバージョンでは、次の場合を除いて、1つのConnectionにつき1つのオープンSessionのみが許可されています。

- JDBC OCI8ドライバを使用して[JMSコネクション](#)を作成する場合
- ユーザーが、JMSコネクションの作成時にOracleOCIConnectionPoolインスタンスを提供する場合

Connectionは停止モードで作成されます。この状態では、メッセージを受信できません。通常、設定が完了するまで、Connectionを停止モードにしておきます。その時点でConnection start()メソッドがコールされ、メッセージがConnectionコンシューマに配信され始めます。このような設定規則を確立することにより、クライアントが設定されている最中の[非同期](#)メッセージ配信の結果、発生するクライアントの混乱を最小限に止められます。

Connectionを起動し、続けて設定を実行できます。これを実行するクライアントは、設定処理中に非同期メッセージの配信処理の準備をする必要があります。MessageProducerは、Connectionが停止中でもメッセージを[送信](#)できます。

JMSセッション

JMS Sessionは、メッセージの作成および処理用の単一スレッドのコンテキストです。これは、[Java Virtual Machine](#)(JVM)の外でプロバイダ・リソースを割り当てますが、軽量なJMSオブジェクトとみなされます。

Sessionには、次の役割があります。

- MessageProducerおよびMessageConsumerオブジェクトのファクトリを構成します。
- 宛先オブジェクト(キュー/トピック)に対するハンドルの取得方法を提供します。
- プロバイダが最適化したメッセージ・ファクトリを提供します。
- セッションのMessageProducerおよびMessageConsumerオブジェクトにまたがる作業を組み合わせる一連のトランザクションをサポートし、これらを単位に構成します。
- セッションが処理および作成するメッセージのシリアル順序を定義します。
- セッションに登録されたMessageListenerオブジェクトの実行をシリアル化します。

Oracle Database 10gでは、JDBC thinまたはJDBC thick(OCI)ドライバのいずれかを使用すると、単一のJMS Connectionを使用して、リソースが許すかぎり多くのJMS Sessionを作成できます。

プロバイダが、JVM外のSessionのためのリソースを割り当てることができるため、クライアントはリソースが必要ないときはこれらを閉じる必要があります。ガベージ・コレクションによる最終的なリソースの解放を待つ必要はありません。Sessionが作成したMessageProducerおよびMessageConsumerオブジェクトについても同様です。

Sessionオブジェクトのメソッドには、次のものが含まれます。

- `commit()`
このメソッドは、トランザクションで実行されるすべてのメッセージをコミットして、現在保持されているロックを解放します。
- `rollback()`
このメソッドは、トランザクションで実行されたすべてのメッセージをロールバックして、現在保持されているロックを解放します。
- `close()`
このメソッドはSessionをクローズします。
- `getDBConnection()`
このメソッドは、基礎となるJDBCコネクションに対するハンドルを取得します。このハンドルを使用して、他のSQL [DML](#) 操作を同じSessionの一部として実行できます。このメソッドは、Oracle JMSに固有です。
- `acknowledge()`
このメソッドは、非トランザクション・セッションでメッセージの受信を承認します。
- `recover()`
このメソッドは、非トランザクション・セッションでメッセージの配信を再開します。セッション中に配信された一連のメッセージは、最後に承認されたメッセージの後にリセットされます。

Oracle JMSの拡張例は、次のとおりです。

- `createQueueTable()`
このメソッドは[キュー表](#)を作成します。
- `getQueueTable()`
このメソッドは既存のキュー表に対するハンドルを取得します。
- `createQueue()`
このメソッドはキューを作成します。
- `getQueue()`
このメソッドは既存のキューに対するハンドルを取得します。
- `createTopic()`
このメソッドはトピックを作成します。
- `getTopic()`
このメソッドは既存のトピックに対するハンドルを取得します。

どの拡張機能を使用する場合も、SessionオブジェクトをAQJMSSessionにキャストする必要があります。

注意:

JMS仕様では、開始されていない JMS Connection インスタンスで受信が完了したとき、プロバイダにより NULL メッセージが返されることを要求します。



javax.jms.Connection インスタンスの作成後、start() メソッドをそのインスタンスにコールしてメッセージを受信できるようにする必要があります。コネクション確立後で実際に受信する前に、t_conn.start(); などの行を追加するとメッセージを受信できます。

JMS宛先

Destinationは、クライアントがメッセージの送信先および受信元の指定に使用するオブジェクトです。DestinationオブジェクトはQueueまたはTopicに指定できます。Oracle Database Advanced Queuingでは、特定のデータベースの *schema.queue* にマップされます。Queueは単一コンシューマ・キューに、Topicはマルチ・コンシューマ・キューにマップします。

JMSセッションを使用したDestinationオブジェクトの取得

Destinationオブジェクトは、Sessionオブジェクトからドメイン固有の次のSessionメソッドを使用して作成されます。

- AQjmsSession.getQueue(queue_owner, queue_name)

このメソッドはJMSキューに対するハンドルを取得します。

- AQjmsSession.getTopic(topic_owner, topic_name)

このメソッドは[JMSトピック](#)に対するハンドルを取得します。

JNDIを使用したDestinationオブジェクトの検索

LDAPサーバーに[スキーマ](#)オブジェクトを登録するようにデータベースを設定できます。データベースがLDAPを使用できるように設定され、GLOBAL_TOPIC_ENABLEDパラメータがTRUEに設定されている場合、すべてのJMSキューおよびトピックは、作成時に自動的にLDAPサーバーに登録されます。管理者は、LDAPに登録されたキューおよびトピックの別名を作成することもできます。LDAPに登録されたキューおよびトピックは、JNDIを介してキューまたはトピックの名前または別名を使用してルックアップできます。

関連項目:

[LDAPサーバーへの別名の追加](#)

JMS宛先メソッド

Destinationオブジェクトのメソッドには、次のものが含まれます。

- alter()

このメソッドはキューまたはトピックを変更します。

- schedulePropagation()

このメソッドは、ソースから宛先への[伝播](#)をスケジューリングします。

- unschedulePropagation()

このメソッドは、スケジュール済伝播のスケジュールを解除します。

- `enablePropagationSchedule()`

このメソッドは、伝播スケジュールを有効化します。

- `disablePropagationSchedule()`

このメソッドは、伝播スケジュールを無効化します。

- `start()`

このメソッドはキューまたはトピックを開始します。キューは、エンキューまたは[デキュー](#)を可能にするために開始されます。トピックは、パブリッシュまたはサブスクライブを可能にするために開始できます。

- `stop()`

このメソッドはキューまたはトピックを停止します。キューは、エンキューまたはデキューを実行不可にするために停止できます。トピックは、パブリッシュまたはサブスクライブを実行不可にするために停止できます。

- `drop()`

このメソッドはキューまたはトピックを削除します。

JMSでのシステム・レベルのアクセス制御

Oracle8i以上では、すべてのキューイング操作に対してシステム・レベルのアクセス制御をサポートします。この機能によって、アプリケーション設計者またはDBAは、ユーザーをキュー管理者にできます。キュー管理者は、データベースのどのキューに対してもJMSインタフェース(管理および操作)を起動できます。これによって、データベース上のキュー全体に対するすべての管理スクリプトを1つのスキーマで管理できるため、管理作業が容易になります。

メッセージが宛先キューに届くと、ソース・キューのスキーマ名に基づいたセッションによって、新しく届いたメッセージが宛先キューにエンキューされます。つまり、ソース・キューのスキーマに宛先キューに対するエンキュー権限を付与する必要があります。

リモートの宛先キューに伝播するために、エージェント構造体のアドレス・フィールドのデータベース・リンクに指定されたログイン・ユーザーには、ENQUEUE_ANY権限を付与するか、または宛先キューに対するエンキュー権限を付与する必要があります。ただし、データベース・リンクのログイン・ユーザーが宛先のキュー表を所有している場合はどのような明示的な権限も付与する必要はありません。

関連項目:

[Oracle Enterprise Managerのサポート](#)

JMSでの宛先レベルのアクセス制御

Oracle8i以上では、エンキューおよびデキュー操作に対してキュー・レベルまたはトピック・レベルのアクセス制御をサポートしています。この機能によって、アプリケーション設計者は、あるスキーマに作成されたキューおよびトピックを他のスキーマで実行中のアプリケーションから保護できます。そのキューまたはトピックが属するスキーマの外で実行しているアプリケーションには、最小限のアクセス権限のみを付与できます。キューまたはトピックに対するアクセス権限として、ENQUEUE、DEQUEUEおよびALLがサポートされています。

関連項目:

[Oracle Enterprise Managerのサポート](#)

JMSでの保存およびメッセージ履歴

メッセージは、相互に関連していることがよくあります。たとえば、あるメッセージを処理した結果として他のメッセージが生成された場合、両者は関連付けられています。アプリケーション設計者としては、そのような関連を追跡することが必要な場合があります。Oracle Database Advanced Queuingでは、ユーザーがメッセージをキュー表に保持し、分析のためにSQLで問合せできます。

保存機能およびメッセージ識別子とともに、メッセージ・ジャーナルがOracle Database Advanced Queuingによって自動作成され、追跡ジャーナルまたはイベント・ジャーナルをコールできます。保存、メッセージ識別子およびSQL問合せの協調によって、強力なメッセージ・ウェアハウスを構築できます。

JMSでのOracle Real Application Clustersのサポート

Oracle Database 12cリリース1 (12.1)では、Advanced Queuingに、高性能でスケラブルなJMSシャード・キューが導入されました。シャード・キューとは、システムで管理されているパーティション化を使用して、独立した複数の物理キューに分割される1つの論理キューです。シャード・キューは、Oracle RACインスタンス全体で使用されるキュー、エンキュー率またはデキュー率の高いキュー、またはサブスクリバの多いキューについて推奨されるJMSキューです。詳細は、[「シャード・キューおよびOracle Real Application Clusters \(Oracle RAC\)」](#)を参照してください。

非シャード・キューの場合は、Oracle Real Application Clusters (Oracle RAC)を使用して、各インスタンスを別々のキューで管理できるようにすることで、Oracle Database Advanced Queuingのパフォーマンスを向上できます。このためには、キューを格納するキュー表に様々なインスタンス・アフィニティ(作業環境)を指定します。これによって、様々なキュー/トピックに対するキュー操作(エンキューまたはデキュー)またはトピック操作([パブリッシュ・サブスクリブ](#))を並行して行うことができるようになります。

Oracle Database Advanced Queuingのキュー・モニター・プロセスは、キュー表のインスタンス・アフィニティを継続的に監視します。キュー・モニターは指定されたプライマリ・インスタンスが使用可能な場合はそれにキュー表の所有権を割り当て、失敗した場合は指定されたセカンダリ・インスタンスに割り当てます。

キュー表を所有しているインスタンスが終了すると、キュー・モニターはセカンダリ・インスタンスなどの適切なインスタンスに所有者を変更します。

Oracle Database Advanced Queuingの伝播はOracle Real Application Clustersでも使用できますが、これはユーザーに対して透過的です。伝播スケジュールのジョブ・アフィニティは、それぞれのキュー表のアフィニティと同じ値に設定されます。このように、キュー表を所有するインスタンスに対応付けられたjob_queue_processは、そのキュー表に格納されているキューからの伝播を処理し、ping操作を最小限に抑えます。

関連項目:

- [シャード・キュー](#)
- [キューの伝播のスケジューリング](#)
- [『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)

JMSでの統計ビューのサポート

各インスタンスは、それぞれのOracle Database Advanced Queuing統計情報を[システム・グローバル領域\(SGA\)](#)に所

有し、他のインスタンスによって収集された統計については認識しません。そのため、インスタンスがGV\$AQビューを問い合わせると、その他のすべてのインスタンスからそれぞれのその時点の統計情報が問合せ元のインスタンスに集まります。

GV\$AQビューは、待機中、準備完了または期限切れの各メッセージ数を必要なときにいつでも問い合わせることができます。また、メッセージが処理されるまでの平均待機秒数も表示します。

関連項目:

[V\\$AQ: データベース内の状態ごとのメッセージ数](#)

JMSでの構造化ペイロード/メッセージの型

JMSメッセージは、ヘッダー、プロパティおよび本体で構成されています。

ヘッダーは、クライアントおよびプロバイダの両方でメッセージの識別およびルーティングに使用される値が含まれるヘッダー・フィールドで構成されています。すべてのメッセージは、同じ一連のヘッダー・フィールドをサポートしています。

プロパティは、オプションのヘッダー・フィールドです。JMSによって定義された標準プロパティの他に、プロバイダ固有およびアプリケーション固有のプロパティを含めることもできます。

本体はメッセージ・ペイロードです。JMSは、様々な型のメッセージ・ペイロードおよびJMSで指定されたメッセージ型のJMSメッセージを格納できる型を定義します。

この項の内容は次のとおりです。

- [JMSメッセージ・ヘッダー](#)
- [JMSメッセージ・プロパティ](#)
- [JMSメッセージ本体](#)
- [異なるメッセージ型でのメッセージ・プロパティの使用](#)
- [Oracle JMSを使用したバッファ済メッセージ](#)

JMSメッセージ・ヘッダー

JMSメッセージ・ヘッダーには、次のフィールドが含まれます。

- **JMSDestination**
このフィールドには、メッセージの送信先が含まれます。Oracle Database Advanced Queuingでは、これは送信先キューまたはトピックに相当します。Sendメソッドの完了後にJMSによって設定されるDestination型です。
- **JMSDeliveryMode**
このフィールドは、メッセージのログが記録されるかどうかを決定します。JMSでは、PERSISTENT送信(メッセージが決まった記憶域にロギングされる)およびNONPERSISTENT送信(メッセージがロギングされない)をサポートしています。Sendメソッドの完了後にJMSによって設定されるINTEGERです。JMSでは、クライアントが指定したJMSDeliveryModeの値をオーバーライドするよう管理者がJMSを構成することが可能です。
- **JMSMessageID**
このフィールドは、プロバイダ内のメッセージを一意的に識別します。すべてのメッセージIDは先頭に文字列ID:を使用する必要があります。Sendメソッドの完了後にJMSによって設定されるString型です。

- **JMSTimeStamp**

このフィールドには、送信先のプロバイダにメッセージが渡された時間が含まれます。Oracle Database Advanced Queuingのメッセージのエンキュー時間にマッピングされます。Sendメソッドの完了後にJMSによって設定されるLong型です。

- **JMSCorrelationID**

このフィールドは、クライアントがメッセージを別のメッセージにリンクするために使用できます。JMSクライアントによって設定されるString型です。

- **JMSReplyTo**

このフィールドには、メッセージ送信時にクライアントが指定するDestination型が含まれます。クライアントはoracle.jms.AQjmsAgent、javax.jms.Queueまたはjavax.jms.Topicを使用できます。

- **JMSType**

このフィールドには、送信時にクライアントが指定するメッセージ・タイプの識別子が含まれます。これはString型です。移植性を高めるために、JMSTypeに記号値を使用することをお勧めします。

- **JMSExpiration**

このフィールドは、非Java EE準拠モードでは、エンキュー時間とTimeToLiveの合計です。準拠モードでは、デキューされたメッセージのJMSExpirationヘッダー値は、メッセージのエンキュー時のJMSTimeStamp (ミリ秒単位のグリニッジ標準時)とTimeToLive (ミリ秒単位)の合計です。Sendメソッドの完了後にJMSによって設定されるLong型です。JMSでは、クライアントが指定したJMSExpirationの値をオーバーライドするよう管理者がJMSを構成することが可能です。

- **JMSPriority**

このフィールドには、メッセージの優先度が含まれます。Sendメソッドの完了後にJMSによって設定されるINTEGERです。Java EE準拠モードでは、優先度を使用できる値は0から9で、9の優先度が最も高く、4がデフォルトで、Sun Microsystems JMS 1.1標準に準拠しています。デフォルトは非準拠モードです。JMSでは、クライアントが指定したJMSPriorityの値をオーバーライドするよう管理者がJMSを構成することが可能です。

- **JMSRedelivered**

このフィールドは、JMSプロバイダによって設定されたブールです。

関連項目:

[Java EEの準拠](#)

JMSメッセージ・プロパティ

JMSプロパティは、クライアントによって明示的に設定されるか、またはJMSプロバイダによって自動的に設定されます(通常、これらは読み取り専用です)。JMSプロパティには、SendおよびReceive操作に指定されたパラメータを使用して設定されるものもあります。

プロパティで、メッセージにオプションのヘッダー・フィールドを追加できます。プロパティによって、クライアントは、messageSelectorを使用して、クライアントのかわりにJMSプロバイダにアプリケーション固有基準を使用してメッセージを選択させることができます。プロパティ名は文字列で、値はBoolean、byte、short、int、long、float、doubleおよびstringです。

JMS定義のプロパティは、すべてJMSXで始まります。次のものがあります。

- **JMSXUserID**

このフィールドは、メッセージの送信元ユーザーの識別情報です。Sendメソッドの完了後にJMSによって設定されるString型です。

- **JMSXAppID**

このフィールドは、メッセージの送信元アプリケーションの識別情報です。Sendメソッドの完了後にJMSによって設定されるString型です。

- **JMSXDeliveryCount**

このフィールドは、メッセージの送信試行回数です。Sendメソッドの完了後にJMSによって設定されるIntegerです。

- **JMSXGroupid**

このフィールドは、メッセージが属するメッセージ・グループの識別情報です。JMSクライアントによって設定されるString型です。

- **JMSXGroupSeq**

このフィールドは、グループ内のメッセージの順序番号です。JMSクライアントによって設定されるInteger型です。

- **JMSXRcvTimeStamp**

このフィールドは、コンシューマにメッセージが配信された時間(デキュー時間)です。Receiveメソッドの完了後にJMSによって設定されるString型です。

- **JMSXState**

このフィールドは、プロバイダによって設定されたメッセージ状態です。メッセージ状態は、WAITING、READY、EXPIREDまたはRETAINEDです。

Oracle固有のJMSプロパティは、すべてJMS_Oracleで始まります。次のものがあります。

- **JMS_OracleExcpQ**

このフィールドは、元の宛先に配信できないメッセージを送信するキュー名です。JMSクライアントによって設定されるString型です。EXCEPTIONタイプの宛先のみをJMS_OracleExcpQプロパティで指定できます。

- **JMS_OracleDelay**

このフィールドは、メッセージの配信を遅延する時間(秒単位)です。JMSクライアントによって設定されるInteger型です。これは、メッセージの配信順序に影響を与える可能性があります。

- **JMS_OracleOriginalMessageId**

このフィールドは、メッセージがある宛先から別の宛先に伝播された場合に、ソースのメッセージのメッセージ識別子に設定されます。JMSプロバイダによって設定されるString型です。メッセージが伝播されていない場合、このプロパティはJMSMessageIdと同じ値です。

クライアントは、プロパティを定義することによって、メッセージにヘッダー・フィールドを追加できます。これらのプロパティをmessageSelectorで使用して特定のメッセージを選択できます。

JMSメッセージ本体

JMSでは、次の5つのフォーマットのメッセージ本体が提供されます。

- [StreamMessage](#)

- [BytesMessage](#)

- [MapMessage](#)

- [TextMessage](#)

- [ObjectMessage](#)
- [AdtMessage](#)

StreamMessage

StreamMessageオブジェクトは、Javaプリミティブのストリームの送信に使用されます。格納と読取りは順次的に行われます。これはMessageから継承し、StreamMessage本文を追加します。メソッドは主にjava.io.DataInputStreamとjava.io.DataOutputStreamのメソッドに基づいています。

基本データ型の値は、それぞれの型のメソッドを使用して、明示的に読み込みまたは書き込みできます。抽象的なオブジェクトとして読み込みまたは書き込みすることもできます。StreamMessageオブジェクトを使用するには、SYS.AQ\$_JMS_STREAM_MESSAGEまたはAQ\$_JMS_MESSAGEペイロード型を持つキュー表を作成します。

StreamMessageオブジェクトは、[表11-1](#)に示す変換をサポートします。行の型として書き込まれる値は、列の型として読み込むことができます。

表11-1 StreamMessageの変換

入力	Boolean	byte	short	char	int	long	float	double	String	byte[]
Boolean	X	-	-	-	-	-	-	-	X	-
byte	-	X	X	-	X	X	-	-	X	-
short	-	-	X	-	X	X	-	-	X	-
char	-	-	-	X	-	-	-	-	X	-
int	-	-	-	-	X	X	-	-	X	-
long	-	-	-	-	-	X	-	-	X	-
float	-	-	-	-	-	-	X	X	X	-
double	-	-	-	-	-	-	-	X	X	-
string	X	X	X	X	X	X	X	X	X	-
byte[]	-	-	-	-	-	-	-	-	-	X

BytesMessage

BytesMessageオブジェクトは、1つの未解釈バイトのストリームを含むメッセージを送信する場合に使用します。これはMessageから継承し、BytesMessage本文を追加します。メッセージの受信者が、そのバイト列を解析します。メソッドは主にjava.io.DataInputStreamとjava.io.DataOutputStreamのメソッドに基づいています。

これは、クライアントが既存のメッセージ・フォーマットをコード化するためのメッセージ型です。可能であれば、かわりに他の自己定

義メッセージ型を使用してください。

基本データ型の値は、それぞれの型のメソッドを使用して、明示的に書込みができます。また、抽象的なオブジェクトとして書き込むこともできます。BytesMessageオブジェクトを使用するには、SYS. AQ\$_JMS_BYTES_MESSAGEまたはAQ\$_JMS_MESSAGEペイロード型を持つキュー表を作成します。

MapMessage

MapMessageオブジェクトは、名前がString型で値がJava基本データ型である名前/値ペアの集合を送信する場合に使用します。このエントリには、名前を指定して順次的またはランダムにアクセスできます。エントリの順序は定義されません。これはMessageから継承し、MapMessage本文を追加します。基本データ型の値は、それぞれの型のメソッドを使用して、明示的に読み込みまたは書込みできます。抽象的なオブジェクトとして読み込みまたは書込みすることもできます。

MapMessageオブジェクトを使用するには、SYS. AQ\$_JMS_MAP_MESSAGEまたはAQ\$_JMS_MESSAGEペイロード型を持つキュー表を作成します。MapMessageオブジェクトは、[表11-2](#)に示す変換をサポートします。この表の「X」は、行の型として書き込まれる値を列の型として読み取ることができることを意味します。

表11-2 MapMessageの変換

入力	Boolean	byte	short	char	int	long	float	double	String	byte[]
Boolean	X	-	-	-	-	-	-	-	X	-
byte	-	X	X	-	X	X	-	-	X	-
short	-	-	X	-	X	X	-	-	X	-
char	-	-	-	X	-	-	-	-	X	-
int	-	-	-	-	X	X	-	-	X	-
long	-	-	-	-	-	X	-	-	X	-
float	-	-	-	-	-	-	X	X	X	-
double	-	-	-	-	-	-	-	X	X	-
string	X	X	X	X	X	X	X	X	X	-
byte[]	-	-	-	-	-	-	-	-	-	X

TextMessage

TextMessageオブジェクトは、java.lang.StringBufferを含むメッセージを送信する場合に使用します。これはMessageから継承し、TextMessage本文を追加します。テキスト情報は、getText()およびsetText(...)メソッドを使用して読み込みまたは書込みができます。TextMessageオブジェクトを使用するには、SYS. AQ\$_JMS_TEXT_MESSAGEまたはAQ\$_JMS_MESSAGEペイ

ロード型を持つキュー表を作成します。

ObjectMessage

ObjectMessageオブジェクトは、シリアライズ可能なJavaオブジェクトを含むメッセージを送信する場合に使用します。これはMessageから継承し、単一のJava参照を含む本文を追加します。シリアライズ可能なJavaオブジェクトのみ使用できます。Javaオブジェクトのコレクションを送信する必要がある場合、JDK 1.4で提供されているコレクション・クラスのいずれかを使用できます。オブジェクトは、getObject () メソッドおよび setObject (...) メソッドを使用して読み込みまたは書き込みできます。ObjectMessageオブジェクトを使用する場合、SYS.AQ\$_JMS_OBJECT_MESSAGEまたはAQ\$_JMS_MESSAGEペイロード型でキュー表を作成します。

AdtMessage

AdtMessageオブジェクトは、Oracleのオブジェクト型に対応するJavaオブジェクトを含むメッセージを送信する場合に使用します。これらのオブジェクトはMessageから継承され、CustomDatumまたはORADaataインタフェースを実装するJavaオブジェクトを含む本体を追加します。

AdtMessageオブジェクトを使用するには、Oracleのオブジェクト型としてのペイロード型を持つキュー表を作成します。

AdtMessageのペイロードは、getAdtPayloadおよびsetAdtPayloadメソッドを使用して、読み取りおよび書き込みができます。

AdtMessageオブジェクトを使用してSYS.XMLType型のキューへメッセージを送信することもできます。oracle.xdb.XMLTypeクラスを使用してメッセージを作成する必要があります。

AdtMessageオブジェクトの場合、クライアントが次を取得できます。

- JMSXDeliveryCount
- JMSXRecvTimeStamp
- JMSXState
- JMS_OracleExcpQ
- JMS_OracleDelay

関連項目:

CustomDatumインタフェースおよびORADaataインタフェースの詳細は、[『Oracle Database Java開発者ガイド』](#)を参照してください。

異なるメッセージ型でのメッセージ・プロパティの使用

次のメッセージ・プロパティは、クライアントがsetPropertyコールを使用して設定できます。StreamMessage、BytesMessage、ObjectMessage、TextMessageおよびMapMessageオブジェクトの場合、クライアントによって次の設定ができます。

- JMSXAppID
- JMSXGroupID
- JMSXGroupSeq
- JMS_OracleExcpQ
- JMS_OracleDelay

AdtMessageオブジェクトの場合、クライアントによって次の設定ができます。

- JMS_OracleExcpQ

- JMS_OracleDelay

次のメッセージ・プロパティは、クライアントがgetPropertyコールを使用して取得できます。StreamMessage、BytesMessage、ObjectMessage、TextMessageおよびMapMessageオブジェクトの場合、クライアントが次を取得できます。

- JMSXUserID
- JMSXAppID
- JMSXDeliveryCount
- JMSXGroupID
- JMSXGroupSeq
- JMSXRecvTimeStamp
- JMSXState
- JMS_OracleExcpQ
- JMS_OracleDelay
- JMS_OracleOriginalMessageID

Oracle JMSを使用したバッファ済メッセージ

メッセージの送信時にdeliveryModeとしてNON_PERSISTENTを指定すると、非永続JMSメッセージを送信できます。JMS非永続メッセージは決まった記憶域に記録する必要がないため、JMSシステム障害が発生すると消失する可能性があります。JMS非永続メッセージはOracle Database Advanced Queuingで使用可能なバッファ済メッセージに類似していますが、両者には重要な相違点も存在します。

注意:



Oracle JMS の非永続メッセージを、Oracle Database 10g リリース 2(10.2)で非推奨になった Oracle Database Advanced Queuing の非永続キューと混同しないでください。

関連項目:

- [バッファ済メッセージ](#)
- [非永続キュー](#)

トランザクションのコミットとクライアントの確認

JMSのdeliveryModeは、メッセージのトランザクション属性と直交します。JMS非永続メッセージは、処理済セッションまたは非処理済セッションで送受信できます。JMS非永続メッセージを処理済セッションで送受信する場合、JMS操作の効果は処理済セッションのコミット後にのみ参照可能です。CLIENT_ACKNOWLEDGE確認モードの非処理済セッションで受信する場合、このメッセージの受信による効果は、クライアントがメッセージを確認した後にのみ参照可能です。確認されない場合、メッセージは削除されず、クライアントがSession.recoverをコールすると再配信されます。

これに対して、Oracle Database Advanced Queuingのバッファ済メッセージでは、このようなトランザクションまたは確認の概念はサポートされません。バッファ済メッセージの送受信は、どちらもIMMEDIATE可視性モードで実行する必要があります。したがって、セッションがコミットされたかどうかやメッセージが確認されたかどうかは関係なく、ユーザーは送受信操作の効果を即時に参照できます。

各種API

通常のJMS送信およびパブリッシュ・メソッドで送信されるメッセージは、Oracle Database Advanced Queuingでは永続メッセージとして扱われます。通常のJMS受信メソッドは、AQ永続メッセージのみを受信します。バッファ済メッセージを送受信するには、Oracle拡張APIであるbufferSend、bufferPublishおよびbufferReceiveを使用する必要があります。

関連項目:

bufferSend、bufferPublishおよびbufferReceiveの詳細は、[Oracle Databaseアドバンスド・キューイングJava APIリファレンス](#)を参照してください

ペイロード制限

バッファ済メッセージのOracle Database Advanced Queuing実装では、LOB属性はサポートされていません。このため、5種類の標準JMSメッセージのペイロードには次の制限が適用されます。

- JMS TextMessageのペイロードは4000バイト以内です。
Oracle JMSのキャラクタ・セット変換時には、テキスト・ペイロードをデータベースに格納するためにVARCHARのかわりにCLOBを使用するように控え目な選択が必要になる場合があるため、データベース・キャラクタ・セットによってはこの上限値がさらに低くなる場合があります。
- JMS BytesMessageのペイロードは2000バイト以内です。
- JavaでシリアライズされたJMS ObjectMessage、StreamMessageおよびMapMessageデータは、2000バイト以内である必要があります。
- 他のすべてのOracle JMS ADTメッセージの場合、対応するOracleデータベースADTにLOB属性を含めることはできません。

各種定数

[表11-3](#)に示すように、Oracle Database Advanced QueuingとOracle JMSのAPIでは、バッファ済メッセージと永続メッセージの指定に使用される数値が異なります。

表11-3 Oracle Database AQとOracle JMSのバッファ済メッセージ定数

API	永続メッセージ	バッファ済メッセージ
Oracle Database Advanced Queuing	PERSISTENT := 1	BUFFERED := 2
Oracle JMS	PERSISTENT := 2	NON_PERSISTENT := 1

JMSのバッファ済メッセージ

バッファ済メッセージでは、JMSメッセージ標準が完全にサポートされています。Oracle JMSでは、それらの標準がいくつかの方法で拡張されています。

関連項目:

[バッファ済メッセージ](#)

JMSバッファ済メッセージのエンキュー

Oracle JMSでは、永続メッセージおよびバッファ済メッセージを同じJMSキュー/トピックにエンキューできるように、個別のメッセージにJMSDeliveryModeを設定することによって、アプリケーションはバッファ済メッセージを送信できます。

Oracle JMSのバッファ済メッセージは、エンキュー時刻、優先順位またはその両方で順序付けできます。順序付けは異なるメッセージ・タイプ間では適用されません。このため、たとえば、後に送信された永続メッセージが前に送信されたバッファ済メッセージより前に配信されることがあります。Oracle JMSのバッファ済メッセージでは、期限切れもサポートされます。

関連項目:

[JMSメッセージ・ヘッダー](#)

JMSバッファ済メッセージのデキュー

JMSでは、JMSサブスクライバが両方のメッセージ・タイプを対象にできるように、サブスクライバが永続メッセージのみまたはバッファ済メッセージのみを対象とするように宣言することを要求しません。

Oracle JMSは、JMSMessageIDによる高速で効率的なメッセージのデキュー、メッセージ・ヘッダーのセレクト、およびメッセージ・プロパティのセレクトをサポートしています。Oracle JMSデキュー・コールは、永続メッセージとバッファ済メッセージの両方をチェックします。

注意:



Oracle JMS 永続メッセージには一意のメッセージ識別子があります。Oracle JMS バッファ済メッセージ識別子は、キュー/トピック内でのみ一意です。

同時デキュー・プロセスが同じサブスクライバの同じキューからデキューしている場合は、他のプロセスによってロックされているメッセージがスキップされます。

関連項目:

- [MessageSelector](#)
- [メッセージの受信](#)

トランザクションのサポート

バッファ済メッセージがトランザクション・セッションでエンキューされる場合、JMSではそれらに対するトランザクション・サポートが必要となります。Oracle JMSでは、バッファ済メッセージに関連するトランザクション・セッションで次の標準が満たされることが保証されます。

- 原子性

Oracle JMSトランザクション内の永続メッセージおよびバッファ済メッセージは、アトミックにコミットまたはロールバックされます。バッファ済メッセージがディスクに書き込まれた場合(LOBに関連するメッセージの場合など)でも、ロールバックによってそれらが削除されます。

- 一貫性

永続メッセージ操作およびバッファ済メッセージ操作がトランザクションでインターリーブされる場合、すべてのOracle JMSユーザーは影響を受けるキュー/トピックの一貫性のあるビューを共有します。トランザクションによってエンキューされたすべての永続メッセージおよびバッファ済メッセージは、コミット時に表示できるようになります。プロセスがトランザクションの途中で終了した場合は、永続メッセージおよびバッファ済メッセージの両方が元に戻されます。Oracle JMSユーザーには、トランザクションのすべての永続メッセージおよびバッファ済メッセージが表示されるか、それらのいずれも表示されません。

- 分離

トランザクションのバッファ済エンキュー操作は、トランザクションがコミットされるまではオーナー・トランザクションにのみ表示されます。トランザクションがコミットされると、すべてのコンシューマに表示されます。

デキュー・トランザクションによってロックされているメッセージはブラウズできます。

確認メッセージの受信

非トランザクション・セッションの確認メッセージ受信のためのack_modeパラメータには、3つの値が定義されています。

- DUPS_OK_ACKNOWLEDGE

このモードでは、重複メッセージが許可されます。

- AUTO_ACKNOWLEDGE

このモードでは、セッションはメッセージを自動的に確認します。

- CLIENT_ACKNOWLEDGE

このモードでは、メッセージ・プロデューサ確認メソッドをコールすることによって、クライアントはメッセージを明示的に確認します。メッセージを確認すると、以前に処理されたすべてのメッセージが確認されます。

関連項目:

[セッションの作成](#)

バッファ済メッセージのサービス品質

JMSは、伝播されていないバッファ済メッセージを最大1回配信することをサポートするようにプロバイダに要求しています。バッファ済メッセージのリカバリが無効にされている場合、Oracle JMSはこの標準を満たしています。

現在のメッセージ伝播の実装では、メッセージの重複配信が行われることがあります。ただし、メッセージの伝播はOracle JMSによって提供される拡張であるため、これはJMS標準に違反しません。

関連項目:

バッファ済メッセージが重複配信される原因については、[バッファ済メッセージの伝播](#)を参照してください。

バッファ済メッセージでサポートされるJMSタイプ

Oracle JMSは、JMSで定義されているタイプをOracleのユーザー定義タイプにマップし、JMSメッセージを格納するためにそれらのユーザー定義タイプのキューを作成します。これらのタイプの一部にはLOB属性があり、メッセージが永続またはバッファ済のいずれであってもOracle JMSはディスクに書き込みます。

たとえば、JMSタイプJMSTextMessageのユーザー定義タイプSYS.AQ\$_JMS_TEXT_MESSAGEでは、4kより小さいテキスト文字列がVARCHAR2列に格納されます。ただし、4kより大きいテキスト文字列を格納するためのCLOB属性があります。

JMSメッセージは4kより大きい場合がよくあるため、Oracle JMSは大きいメッセージをメモリーに格納できる新しいADTを提供しています。ADTのディスク表示は変わりませんが、いくつかのVARCHAR2/RAW属性では最大100kのサイズのJMSメッセージをメモリーに格納できます。100kより大きいメッセージもバッファ済メッセージとしてパブリッシュできますが、ディスクに書き込まれます。

関連項目:

[バッファ済メッセージのエンキュー](#)

JMS Point-to-Pointモデル機能

Point-to-Pointモデルでは、クライアントは、1つのポイントから別のポイントへメッセージを交換します。メッセージのプロデューサとコンシューマは、シングル・コンシューマ・キューを使用してメッセージを送受信します。管理者は、AQjmsSessionのcreateQueueメソッドを使用して、シングル・コンシューマ・キューを作成します。キューを使用する前に、AQjmsDestinationでstartコールを使用して、キューをエンキュー/デキューに対して有効にする必要があります。クライアントは、AQjmsSessionのgetQueueメソッドを使用して事前に作成されたキューに対するハンドルを取得します。

シングル・コンシューマ・キューの場合、メッセージは1つのコンシューマが1度のみ処理できます。同じキューから同時にデキューするプロセスまたはオペレーティング・システム・スレッドが複数存在する場合、各プロセスはキューの先頭にあるロックされていない最初のメッセージをデキューします。ロックを作成したプロセス以外のプロセスは、ロックされたメッセージをデキューできません。

処理が済むと、キューの保存期間が0(ゼロ)の場合はそのメッセージは削除され、そうでない場合は指定された期間保存されます。メッセージが保存されている間は、キュー表ビューに対してSQLを使用して問い合わせたり、QueueBrowserで処理済メッセージのメッセージ識別子を指定してデキューできます。

QueueSender

クライアントは、QueueSenderを使用して、キューにメッセージを送信します。これは、クライアントのSessionでキューをcreateSenderメソッドに渡すことによって作成されます。また、クライアントにはキューを指定しないで、QueueSenderを作成するオプションがあります。この場合、キューを送信操作のたびに指定する必要があります。

クライアントは、QueueSenderによって送信されたすべてのメッセージのデフォルト配信モード、優先順位およびTimeToLiveを指定できます。または、クライアントはこれらのオプションを各メッセージに対して定義できます。

QueueReceiver

クライアントはQueueReceiverを使用してキューからメッセージを受信します。これは、クライアントのSessionでcreateQueueReceiverを使用して作成されます。messageSelectorがあってもなくても作成できます。

QueueBrowser

QueueBrowserを使用すると、クライアントはメッセージを削除しないで、キュー上でメッセージを参照できます。このブラウザ用メソッドは、キュー内のメッセージをスキャンするために使用されるjava.util.Enumerationを戻します。nextElementに対する最初のコールが、キューのスナップショットを取得します。QueueBrowserは、messageSelectorを使用して作成しても、使用せずに作成してもかまいません。

QueueBrowserも、メッセージをスキャン中にオプションでロックできます。これは、メッセージに対するSELECT...for UPDATEコマンドの場合と似ています。これによって、他のコンシューマがスキャン中のメッセージを削除することはなくなります。

MessageSelector

messageSelectorによって、クライアントは、コンシューマに配信されるメッセージをmessageSelector式と一致するメッセージに制限できるようになります。TextMessage型、StreamMessage型、BytesMessage型、ObjectMessage型またはMapMessage型のペイロードを含むキューのmessageSelectorには、次の1つ以上を持つ任意の式を含めることができます。

- 接頭辞「ID:」が付いたJMSメッセージ識別子

```
JMSMessageID = 'ID:23452345'
```

- JMSメッセージ・ヘッダー・フィールドまたはプロパティ

```
JMSPriority < 3 AND JMSCorrelationID = 'Fiction'
```

```
JMSCorrelationID LIKE 'RE%'
```

- ユーザー定義のメッセージ・プロパティ

```
color IN ('RED', 'BLUE', 'GREEN') AND price < 30000
```

AdtMessage型のペイロードを含むキューのmessageSelectorには、次の1つ以上を持つ任意の式を含めることができます。

- 接頭辞「ID:」なしのメッセージ識別子

```
msgid = '23434556566767676'
```

- 優先順位または相関識別子、あるいはその両方

```
priority < 3 AND corrid = 'Fiction'
```

- メッセージ・ペイロード

```
tab.user_data.color = 'GREEN' AND tab.user_data.price < 30000
```

JMSパブリッシュ・サブスクライブ・モデル機能

この項の内容は次のとおりです。

- [JMSパブリッシュ・サブスクライブの概要](#)
- [DurableSubscriber](#)
- [RemoteSubscriber](#)
- [TopicPublisher](#)
- [Recipient Lists](#)
- [TopicReceiver](#)
- [TopicBrowser](#)
- [JMSパブリッシュ・サブスクライブ操作の設定](#)

JMSパブリッシュ・サブスクライブの概要

JMSでは、パブリッシャ(出版者)の機能を持つアプリケーションとサブスクライバ(購読者)の役割を果たすアプリケーションとの間の

柔軟で動的な通信が可能です。アプリケーションが結合されることはなく、メッセージとその内容に基づいて相互に作用します。

メッセージの配信では、パブリッシャ・アプリケーションが明示的にメッセージ受信者を処理または管理する必要はありません。このため、パブリッシャ・アプリケーションの論理を変更しなくても、新しいサブスクライバ・アプリケーションを動的に追加できます。

同様にサブスクライバ・アプリケーションは、メッセージを送信しているパブリッシャ・アプリケーションに関係なく、メッセージの内容に基づいてメッセージを受信します。このため、サブスクライバ・アプリケーションの論理を変更しなくても、新しいパブリッシャ・アプリケーションを動的に追加できます。

サブスクライバ・アプリケーションは、メッセージ・プロパティまたはトピックのメッセージ内容に対してルールベースのサブスクリプション(予約購読)を定義することで、どのようなメッセージに関心があるのかを指定できます。システムは、ルールベースのサブスクリプションを使用して、パブリッシュされたメッセージの受信者を計算し、自動的にルーティングします。

パブリッシュ・サブスクライブ・モデルでは、メッセージはトピックに対してパブリッシュされ、トピックから受信されます。トピックは、AQjmsSessionのCreateTopic()メソッドを使用して作成されます。クライアントは、AQjmsSessionのgetTopic()メソッドを使用して事前に作成されたトピックに対するハンドルを取得できます。

DurableSubscriber

クライアントはDurableSubscriberを、クライアントのSession内でcreateDurableSubscriber()メソッドを使用して作成します。messageSelectorがあってもなくても作成できます。

messageSelectorによって、クライアントは、サブスクライバに配信されるメッセージをセレクトにマッチするメッセージに制限できます。セレクト構文の詳細は、[Oracle Databaseアドバンスド・キューイングJava APIリファレンス](#)のcreateDurableSubscriberに関する項を参照してください。

永続サブスクライバが同じ名前を使用するときのアクションは、実行時に[Oracle Java Message Service \(Oracle JMS\)](#)クライアントに設定されるJava EE準拠モードによって異なります。

非準拠モードでは、同じ名前の2つの永続的なTopicSubscriberオブジェクトが、2つの異なるトピックに対してアクティブになることができます。準拠モードでは、複数の永続サブスクライバが同じ名前を持つことは許可されません。同じトピックに対して作成された2つのサブスクライバが同じ名前を使用する場合、各サブスクライバに使用されるセレクトが異なると、DBMS_AQJMS.ALTER_SUBSCRIBER()の内部コールを使用して、基になるOracle Database Advanced Queuingサブスクリプションが変更されます。

2つのサブスクライバが同じ名前を使用し、2つの異なるトピックに対して作成される場合、同じサブスクリプション名を使用するクライアントがそれぞれサブスクリプション名を作成すると、既存のサブスクリプションは削除され新しいサブスクリプションが作成されます。

2つのサブスクライバが同じ名前を使用し、2つの異なるトピックに対して作成される場合、別のクライアント(最初にそのサブスクリプション名を作成したクライアントではない)が既存のサブスクリプション名を使用すると、サブスクリプションは削除されずにエラーが発生します。サブスクリプションがJMSまたはPL/SQLのどちらで作成されたかが不明であるため、その他のトピックについてのサブスクリプションは削除しないでください。

関連項目:

- [「MessageSelector」](#)
- [Java EEの準拠](#)

RemoteSubscriber

リモート・サブスクライバは、createRemoteSubscriberコールを使用して定義されます。リモート・サブスクライバは、リモート・トピックでの特定のコンシューマ、またはリモート・トピックでのすべてのサブスクライバになることができます。

リモート・サブスクライバは、AQjmsAgent構造を使用して定義されます。AQjmsAgentは、名前およびアドレスで構成されます。名前は、リモート・トピックのconsumer_nameを参照します。アドレスは、次のようにしてリモート・トピックを参照します。

```
schema.topic_name[@dblink]
```

リモート・トピックで特定のコンシューマに対してメッセージをパブリッシュするには、リモート・トピックでの[受信者の](#)subscription_nameが、AQjmsAgentの名前フィールドに指定されている必要があります。リモート・トピックは、AQjmsAgentのaddressフィールドに指定される必要があります。

リモート・トピックのすべてのサブスクライバに対してメッセージをパブリッシュするには、AQjmsAgentのnameフィールドをNULLに設定する必要があります。リモート・トピックは、AQjmsAgentのaddressフィールドに指定される必要があります。

TopicPublisher

メッセージはTopicPublisherを使用してパブリッシュされます。これは、TopicをcreatePublisherメソッドに渡すことによって作成されます。また、クライアントにはTopicを指定しないで、TopicPublisherを作成するオプションがあります。この場合、Topicをパブリッシュ操作のたびに指定する必要があります。クライアントは、TopicPublisherによって送信されるすべてのメッセージのデフォルト配信モード、優先順位およびTimeToLiveを指定できます。また、各メッセージごとに、これらのオプションを指定することもできます。

受信者リスト

JMSパブリッシュ・サブスクライブ・モデルでは、クライアントは、トピックのすべてのサブスクライバにメッセージを送信するのではなく、明示的な受信者リストを指定できます。これらの受信者は、トピックの既存のサブスクライバである場合もあれば、そうでない場合もあります。受信者リストは、このメッセージのトピックのサブスクリプション・リストをオーバーライドします。受信者リスト機能は、JMSに対するOracleの拡張です。

TopicReceiver

受信者名が受信者リストに明示的に指定されていても、その受信者がキューのサブスクライバではない場合、その受信者に送信されるメッセージは、TopicReceiverを作成することによって受信できます。サブスクライバ名が指定されていない場合、クライアントがメッセージを受信するには、リモート・サイトで永続サブスクライバを使用する必要があります。TopicReceiverは、JMSに対するOracleの拡張です。

TopicReceiverは、messageSelectorを使用して作成できます。これによって、クライアントは、受信者に配信されるメッセージをセレクトにマッチするメッセージに制限できます。

関連項目:

[「MessageSelector」](#)

TopicBrowser

TopicBrowserを使用すると、クライアントはメッセージを削除しないでトピック上でメッセージを参照できます。このブラウザ用メ

ソッドは、トピック・メッセージをスキャンするために使用される `java.util.Enumeration` を戻します。TopicBrowser を作成できるのは、永続サブスクライバのみです。nextElement に対する最初のコールが、トピックのスナップショットを取得します。

TopicBrowser は、メッセージをスキャン中にオプションでロックできます。これは、メッセージに対する SELECT...for UPDATE コマンドの場合と類似しています。これによって、他のコンシューマがスキャン中のメッセージを削除することはなくなります。

TopicBrowser は、messageSelector を使用して作成できます。これによって、クライアントは、コンシューマに配信されるメッセージをセレクトにマッチするメッセージに制限できるようになります。

TopicBrowser は、ページ機能をサポートしています。これによって、TopicBrowser を使用するクライアントは、トピックの現行のブラウズ操作中に参照されたすべてのメッセージを廃棄できます。ページとは、参照済のすべてのメッセージを破壊的に受信することと同じです (TopicSubscriber を使用して削除した場合と似ています)。

ページでは、メッセージが TopicBrowser の `java.lang.Enumeration` の nextElement () 操作へのコールを使用してクライアントに戻された場合、そのメッセージは参照済とみなされます。クライアントがまだ参照していないメッセージは、ページ中には廃棄されません。ページ操作は、同じ TopicBrowser に対して何度も実行できます。

TopicBrowser の作成に使用した JMS Session がコミットされると、ページは正常に実行されます。セッションに対する操作がロールバックされた場合、ページ操作も取り消されます。

関連項目:

- [TopicBrowser の作成: 標準 JMS メッセージ](#)
- [TopicBrowser の作成: 標準 JMS メッセージ、メッセージのロック](#)
- 「[MessageSelector](#)」
- [TopicBrowser を使用したメッセージのブラウズ](#)

JMS パブリッシュ・サブスクライバ操作の設定

JMS でパブリッシュ・サブスクライバ・モデルの通信を使用するステップは、次のとおりです。

1. メッセージを保持するために1つ以上のトピックを設定します。これらのトピックは、関心がある領域またはサブジェクトを表します。たとえば、トピックで請求済注文情報を表すことができます。
2. `AQjmsDestination` で start コールを使用して、トピックに対するエンキュー/デキューを有効にします。
3. 永続サブスクライバのセットを作成します。各サブスクライバは、受信を希望するメッセージを選択する `messageSelector` を指定できます。NULL の `messageSelector` は、そのトピックに対してパブリッシュされたすべてのメッセージの受信をサブスクライバが希望していることを示します。

サブスクライバはローカルでもリモートでもかまいません。ローカル・サブスクライバは、メッセージがパブリッシュされるトピックと同じトピックに対して定義された永続サブスクライバです。リモート・サブスクライバは、特定のキューのサブスクライバとして定義された別のトピックまたはそのトピックに対する受信者です。リモート・サブスクライバを使用するには、ソース・トピックと宛先トピック間の伝播を設定する必要があります。リモート・サブスクライバと伝播は、JMS に対する Oracle の拡張です。

関連項目:

[伝播の管理](#)

4. パブリッシャのSessionでcreatePublisher () メソッドを使用してTopicPublisherオブジェクトを作成します。メッセージは、publishコールを使用してパブリッシュされます。メッセージは、トピックのすべてのサブスクライバ、またはトピックに対する指定された受信者のサブセットに対してパブリッシュできます。
5. サブスクライバは、receiveメソッドを使用してトピックに関するメッセージを受信します。
6. サブスクライバは、メッセージ・リスナーを使用して、非同期にメッセージを受信することもできます。

関連項目:

[1つ以上のキューのリスニング](#)

JMSメッセージ・プロデューサの機能

- [メッセージの優先順位および順序付け](#)
- [メッセージ遅延の指定](#)
- [メッセージ期限切れの指定](#)
- [メッセージのグループ化](#)

メッセージの優先順位および順序付け

メッセージの順序付けは、メッセージがキューまたはトピックから受信される順序を決定します。キューまたはトピックのキュー表の作成時に順序づけの方式が指定されます。現在は、Oracle Database Advanced Queuingでは、メッセージの優先順位とエンキュー時間に基いた順序づけをサポートしており、これによって順序づけの方法が4種類あります。

- First-In, First-Out (FIFO)

エンキュー時刻が順序付け基準として選択されると、メッセージはエンキュー時刻の順序で受信されます。エンキュー時刻は、メッセージのパブリッシュ/送信時にOracle Database Advanced Queuingによってメッセージに割り当てられます。これはデフォルトの順序付けです。

- 優先順位による順序付け

優先順位による順序付けが選択されると、各メッセージに優先順位が割り当てられます。優先順位は、パブリッシュ/送信時にメッセージ・プロデューサによってメッセージ・プロパティとして指定できます。メッセージは、割り当てられた優先順位の順序で受信されます。

- FIFO優先順位

FIFO優先順位による順序付けが選択されると、トピックまたはキューは優先順位による順序付けの場合と同様に機能します。2つのメッセージに同じ優先順位が割り当てられた場合、両者はエンキュー時刻の順に受信されます。

- エンキュー時刻に続く優先順位による順序付け

同じエンキュー時刻のメッセージは、そのメッセージの優先順位に従って受信されます。2つのメッセージの順序付け基準が同じ場合、受信される順序は予想できません。ただし、1つのセッション中に特定の順序付け基準で生成されたメッセージは、送信された順序で受信されることがOracle Database Advanced Queuingにより保証されます。

永続メッセージに使用できる順序付けスキームはすべて、バッファ済メッセージにも使用できますが、使用できるのは各メッセージ・クラス内のみです。同じセッション内でエンキューまたはパブリッシュされた永続メッセージとバッファ済メッセージ間の順序付けは、現在はサポートされていません。

メッセージ遅延の指定

メッセージをキュー/トピックに対して送信/パブリッシュするときに、遅延を指定できます。遅延は、そのメッセージがメッセージ・コンシューマに対して使用可能になるまでの時間を表します。遅延指定されたメッセージは、遅延の期限が切れるまで待機状態になります。遅延指定は、メッセージ識別子による受信でオーバーライドされます。

遅延は、JMSメッセージ・プロパティに対するOracle Database Advanced Queuingの拡張機能です。Oracle Database Advanced Queuingのバックグラウンド・プロセスのキュー・モニターが起動される必要があります。

メッセージ期限切れの指定

メッセージのプロデューサは、メッセージの期限切れまたはTimeToLiveを指定できます。これによって、そのメッセージがメッセージ・コンシューマに対して使用可能な期間が定義されます。

TimeToLiveは、送信/パブリッシュ時に指定するか、MessageProducerの設定済TimeToLiveメソッドを使用して指定することが可能で、前者が後者よりも優先されます。TimeToLiveを実装するには、Oracle Database Advanced Queuingのバックグラウンド・プロセスのキュー・モニターが実行されている必要があります。

メッセージのグループ化

1つのキューまたはトピックに属するメッセージをグループ化してセットを形成し、一度に1人のユーザーのみによって消費されるように設定できます。そのためには、[トランザクション処理](#)によるメッセージのグループ化が可能なキュー表にキューまたはトピックを作成する必要があります。同じグループに属するメッセージはすべて同じトランザクション内に作成する必要があり、同じトランザクション内に作成されたメッセージはすべて同じグループに属します。

メッセージのグループ化は、JMS仕様に対するOracle Database Advanced Queuingの拡張機能です。

この機能を使用すると、複雑なメッセージを、リンクされた一連の単純なメッセージに分割できます。たとえば、請求書キュー宛ての請求書は、ヘッダーのメッセージ、詳細情報の複数のメッセージ、フッターのメッセージの順に分割できます。

小さいオブジェクトに分割できるイメージやビデオなどの複合ラージ・オブジェクトがメッセージ・ペイロードにある場合は、メッセージのグループ化が非常に有効です。

グループに含まれるメッセージの優先順位、遅延および期限切れの各プロパティは、単にグループの最初のメッセージ(ヘッダー)のプロパティによってのみ判断されます。グループの他のメッセージのプロパティは無視されます。

メッセージ・グループは、伝播中も保持されます。宛先トピックは、トランザクション処理のグループ化に対して使用可能にしておく必要があります。

関連項目:

トランザクション処理でグループ化可能なキューからメッセージをデキューするときにメッセージ・グループを保持する場合、注意する必要がある制限については、[「デキュー機能」](#)を参照してください。

JMSメッセージ・コンシューマ機能

この項の内容は次のとおりです。

- [メッセージの受信](#)
- [受信におけるメッセージのナビゲーション](#)

- [メッセージのブラウズ](#)
- [取出しを伴わないメッセージの削除](#)
- [遅延間隔をおいた後の再試行](#)
- [MessageListenerを使用したメッセージの非同期受信](#)
- [例外キュー](#)

メッセージの受信

JMSアプリケーションは、メッセージ・コンシューマを作成することによって、メッセージを受信できます。メッセージは、receiveコールを使用して同期的に受信するか、メッセージ・リスナーを使用して非同期的に受信できます。

受信モードには次の3つがあります。

- メッセージがコンシューマに届くまでブロック
- 指定最大時間までブロック
- 非ブロック

受信におけるメッセージのナビゲーション

コンシューマがナビゲーション・モードを指定しない場合、セッションの最初のreceiveはキューまたはトピックの最初のメッセージを取り出し、第2のreceiveは次のメッセージを取得します。優先順位が高いメッセージがコンシューマに届く場合、すでに届いているメッセージを削除するまで、コンシューマはこのメッセージを受信しません。

コンシューマが、メッセージに対してより効率的にキューのナビゲーションを制御できるように、Oracle Database Advanced QueuingにはJMS拡張機能として複数のナビゲーション・モードが用意されています。これらのモードは、TopicSubscriber、QueueReceiverまたはTopicReceiverで設定できます。

グループ化されていないメッセージには、次の2つのモードを使用できます。

- FIRST_MESSAGE
このモードでは、位置がキューの先頭にリセットされます。コンシューマがキューの最上位にあるメッセージを削除できるため、優先順位に基づくキューで役立ちます。
- NEXT_MESSAGE
このモードは、コンシューマの確立された位置の後のメッセージを取得します。たとえば、4番目のメッセージの位置で適用されたNEXT_MESSAGEは、そのキューの5番目のメッセージを取得します。これはデフォルト・アクションです。

グループ化メッセージには、次の3つのモードを使用できます。

- FIRST_MESSAGE
このモードでは、位置がキューの先頭にリセットされます。
- NEXT_MESSAGE
このモードでは、位置が同一トランザクションの次のメッセージに設定されます。
- NEXT_TRANSACTION
このモードでは、位置が次のトランザクションの最初のメッセージに設定されます。

 注意:

シャード・キューは、前述の3つのモードをサポートしていません。

次の方法でメッセージが受信される場合、グループ化トランザクションのプロパティを無効にできます。

- セレクタに相関識別子を指定して受信
- セレクタにメッセージ識別子を指定して受信
- トランザクション・グループのメッセージがすべて受信される前にコミット

NEXT_MESSAGEまたはNEXT_TRANSACTIONオプションの使用中にコンシューマがキューの最後に到達したとします。ブロッキング `receive()` を指定していた場合は、ナビゲート位置は自動的にそのキューの先頭に変更されます。

デフォルトでは、`QueueReceiver`、`TopicReceiver`または`TopicSubscriber`は、最初の`receive`コールにFIRST_MESSAGEを、以降の`receive()`コールにNEXT_MESSAGEを使用します。

メッセージのブラウズ

デキューするクライアントがキューからメッセージを削除できる通常の`receive`の他に、JMSでは、JMSクライアントがキューで自身のメッセージをブラウズできるようにするインタフェースを提供しています。`QueueBrowser`は、`QueueSession`から`createBrowser`メソッドを使用して作成できます。

メッセージが参照されると、そのメッセージは引き続き処理できます。メッセージがブラウズされた後は、同時セッションから`receive`コールがそのメッセージを削除する場合があるため、JMSセッションに引き続き使用できるとはかぎりません。

一度参照したメッセージが同時JMSクライアントによって削除されないようにするために、ロック・モードでメッセージを参照できます。そのためには、JMSインタフェースに対するOracle Database Advanced Queuingの拡張機能を使用して、ロック・モードを持つ`QueueBrowser`を作成する必要があります。メッセージのロックは、セッションがコミットまたはロールバックを実行すると解放されます。

`QueueBrowser`によって参照されたメッセージを削除するには、セッションが`QueueReceiver`を作成し、JMSmessageIDをセレクタとして使用する必要があります。

取出しを伴わないメッセージの削除

コンシューマは、`receiveNoData`コールを使用してキューまたはトピックからメッセージを取得することなく、メッセージを削除できます。アプリケーションが`QueueBrowser`などを使用してメッセージをすでに確認した場合に役立ちます。このモードを使用すると、JMSクライアントがペイロードをデータベースから取得するオーバーヘッドを回避できます(メッセージが大きい場合、オーバーヘッドが顕著である可能性があります)。

遅延間隔をおいた後の再試行

キュー/トピックからメッセージを受信するトランザクションが失敗した場合、そのメッセージを削除する試行に失敗したとみなされず。Oracle Database Advanced Queuingは、メッセージ削除の試行に失敗した回数をメッセージ履歴に記録します。

アプリケーションでは、メッセージに対する再試行の最大回数をキュー/トピック・レベルで指定できます。メッセージ削除がこの数より多く失敗した場合、メッセージは例外キューに移動されます。

Oracle Database Advanced Queuingでは、ユーザーは`max_retries`とともに`retry_delay`も指定できます。これは、受信の試行に失敗したメッセージを、`retry_delay`間隔後に引き続きキューで参照し、デキューできることを意味します。それまでこのメッセージはWAITING状態になります。Oracle Database Advanced Queuingのバックグラウンド・プロセスのタイム・マネージャは、再試行遅延プロパティを強制的に適用します。

再試行の最大回数および再試行の遅延は、キュー/トピックのプロパティです。このプロパティは、キュー/トピックの作成時、またはキュー/トピックに対する変更メソッドを使用して設定できます。MAX_RETRIESのデフォルト値は5です。

注意:



シャード・キューは、再試行の遅延をサポートしていません。

MessageListenerを使用したメッセージの非同期受信

JMSクライアントは、setMessageListenerメソッドを使用してMessageListenerを設定することによって、メッセージを非同期的に受信できます。

コンシューマにメッセージが届いた場合、メッセージ・リスナーのonMessageメソッドがそのメッセージで起動されます。メッセージ・リスナーは、メッセージの受信をコミットまたは異常終了できます。メッセージ・リスナーは、JMS Connectionが停止されている場合、メッセージを受信しません。一度メッセージ・リスナーがコンシューマに対して設定されると、メッセージの受信にreceiveコールを使用することはできません。

JMSクライアントは、セッションでMessageListenerを設定することによって、そのセッションのすべてのコンシューマに対してメッセージを非同期的に受信できます。一度メッセージ・リスナーが設定されると、そのセッションでは、その他のメッセージ受信モードを使用できません。

例外キュー

例外キューは、すべての期限切れのメッセージまたは使用できないメッセージが格納されるリポジトリです。アプリケーションがメッセージを例外キューに直接エンキューすることはできません。ただし、期限切れまたは処理できないメッセージを処理するアプリケーションは、例外キューからこれらのメッセージを受信または削除できます。

例外キューからメッセージを取り出すには、JMSクライアントはPoint-to-Pointインタフェースを使用する必要があります。トピック用のメッセージの例外キューは、使用可能な複数のコンシューマでキュー表に作成する必要があります。他のキューと同様に、例外キューもAQOracleQueueクラスでstartメソッドを使用して、メッセージを受信する必要があります。例外キューをエンキュー可能に設定しようとすると、例外が発生します。

シャード・キューは、DBMS_AQADM.CREATE_EXCEPTION_QUEUE APIを介して例外キューをサポートするようになりました。

```
PROCEDURE CREATE_EXCEPTION_QUEUE (  
    sharded_queue_name      IN VARCHAR2,  
    exception_queue_name   IN VARCHAR2 DEFAULT NULL,  
    multiple_consumers     IN BOOLEAN DEFAULT FALSE,  
    storage_clause         IN VARCHAR2 DEFAULT NULL,  
    sort_list              IN VARCHAR DEFAULT NULL,  
    comment                IN VARCHAR2 DEFAULT NULL  
);
```

例外キューは、“JMS_OracleExcpQ”と呼ばれるOracle固有のメッセージ・プロパティで、メッセージの送信/パブリッシュ前に設定できます。例外キューが指定されていないと、デフォルトの例外キューが使用されます。非シャード・キューの場合、デフォルトの例外キューはキュー表の作成時に自動的に作成され、AQ\$_queue_table_name_Eと命名されます。デフォルトでは、シャード・キューには例外キューは作成されません。

メッセージは、次の条件が成立するときに例外キューに移されます。

- そのメッセージが、指定されたtimeToLive内にデキューされなかった場合。

複数のサブスクライバを指定したメッセージの場合、指定されたtimeToLive内にそのメッセージをデキューできない受信者が1つ以上あると、メッセージは例外キューに移されます。

- メッセージが正常に受信されたが、メッセージ処理中のエラーのためアプリケーションがreceiveを実行したトランザクションを異常終了した場合。そのメッセージはキュー/トピックに戻され、メッセージ受信のために待機中のどのアプリケーションでも使用可能になります。

アプリケーションがトランザクション全体を異常終了するか、receive前のセーブポイントまでロールバックしたときは、receiveがロールバックまたはUNDOされたとみなされます。

これは失敗したメッセージ受信の試行であるため、その再試行回数は更新されます。メッセージの再試行回数が、メッセージが常駐するキュー/トピックに指定された最大値を超える場合、そのメッセージは例外キューに移されます。

メッセージが複数のサブスクライバを持つ場合、そのメッセージは、すべての受信者が再試行制限を超えたときにのみ、例外キューに移されます。

注意:



サーバー・プロセスがインスタンスで停止した(ALTER SYSTEM KILL SESSION または SHUTDOWN ABORT など)ためにデキュー・トランザクションが失敗した場合、RETRY_COUNT は増分されません。

JMS伝播

この項の内容は次のとおりです。

- [RemoteSubscriber](#)
- [伝播スケジュール](#)
- [拡張伝播スケジュール機能](#)
- [伝播中の例外処理](#)

注意:



シャード・キューは、RemoteSubscriber、伝播のスケジューリング、拡張伝播スケジュール機能および伝播中の例外処理をサポートしていません。

RemoteSubscriber

Oracle Database Advanced Queuingによって、他のデータベースにあるサブスクライバをトピックにサブスクライブできます。トピックに対してパブリッシュされたメッセージがリモート・サブスクライバの基準を満たしている場合は、リモート・サブスクライバに指定されているリモート・データベースにあるキュー/トピックに自動的に伝播します。伝播は、データベース・リンクおよびOracle Net Servicesを使用して実行されます。これによって、同じデータベースに接続しなくても、アプリケーション同士が互いに通信できます。

リモート・サブスクライバを実装するには、次の2つの方法があります。

- createRemoteSubscriberメソッドは、トピック上またはトピックに対してリモート・サブスクライバを作成するために使用します。このリモート・サブスクライバは、クラスAQjmsAgentのインスタンスとして指定されます。
- AQjmsAgentには、名前およびアドレスがあります。アドレスは、キュー/トピックおよびサブスクライバのデータベースへのデータベース・リンクで構成されています。

リモート・サブスクライバには、次の2種類があります。

- リモート・サブスクライバがトピックである場合。

これは、AQjmsAgentオブジェクトのリモート・サブスクライバに名前が指定されず、アドレスがトピックである場合に発生します。サブスクライバのサブスクリプションを満たすメッセージが、リモート・トピックに伝播されます。伝播されたメッセージは、それが満たすリモート・トピックのすべてのサブスクリプションに対して使用可能になります。

- メッセージに対して特定のリモート受信者を指定する場合。

リモート・サブスクリプションは、リモート・データベースにある特定のコンシューマに対して指定できます。リモート受信者の名前が(AQjmsAgentオブジェクトに)指定される場合、サブスクリプションを満たすメッセージが、その受信者専用のリモート・データベースに伝播されます。リモート・データベースにある受信者は、TopicReceiverインタフェースを使用してメッセージを取り出します。リモート・サブスクリプションは、Point-to-Pointキューに対して指定することもできます。

伝播スケジュール

伝播は、メッセージがターゲットの接続先データベースに伝播されるすべてのトピックについて、schedule_propagationメソッドを使用してスケジュールリングされる必要があります。

スケジュールは時間の枠を示し、メッセージはその枠内でソース・トピックから伝播されます。この時間枠は、ネットワーク通信量、ソース・データベースの負荷、接続先データベースの負荷などの複数の要因に左右されます。したがって、スケジュールは特定のソースおよび宛先にあわせて調整する必要があります。スケジュールが作成されると、ジョブは自動的にjob_queue機能に発行され、伝播が処理されます。

伝播スケジュールのための運用管理コールによって、スケジュール管理を柔軟に行うことができます。あるスケジュールの存続時間または伝播枠パラメータによって、伝播が開始される時間枠が指定されます。存続時間が指定されない場合、時間枠は無制限の単一枠になります。枠を定期的に繰り返す必要がある場合、連続する枠の間の周期的間隔を定義するnext_time機能を使用して有限の存続時間を指定します。

あるキューに定義された伝播スケジュールは、そのキューの有効期間中いつでも変更または削除できます。さらに、(スケジュールを削除するかわりに)一時的に使用不可にするコール、および使用不可のスケジュールを使用可能にするコールがあります。メッセージがスケジュール内で伝播されているとき、そのスケジュールはアクティブです。すべての管理コールは、スケジュールがアクティブかどうかに関係なく実行されます。スケジュールがアクティブの場合、コールが実行されるまでに数秒かかります。

伝播が開始されるには、ジョブ・キュー・プロセスを起動する必要があります。少なくとも2つのジョブ・キュー・プロセスを起動する必要があります。接続先データベースへのデータベース・リンクも有効にする必要があります。伝播のソースおよび宛先トピックは、同じメッセージ型である必要があります。リモート・トピックは、エンキューできる必要があります。データベース・リンクのユーザーもリモート・トピックに対するエンキュー権限を持つ必要があります。

関連項目:

[伝播のスケジュールリング](#)

拡張伝播スケジュール機能

伝播のために定義されたカタログ・ビューは、アクティブ・スケジュールに関する次の情報を提供します。

- そのスケジュールを処理しているバックグラウンド・プロセスの名前
- 伝播を処理しているセッションのSID(セッションおよびシリアル番号)
- スケジュールを処理しているインスタンス(Oracle RACを使用している場合)
- 先行して正常に実行されたスケジュール
- 次に実行予定のスケジュール

スケジュールごとに次の伝播統計が保持され、キュー管理者がスケジュール調整に役立てることができます。

- スケジュールの中で伝播されたメッセージ合計数
- スケジュールの中で伝播されたバイト合計数
- 伝播枠の中で伝播されたメッセージの最大数
- 伝播枠の中で伝播されたバイトの最大値
- 伝播枠の中で伝播されたメッセージの平均数
- 伝播済メッセージの平均サイズ
- 伝播済メッセージの平均時間

伝播機能には、障害対処およびエラー・レポートが組み込まれています。たとえば、指定されたデータベース・リンクが無効な場合、リモート・データベースが使用できない場合、またはリモート・トピック/キューにエンキューできない場合、適切なエラー・メッセージがレポートされます。伝播は指数バックオフ・スキームを使用して、障害が発生したスケジュールからの伝播を再試行します。あるスケジュールで続けて障害が発生したときは、最初の再試行は30秒後、次の再試行は60秒後、3回目の再試行は120秒後、というように続きます。再試行時間が現行の伝播枠の期限切れ時刻を超える場合は、次の再試行は、次の伝播枠の開始時刻に行われます。最大16回の再試行が行われた後、そのスケジュールは自動的に使用不可能になります。

注意:



再試行が次の伝播ウィンドウに移動されると、常に移動されるようになり、指数バックオフ・スキームは再試行のスケジュールを管理しません。DBMS_AQADM.SCHEDULE_PROPAGATION()のnext_timeパラメータで指定された日付関数の結果、ウィンドウ間の間隔が短くなると、再試行の失敗数はすぐに16に達し、スケジュールが無効になります。

障害のためにスケジュールが自動的に使用不可になると、関連情報がアラート・ログに書き込まれます。スケジュールで失敗が発生したか、その場合は何回連続して失敗が発生したか、失敗の原因を示すエラー・メッセージ、および最後の失敗が発生した時刻を、いつでも確認できます。この情報を調べることで、管理者は障害を回復し、スケジュールを使用可能にできます。

再試行の間に伝播が成功したときは、障害の数は0(ゼロ)にリセットされます。

伝播機能にはOracle Real Application Clustersサポートが組み込まれていますが、ユーザーおよび管理者には透過的です。伝播を処理するジョブは、ソース・トピックが常駐しているキュー表の所有者と同じインスタンスに送られます。あるインスタンスに障害が発生してトピックを保存しているキュー表が他のインスタンスに移される場合は、伝播ジョブも必ず自動的に新しいインスタンスに移行されます。これによって、インスタンス間のping操作は最小限に抑えられ、パフォーマンスが向上します。伝播は、同時スケジュールをいくつでも処理できるように設計されています。

job_queue_processesの最大数は1000で、その一部は伝播に関連しないジョブの処理に使用できます。このために、伝播にはマルチタスキングおよびロード・バランシングのサポートが組み込まれています。伝播アルゴリズムは、複数スケジュールが単一スナップショット(job_queue)のプロセスによって処理できるように設計されています。job_queueプロセスに対する伝播の負荷は、異なるソース・トピックからのメッセージ到着の割合に基づいて偏りが発生する場合があります。あるプロセスが数個のアクティブ・スケジュールによって過負荷になっている一方で、別のプロセスは受動的なスケジュールが多いために余力があるというとき、伝播はプロセス間で負荷が均等になるようにスケジュールを自動的に再分配します。

伝播中の例外処理

ネットワーク障害のようなシステム・エラーが発生した場合、Oracle Database Advanced Queuingは指数バックオフ・アルゴリズムを使用してメッセージを伝播する試みを続けます。キューからデータベース・リンクへの伝播中のアプリケーション・エラーを示している状況では、Oracle Database Advanced QueuingはそのメッセージにUNDELIVERABLEというマークを付けて alert.log に記録します。このようなエラーは、リモート・キューが存在しない場合、またはソース・キューおよびリモート・キューの型が一致しない場合に発生します。background_dump_destディレクトリのトレース・ファイルには、そのエラーに関する追加情報があります。

新規のジョブ・キュー・プロセスが開始すると、型を再検証できるように型の不一致エラーを消去します。ジョブ・キュー・プロセス数に上限を設定して、伝播のビジー状態が続く場合、ジョブ・キュー・プロセスが終了して再開するまで待つ必要はありません。キューの型は、必要に応じてDBMS_AQADM. VERIFY_QUEUE_TYPESを使用して再検証できます。

注意:



キューからキューへの伝播中に型の不一致が検出されると、伝播は停止してエラーが発生します。このような場合は、DBA_SCHEDULES ビューを問い合わせ、特定の宛先への伝播中に発生した最後のエラーを判断する必要があります。このメッセージには、UNDELIVERABLE マークは付いていません。

JMS AQのメッセージ変換

あるフォーマットのメッセージを別のフォーマットのメッセージにマップするために[変換](#)を定義できます。変換は、同一の情報を異なるフォーマットで表現するアプリケーションを統合する必要がある場合に有効です。変換はSQL式およびPL/SQLファンクションです。メッセージ変換は、標準JMSインタフェースに対するOracle Database Advanced Queuingの拡張機能です。

変換は、DBMS_TRANSFORM. create_transformationプロシージャを使用して作成できます。変換は、次の操作を行う場合に指定できます。

- キューまたはトピックへのメッセージの送信
- キューまたはトピックからのメッセージの受信。
- TopicSubscriberの作成
- RemoteSubscriberの作成。これによって、異なるフォーマットのトピック間でメッセージを伝播できます。

注意:



JMS シャード・キューは、メッセージ変換をサポートしていません。

JMSストリーミング

AQ JMSは、大量のメッセージ・データまたはペイロードを送受信するアプリケーション用に、AQ `jmsBytesMessage` および `jmsStreamMessage` を介した共有キューのエンキューおよびデキューでのストリーミングをサポートします。

JMSストリーミングは、大きい連続したバイトの配列を送受信するのではなく、メッセージ・ペイロードを小さいチャンクに分割することにより、大きいメッセージを扱うときのメモリー要件を緩和します。JMS標準にストリーミング・メカニズムは含まれないため、AQ JMSはAQストリーミング・エンキューおよびデキュー機能を示すための固有のインタフェースを提供します。これにより、ユーザーは既存のjava入出力ストリームを使用してメッセージ・データまたはペイロードを簡単に送受信できます。

データベースのRDBMS 12.2へのアップグレード時に既存のアプリケーションが変更なしで動作できるように、デフォルトではストリーミングAPIは無効化されます。

クライアント・アプリケーションはシステム・プロパティ `oracle.jms.useJmsStreaming` を `true` に設定して使用することにより、JMSストリーミングを有効化できます。

注意:



JMSストリーミングはThinドライバの場合のみサポートされます。

エンキューでのJMSストリーミング

AQ JMSではAQ `jmsBytesMessage` およびAQ `jmsStreamMessage` の新しいAPI `setInputStream(java.io.InputStream)` が提供され、メッセージ・データ用の入力ストリームを設定します。

```
/**
 * @param inputStream - InputStream to read the message payload
 * @throws JMSEException - if the JMS provided fails to read the payload due to
 *                          some internal error
 */
public void setInputStream(InputStream inputStream) throws JMSEException
```

次のコード・スニペットでは、AQ `jmsBytesMessage` タイプのメッセージが作成され、メッセージ・データ用の `FileInputStream` が設定されます。

```
Session session = connection.createSession(false, Session.CLIENT_ACKNOWLEDGE);
Destination destination = session.createQueue("queueName");
MessageProducer producer = session.createProducer(destination);

AQjmsBytesMessage bytesMessage = (AQjmsBytesMessage) session.createBytesMessage();
InputStream input = new FileInputStream("somefile.data");
bytesMessage.setInputStream(input);
producer.send(bytesMessage);
```

注意:



- `BytesMessage` および `StreamMessage` のメソッドは `java.io.DataInputStream` および `java.io.DataOutputStream` で検出されるメソッドに基づいているため、様々な `read*()` および

`write*()` メソッドの意味のある変換はストリームでは不可能です。次の使用例では例外が発生します。

- `bytesMessage.setInputStream(input);`
`bytesMessage.writeInt(99);`
- `bytesMessage.writeInt(99);`
`bytesMessage.setInputStream(input);`
- 通常のエンキュー操作と同様に、ストリーミングを使用したエンキューは同期操作となり、エンキューが完了した後でのみ、制御がクライアントに戻されます。
- これらの API が明示的にクライアントにより使用される場合にのみ、ストリーミングがエンキューで使用されず。AQ JMS はメッセージ・データのサイズと関係なく、通常のエンキューではストリーミングを使用しません。

デキューでのJMSストリーミング

ストリーミングを使用したデキュー操作は、2つのステップで実行されます。サーバーでは、メッセージ本文のサイズに基づき、メッセージ本文をストリーミングするかどうかが決まります。デフォルトのしきい値制限は10MBです。そのため、メッセージ本文が10MBよりも大きく、システム・プロパティ`oracle.jms.useJmsStreaming`を使用して、ストリーミングがクライアントで有効な場合、サーバーはデキューでストリーミングを使用します。

- これはクライアントが`receive()` メソッドを呼び出す場合の通常のデキュー・プロセスです。

```
Destination destination = session.createQueue("queueName");
AQJmsConsumer consumer = (AQJmsConsumer) session.createConsumer(destination);
Message message = consumer.receive(10000);
```

- クライアントがペイロードなしのメッセージを受信する場合、クライアントは受信メッセージでの`isLargeBody()`の呼び出しにより、ストリーミングがデキューで使用されるかどうかを特定します。

```
/**
 * This method can be used by the client applications to check whether the message
 * contains large message body and hence requires streaming with dequeue.
 *
 * @return true when the message body is large and server decides to stream
 *         the payload with dequeue
 */
public boolean isLargeBody()
```

`isLargeBody()` により戻された`true`の値は、デキューでのストリーミングを示します。デキューでストリーミングを使用する場合、AQ JMSは`AQJmsStreamMessage`と`AQJmsBytesMessage`に対して正確にメッセージ本文の長さを移入します。そのため、クライアント・アプリケーションはメッセージで`getBodyLength()`を呼び出し、ペイロードのサイズを判断できます。

```
public long getBodyLength()
```

クライアントがデキューでのストリーミングについて理解したら、受信メッセージで次のAPIのいずれかを使用してメッセージ・データをフェッチできます。

クライアント・アプリケーションは、`AQJmsBytesMessage`および`AQJmsStreamMessage`で使用可能な次のAPIを使用して、メッセージ・データを受信できます。

```

/**
 * Writes the message body to the OutputStream specified.
 *
 * @param outputStream - the OutputStream to which message body can be written
 * @return the OutputStream containing the message body.
 * @throws JMSEException - if the JMS provided fails to receive the message body
 *                          due to some internal error
 */
public OutputStream getBody(OutputStream outputStream) throws JMSEException

/**
 * Writes the message body to the OutputStream specified, with chunkSize bytes
 * written at a time.
 *
 * @param outputStream - the OutputStream to which message body can be written
 * @param chunkSize - the number of bytes to be written at a time, default value
 *                   8192 (ie. 8KB)
 * @return the OutputStream containing the message body.
 * @throws JMSEException - if the JMS provided fails to receive the message body
 *                          due to some internal error
 */
public OutputStream getBody(OutputStream outputStream, int chunkSize) throws JMSEException

/**
 * Writes the message body to the OutputStream specified. This method waits until
 * the message body is written completely to the OutputStream or the timeout expires.
 *
 * A timeout of zero never expires, and a timeout of negative value is ignored.
 *
 * @param outputStream - the OutputStream to which message body can be written
 * @param timeout - the timeout value (in milliseconds)
 * @return the OutputStream containing the message body.
 * @throws JMSEException - if the JMS provided fails to receive the message body
 *                          due to some internal error
 */
public OutputStream getBody(OutputStream outputStream, long timeout) throws JMSEException

/**
 * Writes the message body to the OutputStream specified, chunkSize bytes at a time.
 * This method waits until the message body is written completely to the OutputStream
 * or the timeout expires.
 *
 * A timeout of zero never expires, and a timeout of negative value is ignored.
 *
 * @param outputStream - the OutputStream to which message body can be written
 * @param chunkSize - the number of bytes to be written at a time,
 *                   default value 8192 (ie. 8KB)
 * @param timeout - the timeout value (in milliseconds)
 * @return the OutputStream containing the message body.
 * @throws JMSEException - if the JMS provided fails to receive the message body
 *                          due to some internal error
 */
public OutputStream getBody(OutputStream outputStream, int chunkSize, long timeout) throws
JMSEException

```

次のコード・スニペットは、ストリーミングがデキューで使用され、受信したペイロードがFileOutputStreamに書き込まれるかどうかをチェックします。

```

if (message instanceof BytesMessage && (AQjmsBytesMessage)message.isLargeBody()) {
    // optional : check the size of the payload and take appropriate action before
    // receiving the payload.
    (AQjmsBytesMessage) message.getBody(new FileOutputStream(new File("...")));
} else {
    // normal dequeue
}

```

通常、両方のステップが完了すると、メッセージは完全に消費されたとみなされます。AQサーバーにより、ステップ1の後でメッセージでのロックは保持され、ステップ2の後でのみ解除されます。

メッセージ・コンシューマによりメッセージが部分的に消費されることにより発生する可能性のある問題を考慮して、確認モード CLIENT_ACKNOWLEDGEおよびSESSION_TRANSACTEDでのセッション用のストリーミングAPIは制限されました。

そのため、部分的に消費されたメッセージを含む、すべてのメッセージは、次の時点で完全に消費されたとみなされます。

- message.acknowledge() はCLIENT_ACKNOWLEDGEセッションで呼び出されます。
- セッションのcommit() はトランザクション・セッションで呼び出されます。

通常の場合、セッションrollback() は、そのセッションで受信したメッセージをロールバックします。

JMSストリーミングは使用可能ですが、次の制限があります。

- ストリーミングはデフォルトで無効化されており、システム・プロパティoracle.jms.useJmsStreamingを使用してクライアント・アプリケーションにより有効化できる
- メッセージ・データのサイズがしきい値よりも大きい場合に、デキューでストリーミングが使用されます。デフォルトのしきい値は10MBです。
- ストリーミング・サポートはAQjmsBytesMessageおよびAQjmsStreamMessageで使用可能
- ストリーミング・サポートは共有キューでのみ使用可能
- ストリーミング・サポートはThinドライバでのみ使用可能
- ストリーミング・サポートは、メッセージ・プロデューサがメッセージ配信モードにNON_PERSISTENTを使用しているときは使用できない
- ストリーミングはメッセージ・リスナーではサポートされません。そのため、MessageConsumerにメッセージ・リスナー・セットがある場合、およびメッセージ・データがしきい値制限を超えている場合は、内部的に通常のデキューが使用されます。
- ストリーミング・サポートは、確認モードCLIENT_ACKNOWLEDGEおよびSESSION_TRANSACTEDを使用してセッションで使用可能です。

Java EEの準拠

Oracle JMSは、Oracle Sun Microsystems JMS 1.1標準に準拠しています。実行時に、[Oracle Java Message Service](#) (Oracle JMS)クライアントに対してJava EE準拠モードを定義できます。準拠モードにするには、コマンドライン・オプションとしてJavaプロパティ oracle.jms.j2eeCompliantをTRUEに設定します。非準拠モードにする場合は何もしません。FALSEがデフォルト値です。

Java EE準拠をサポートし、非準拠モードでも使用できるOracle Database Advanced Queuingの機能は次のとおりです。

- 非トランザクション・セッション

- 永続サブスクライバ
- 一時キューおよびトピック
- 非永続配信モード
- AQ\$_JMS_MESSAGE型のOracle Database Advanced Queuingキューを使用した単一のJMS[キュー](#)またはトピックに対する複数のJMSメッセージ・タイプ
- 永続サブスクライバに対するnoLocalオプション
- JMSシャード・キューはJMSでネイティブにサポートされており、Java EEコンプライアンスに準拠しています

関連項目:

- 『*Java Message Service Specification*』、バージョン1.1、2002年3月18日、Sun Microsystems, Inc.
- JMSPriorityおよびJMSExpirationに対するJavaプロパティoracle.jms.j2eeCompliantの影響の詳細は、[「JMSメッセージ・ヘッダー」](#)を参照してください。
- 永続サブスクライバに対するJavaプロパティoracle.jms.j2eeCompliantの影響の詳細は、[「DurableSubscriber」](#)を参照してください。

12 Oracle Java Message Serviceの基本操作

次のトピックでは、Oracle Database Advanced Queuing (AQ)の基本操作のための[Java Message Service \(JMS\)管理インタフェース](#)について説明します。

- [DBMS_AQINに対するEXECUTE権限](#)
- [ConnectionFactoryの登録](#)
- [キュー/トピックConnectionFactoryの登録解除](#)
- [QueueConnectionFactoryまたはTopicConnectionFactoryの取得](#)
- [LDAP内のキューまたはトピックの取得](#)
- [非シャード・キュー表の作成](#)
- [キューの作成](#)
- [非シャード・キュー表の取得](#)
- [権限の付与および取消し](#)
- [宛先の管理](#)
- [伝播スケジュール](#)

DBMS_AQINに対するEXECUTE権限

ユーザーがDBMS_AQINパッケージ内のメソッドを直接コールすることはありませんが、DBMS_AQINに対するEXECUTE権限が必要です。次の構文を使用して実行します。

```
GRANT EXECUTE ON DBMS_AQIN to user;
```

ConnectionFactoryの登録

[ConnectionFactory](#)は、次の4つの方法で登録できます。

- [データベースを介した登録: JDBCコネクション・パラメータの使用](#)
- [データベースを介した登録: JDBC URLの使用](#)
- [LDAPを介した登録: JDBCコネクション・パラメータの使用](#)
- [LDAPを介した登録: JDBC URLの使用](#)

データベースを介した登録: JDBCコネクション・パラメータの使用

```
public static int registerConnectionFactory(java.sql.Connection connection,
                                           java.lang.String conn_name,
                                           java.lang.String hostname,
                                           java.lang.String oracle_sid,
                                           int portno,
                                           java.lang.String driver,
                                           java.lang.String type)
                                           throws JMSEException
```

このメソッドは、JDBCコネクション・パラメータを使用し、データベースを介してQueueConnectionFactoryまたは

TopicConnectionFactoryを[Lightweight Directory Access Protocol](#)(LDAP)サーバーに登録します。このメソッドは静的であり、次のパラメータを取ります。

パラメータ	説明
connection	登録に使用される JDBC コネクション
conn_name	登録されるコネクションの名前
hostname	Oracle Database Advanced Queuing を実行しているホストの名前
oracle_sid	Oracle システム識別子
portno	ポート番号
driver	JDBC ドライバの型
type	コネクション・ファクトリのタイプ(Queue または Topic)

registerConnectionFactoryに渡されるデータベース接続には、AQ_ADMINISTRATOR_ROLEを付与する必要があります。登録後は、[Java Naming and Directory Interface](#)(JNDI)を使用してコネクション・ファクトリを検索できます。

例12-1 データベースを介した登録: JDBCコネクション・パラメータの使用

```
String url;
java.sql.connection db_conn;

url = "jdbc:oracle:thin:@sun-123:1521:db1";
db_conn = DriverManager.getConnection(url, "scott", "tiger");
AQjmsFactory.registerConnectionFactory(
    db_conn, "queue_conn1", "sun-123", "db1", 1521, "thin", "queue");
```

データベースを介した登録: JDBC URLの使用

```
public static int registerConnectionFactory(java.sql.Connection connection,
                                           java.lang.String conn_name,
                                           java.lang.String jdbc_url,
                                           java.util.Properties info,
                                           java.lang.String type)
                                           throws JMSEException
```

このメソッドは、JDBC URLを使用し、データベースを介してQueueConnectionFactoryまたはTopicConnectionFactoryをLDAPに登録します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
connection	登録に使用される JDBC コネクション
conn_name	登録されるコネクションの名前

パラメータ	説明
jdbc_url	接続先の URL
info	プロパティの情報
portno	ポート番号
type	コネクション・ファクトリのタイプ(Queue または Topic)

registerConnectionFactoryに渡されるデータベース接続には、AQ_ADMINISTRATOR_ROLEを付与する必要があります。登録後、JNDIを使用してコネクション・ファクトリを検索できます。

例12-2 データベースを介した登録: JDBC URLの使用

```
String url;
java.sql.connection db_conn;

url = "jdbc:oracle:thin:@sun-123:1521:db1";
db_conn = DriverManager.getConnection(url, "scott", "tiger");
AQjmsFactory.registerConnectionFactory(
    db_conn, "topic_conn1", url, null, "topic");
```

LDAPを介した登録: JDBCコネクション・パラメータの使用

```
public static int registerConnectionFactory(java.util.Hashtable env,
                                           java.lang.String conn_name,
                                           java.lang.String hostname,
                                           java.lang.String oracle_sid,
                                           int portno,
                                           java.lang.String driver,
                                           java.lang.String type)
    throws JMSEException
```

このメソッドは、JDBCコネクション・パラメータを使用し、LDAPを介してQueueConnectionFactoryまたはTopicConnectionFactoryをLDAPに登録します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
env	LDAP コネクションの環境
conn_name	登録されるコネクションの名前
hostname	Oracle Database Advanced Queuing を実行しているホストの名前
oracle_sid	Oracle システム識別子
portno	ポート番号

パラメータ	説明
driver	JDBC ドライバの型
type	コネクション・ファクトリのタイプ(Queue または Topic)

registerConnectionFactory() に渡されるハッシュ表に、LDAPサーバーと使用可能なコネクションを確立するための情報が含まれている必要があります。さらに、このコネクションには、LDAPサーバー内のコネクション・ファクトリのエントリに対する書き込み権限が必要です(LDAPユーザーがデータベースそのものであるか、またはLDAPユーザーにGLOBAL_AQ_USER_ROLEが付与されている必要があります)。登録後、JNDIを使用してコネクション・ファクトリを検索します。

例12-3 LDAPを介した登録: JDBCコネクション・パラメータの使用

```

Hashtable env = new Hashtable(5, 0.75f);
/* the following statements set in hashtable env:
 * service provider package
 * the URL of the ldap server
 * the distinguished name of the database server
 * the authentication method (simple)
 * the LDAP username
 * the LDAP user password
 */
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldap://sun-456:389");
env.put("searchbase", "cn=db1, cn=oraclecontext, cn=acme, cn=com");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn=db1aqadmin, cn=acme, cn=com");
env.put(Context.SECURITY_CREDENTIALS, "welcome");

AQjmsFactory.registerConnectionFactory(env,
                                     "queue_conn1",
                                     "sun-123",
                                     "db1",
                                     1521,
                                     "thin",
                                     "queue");

```

LDAPを介した登録: JDBC URLの使用

```

public static int registerConnectionFactory(java.util.Hashtable env,
                                           java.lang.String conn_name,
                                           java.lang.String jdbc_url,
                                           java.util.Properties info,
                                           java.lang.String type)
    throws JMSEException

```

このメソッドは、JDBCコネクション・パラメータを使用し、LDAPを介してQueueConnectionFactoryまたはTopicConnectionFactoryをLDAPに登録します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
env	LDAP コネクションの環境

パラメータ	説明
conn_name	登録されるコネクションの名前
jdbc_url	接続先の URL
info	プロパティの情報
type	コネクション・ファクトリのタイプ(Queue または Topic)

registerConnectionFactory() に渡されるハッシュ表に、LDAPサーバーと使用可能なコネクションを確立するための情報が含まれている必要があります。さらに、このコネクションには、LDAPサーバー内のコネクション・ファクトリのエントリに対する書き込み権限が必要です(LDAPユーザーがデータベースそのものであるか、またはLDAPユーザーにGLOBAL_AQ_USER_ROLEが付与されている必要があります)。登録後、JNDIを使用してコネクション・ファクトリを検索します。

例12-4 LDAPを介した登録: JDBC URLの使用

```
String url;
Hashtable env = new Hashtable(5, 0.75f);

/* the following statements set in hashtable env:
 * service provider package
 * the URL of the ldap server
 * the distinguished name of the database server
 * the authentication method (simple)
 * the LDAP username
 * the LDAP user password
 */
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldap://sun-456:389");
env.put("searchbase", "cn=db1,cn=OracleContext,cn=acme,cn=com");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn=db1admin,cn=acme,cn=com");
env.put(Context.SECURITY_CREDENTIALS, "welcome");
url = "jdbc:oracle:thin:@sun-123:1521:db1";
AQJMSFactory.registerConnectionFactory(env, "topic_conn1", url, null, "topic");
```

キュー/トピックConnectionFactoryの登録解除

LDAPのキュー/トピックConnectionFactoryを登録解除するには、2つの方法があります。

- [データベースを介した登録解除](#)
- [LDAPを介した登録解除](#)

データベースを介した登録解除

```
public static int unregisterConnectionFactory(java.sql.Connection connection,
                                             java.lang.String conn_name)
                                             throws JMSEException
```

このメソッドは、LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの登録を解除します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
connection	登録に使用される JDBC コネクション
conn_name	登録されるコネクションの名前

unregisterConnectionFactory () に渡されるデータベース接続には、AQ_ADMINISTRATOR_ROLEを付与する必要があります。

例12-5 データベースを介した登録解除

```
String url;
java.sql.connection db_conn;

url = "jdbc:oracle:thin:@sun-123:1521:db1";
db_conn = DriverManager.getConnection(url, "scott", "tiger");
AQjmsFactory.unregisterConnectionFactory(db_conn, "topic_conn1");
```

LDAPを介した登録解除

```
public static int unregisterConnectionFactory(java.util.Hashtable env,
                                             java.lang.String conn_name)
                                             throws JMSEException
```

このメソッドは、LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの登録を解除します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
env	LDAP コネクションの環境
conn_name	登録されるコネクションの名前

unregisterConnectionFactory () に渡されるハッシュ表に、LDAPサーバーと使用可能なコネクションを確立するための情報が含まれている必要があります。さらに、このコネクションには、LDAPサーバー内のコネクション・ファクトリのエントリに対する書き込み権限が必要です(LDAPユーザーがデータベースそのものであるか、またはLDAPユーザーにGLOBAL_AQ_USER_ROLEが付与されている必要があります)。

例12-6 LDAPを介した登録解除

```
Hashtable env = new Hashtable(5, 0.75f);

/* the following statements set in hashtable env:
 * service provider package
 * the distinguished name of the database server
 * the authentication method (simple)
 * the LDAP username
 * the LDAP user password
 */
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldap://sun-456:389");
env.put("searchbase", "cn=db1,cn=Oraclecontext,cn=acme,cn=com");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn=db1aqadmin,cn=acme,cn=com");
env.put(Context.SECURITY_CREDENTIALS, "welcome");
```

```
url = "jdbc:oracle:thin:@sun-123:1521:db1";
AQjmsFactory.unregisterConnectionFactory(env, "queue_conn1");
```

QueueConnectionFactoryまたはTopicConnectionFactoryの取得

この項の内容は次のとおりです。

- [QueueConnectionFactoryの取得: JDBC URLの使用](#)
- [QueueConnectionFactoryの取得: JDBCコネクション・パラメータの使用](#)
- [TopicConnectionFactoryの取得: JDBC URLの使用](#)
- [TopicConnectionFactoryの取得: JDBCコネクション・パラメータの使用](#)
- [LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの取得](#)

QueueConnectionFactoryの取得: JDBC URLの使用

```
public static javax.jms.QueueConnectionFactory getQueueConnectionFactory(
    java.lang.String jdbc_url,
    java.util.Properties info)
    throws JMSEException
```

このメソッドは、JDBC URLを使用してQueueConnectionFactoryを取得します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
jdbc_url	接続先の URL
info	プロパティの情報

例12-7 QueueConnectionFactoryの取得: JDBC URLの使用

```
String url = "jdbc:oracle:oci10:internal/oracle";
Properties info = new Properties();
QueueConnectionFactory qc_fact;

info.put("internal_logon", "sysdba");
qc_fact = AQjmsFactory.getQueueConnectionFactory(url, info);
```

QueueConnectionFactoryの取得: JDBCコネクション・パラメータの使用

```
public static javax.jms.QueueConnectionFactory getQueueConnectionFactory(
    java.lang.String hostname,
    java.lang.String oracle_sid,
    int portno,
    java.lang.String driver)
    throws JMSEException
```

このメソッドは、JDBC接続パラメータを使用してQueueConnectionFactoryを取得します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
-------	----

パラメータ	説明
hostname	Oracle Database Advanced Queuing を実行しているホストの名前
oracle_sid	Oracle システム識別子
portno	ポート番号
driver	JDBC ドライバの型

例12-8 QueueConnectionFactoryの取得: JDBCコネクション・パラメータの使用

```
String host = "dlsun";
String ora_sid = "rdbms10i";
String driver = "thin";
int port = 5521;
QueueConnectionFactory qc_fact;

qc_fact = AQjmsFactory.getQueueConnectionFactory(host, ora_sid, port, driver);
```

TopicConnectionFactoryの取得: JDBC URLの使用

```
public static javax.jms.QueueConnectionFactory getQueueConnectionFactory(
    java.lang.String jdbc_url,
    java.util.Properties info)
    throws JMSEException
```

このメソッドは、JDBC URLを使用してTopicConnectionFactoryを取得します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
jdbc_url	接続先の URL
info	プロパティの情報

例12-9 TopicConnectionFactoryの取得: JDBC URLの使用

```
String url = "jdbc:oracle:oci10:internal/oracle";
Properties info = new Properties();
TopicConnectionFactory tc_fact;

info.put("internal_logon", "sysdba");
tc_fact = AQjmsFactory.getTopicConnectionFactory(url, info);
```

TopicConnectionFactoryの取得: JDBCコネクション・パラメータの使用

```
public static javax.jms.TopicConnectionFactory getTopicConnectionFactory(
    java.lang.String hostname,
    java.lang.String oracle_sid,
    int portno,
    java.lang.String driver)
    throws JMSEException
```

このメソッドは、JDBC接続パラメータを使用してTopicConnectionFactoryを取得します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
hostname	Oracle Database Advanced Queuing を実行しているホストの名前
oracle_sid	Oracle システム識別子
portno	ポート番号
driver	JDBC ドライバの型

例12-10 TopicConnectionFactoryの取得: JDBCコネクション・パラメータの使用

```
String host = "dlsun";
String ora_sid = "rdbms10i";
String driver = "thin";
int port = 5521;
TopicConnectionFactory tc_fact;

tc_fact = AQjmsFactory.getTopicConnectionFactory(host, ora_sid, port, driver);
```

LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの取得

このメソッドは、LDAPからQueueConnectionFactoryまたはTopicConnectionFactoryを取得します。

例12-11 LDAP内のQueueConnectionFactoryまたはTopicConnectionFactoryの取得

```
Hashtable env = new Hashtable(5, 0.75f);
DirContext ctx;
queueConnectionFactory qc_fact;

/* the following statements set in hashtable env:
 * service provider package
 * the URL of the ldap server
 * the distinguished name of the database server
 * the authentication method (simple)
 * the LDAP username
 * the LDAP user password
 */
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldap://sun-456:389");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn=db1aquser1,cn=acme,cn=com");
env.put(Context.SECURITY_CREDENTIALS, "welcome");

ctx = new InitialDirContext(env);
ctx = (DirContext)ctx.lookup("cn=OracleDBConnections,cn=db1,cn=Oraclecontext,cn=acme,cn=com");
qc_fact = (queueConnectionFactory)ctx.lookup("cn=queue_conn1");
```

LDAP内のキューまたはトピックの取得

このメソッドは、LDAPからキューまたはトピックを取得します。

例12-12 LDAP内のキューまたはトピックの取得

```
Hashtable          env = new Hashtable(5, 0.75f);
DirContext         ctx;
topic              topic_1;

/* the following statements set in hashtable env:
 * service provider package
 * the URL of the ldap server
 * the distinguished name of the database server
 * the authentication method (simple)
 * the LDAP username
 * the LDAP user password
 */
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldap://sun-456:389");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn=db1auser1,cn=acme,cn=com");
env.put(Context.SECURITY_CREDENTIALS, "welcome");

ctx = new InitialDirContext(env);
ctx = (DirContext)ctx.lookup("cn=OracleDBQueues,cn=db1,cn=Oraclecontext,cn=acme,cn=com");
topic_1 = (topic)ctx.lookup("cn=topic_1");
```

非シャード・キュー表の作成

```
public oracle.AQ.AQQueueTable createQueueTable(
    java.lang.String owner,
    java.lang.String name,
    oracle.AQ.AQQueueTableProperty property)
    throws JMSEException
```

このメソッドは[キュー表](#)を作成します。次のパラメータがあります。

パラメータ	説明
owner	キュー表の所有者(スキーマ)
name	キュー表名
property	キュー表のプロパティ

キューを保持するためにキュー表を使用する場合、キュー表はマルチコンシューマに対して使用可能にすることはできません(デフォルト)。トピックを保持するためにキュー表を使用する場合、キュー表はマルチコンシューマに対して使用可能にする必要があります。

[CLOB](#)、[BLOB](#)および[BFILE](#)オブジェクトはOracle Database Advanced Queuing [オブジェクト型](#)のロードで有効な属性です。ただし、CLOBおよびBLOBのみ、Oracle 8i以降のOracle Database Advanced Queuing [伝播](#)を使用して伝播できます。



注意:

現在、JMS シャード・キューは、DBMS_AQADM PL/SQL API を介してのみ作成および削除できます。

例12-13 キュー表の作成

```
QueueSession      q_sess   = null;
AQQueueTable      q_table  = null;
AQQueueTableProperty qt_prop = null;

qt_prop = new AQQueueTableProperty("SYS.AQ$_JMS_BYTES_MESSAGE");
q_table = ((AQJmsSession)q_sess).createQueueTable(
    "boluser", "bol_ship_queue_table", qt_prop);
```

キューの作成

この項の内容は次のとおりです。

- [Point-to-Pointキューの作成](#)
- [パブリッシュ・サブスクライブ・トピックの作成](#)
- [Point-to-Pointキューおよびパブリッシュ・サブスクライブ・トピックのためのシャード・キューの作成](#)

Point-to-Pointキューの作成

```
public javax.jms.Queue createQueue(
    oracle.aq.aqqueueTable q_table,
    java.lang.String queue_name,
    oracle.jms.AQJmsDestinationProperty dest_property)
    throws JMSEException
```

このメソッドは、指定したキュー表にキューを作成します。次のパラメータがあります。

パラメータ	説明
q_table	キューが作成されるキュー表。キュー表はシングル・コンシューマ・キュー表である必要があります。
queue_name	作成されるキューの名前。
dest_property	キューのプロパティ。

このメソッドは、Oracle JMSに固有です。このメソッドでは標準のJava `javax.jms.Session` オブジェクトは使用できません。かわりに、標準の型をOracle JMSの具体的なクラス `oracle.jms.AQJmsSession` にキャストする必要があります。

例12-14 Point-to-Pointキューの作成

```
QueueSession      q_sess;
AQQueueTable      q_table;
AQJmsDestinationProperty dest_prop;
Queue              queue;

queue = ((AQJmsSession)q_sess).createQueue(q_table, "jms_q1", dest_prop);
```

パブリッシュ・サブスクライブ・トピックの作成

```
public javax.jms.Topic createTopic(
```

```

oracle.AQ.AQQueueTable q_table,
java.lang.String topic_name,
oracle.jms.AQjmsDestinationProperty dest_property)
throws JMSEException

```

このメソッドは、[パブリッシュ・サブスクライブ](#)・モデルにトピックを作成します。次のパラメータがあります。

パラメータ	説明
q_table	キューが作成されるキュー表。キュー表はマルチ・コンシューマ・キュー表である必要があります。
queue_name	作成されるキューの名前。
dest_property	キューのプロパティ。

このメソッドは、Oracle JMSに固有です。このメソッドでは標準のJava `javax.jms.Session` オブジェクトは使用できません。かわりに、標準の型をOracle JMSの具体的なクラス `oracle.jms.AQjmsSession` にキャストする必要があります。

[例12-16](#)では、在庫不足のために受注を満たせない場合は、注文を処理しているトランザクションが終了されます。`bookedorders` トピックは、`max_retries = 4`、`retry_delay = 12時間`で設定されています。したがって、注文が2日で満たされない場合は、例外キューに移動されます。

例12-15 パブリッシュ・サブスクライブ・トピックの作成

```

TopicSession      t_sess;
AQQueueTable      q_table;
AQjmsDestinationProperty dest_prop;
Topic              topic;

topic = ((AQjmsSession)t_sess).createTopic(q_table, "jms_t1", dest_prop);

```

例12-16 メッセージの最大試行回数と最大遅延の指定

```

public BolOrder process_booked_order (TopicSession jms_session)
{
    Topic          topic;
    TopicSubscriber tsubs;
    ObjectMessage  obj_message;
    BolCustomer    customer;
    BolOrder       booked_order = null;
    String         country;
    int            i = 0;

    try
    {
        /* get a handle to the OE_bookedorders_topic */
        topic = ((AQjmsSession) jms_session).getTopic("WS",
                                                    "WS_bookedorders_topic");

        /* Create local subscriber - to track messages for Western Region */
        tsubs = jms_session.createDurableSubscriber(topic, "SUBS1",
                                                    "Region = 'Western' ",
                                                    false);

        /* wait for a message to show up in the topic */
        obj_message = (ObjectMessage) tsubs.receive(10);
    }
}

```

```

booked_order = (BolOrder)obj_message.getObject();

customer = booked_order.getCustomer();
country   = customer.getCountry();

if (country == "US")
{
    jms_session.commit();
}
else
{
    jms_session.rollback();
    booked_order = null;
}
} catch (JMSEException ex)
{ System.out.println("Exception " + ex); }

return booked_order;
}

```

Point-to-Pointキューおよびパブリッシュ・サブスクライブ・トピックのためのシャード・キューの作成

AQ JMSでは、シャード・キューを作成および削除するための新しいAPIが定義されました。JMSにはキュー変更APIはありません。署名は次のとおりです。

```

/**
 * Create a JMS sharded queue. It also internally creates the related queue
 * objects (table, indexes) based on this name.
 *
 * @param queueName name of the queue to be created, format is schema.queueName
 *           (where the schema is optional
 * @param isMultipleConsumer flag to indicate whether the queue is a
 *           multi-consumer or single-consumer queue
 * @return javax.jms.Destination
 * @throws JMSEException if the queue could not be created
 */
public synchronized javax.jms.Destination createJMSShardedQueue(String queueName,
        boolean isMultipleConsumer) throws JMSEException {
    return createJMSShardedQueue(queueName, isMultipleConsumer, null, 0, null);
}

/**
 * Create a JMS sharded queue. It also internally creates the related queue
 * objects (table, indexes) based on this name.
 *
 * @param queueName name of the queue to be created, format is schema.queueName
 *           (where the schema is optional
 * @param isMultipleConsumer flag to indicate whether the queue is a
 *           multi-consumer or single-consumer queue
 * @param storageClause additional storage clause
 * @param maxRetries retry count before skip the message while dequeue
 * @param comment comment for the queue
 * @return javax.jms.Destination
 * @throws JMSEException if the queue could not be created
 */
public Destination createJMSShardedQueue(java.lang.String queueName,
        boolean isMultipleConsumer,

```

```
java.lang.String storageClause,  
int maxRetries,  
java.lang.String comment)  
throws JMSEException
```

非シャード・キュー表の取得

```
public oracle.AQ.AQQueueTable getQueueTable(java.lang.String owner,  
java.lang.String name)  
throws JMSEException
```

このメソッドは、非シャード・キューのキュー表を取得します。次のパラメータがあります。

パラメータ	説明
owner	キュー表の所有者(スキーマ)
name	キュー表名

コネクションをオープンしたコール側がキュー表の所有者ではない場合、コール側にはキュー表内のキュー/トピックに対する Oracle Database Advanced Queuing [エンキュー/デキュー](#) 権限が必要です。この権限がない場合、キュー表は取得できません。

例12-17 キュー表の取得

```
QueueSession      q_sess;  
AQQueueTable      q_table;  
  
q_table = ((AQjmsSession)q_sess).getQueueTable(  
    "boluser", "bol_ship_queue_table");
```

権限の付与および取消し

この項の内容は次のとおりです。

- [Oracle Database Advanced Queuingシステム権限の付与](#)
- [Oracle Database Advanced Queuingシステム権限の取消し](#)
- [パブリッシュ・サブスクライブ・トピック権限の付与](#)
- [パブリッシュ・サブスクライブ・トピック権限の取消し](#)
- [Point-to-Pointキュー権限の付与](#)
- [Point-to-Pointキュー権限の取消し](#)

Oracle Database Advanced Queuingシステム権限の付与

```
public void grantSystemPrivilege(java.lang.String privilege,  
java.lang.String grantee,  
boolean admin_option)  
throws JMSEException
```

このメソッドは、ユーザーおよびロールにOracle Database Advanced Queuingシステム権限を付与します。

パラメータ	説明
privilege	ENQUEUE_ANY、DEQUEUE_ANY または MANAGE_ANY
grantee	権限受領者(ユーザー、ロールまたは PUBLIC)
admin_option	TRUE に設定すると、権限受領者はこのプロシージャを使用して他のユーザーまたはロールにシステム権限を付与できます。

最初は、SYSおよびSYSTEMのみがこのプロシージャを正常に使用できます。ENQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューにメッセージをエンキューできます。DEQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューからメッセージをデキューできます。MANAGE_ANY権限を付与されたユーザーは、データベースのすべてのスキーマに対してDBMS_AQADMコールを実行できます。

例12-18 Oracle Database Advanced Queuingシステム権限の付与

```
TopicSession    t_sess;

((AQjmsSession)t_sess).grantSystemPrivilege("ENQUEUE_ANY", "scott", false);
```

Oracle Database Advanced Queuingシステム権限の取消し

```
public void revokeSystemPrivilege(java.lang.String privilege,
                                   java.lang.String grantee)
                                   throws JMSEException
```

このメソッドは、ユーザーまたはロールからOracle Database Advanced Queuingシステム権限を取り消します。次のパラメータがあります。

パラメータ	説明
privilege	ENQUEUE_ANY、DEQUEUE_ANY または MANAGE_ANY
grantee	権限受領者(ユーザー、ロールまたは PUBLIC)

ENQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューにメッセージをエンキューできます。DEQUEUE_ANY権限を付与されたユーザーは、データベース内の任意のキューからメッセージをデキューできます。MANAGE_ANY権限を付与されたユーザーは、データベースのすべてのスキーマに対してDBMS_AQADMコールを実行できます。

例12-19 Oracle Database Advanced Queuingシステム権限の取消し

```
TopicSession    t_sess;

((AQjmsSession)t_sess).revokeSystemPrivilege("ENQUEUE_ANY", "scott");
```

パブリッシュ・サブスクライブ・トピック権限の付与

```
public void grantTopicPrivilege(javax.jms.Session session,
                                 java.lang.String privilege,
                                 java.lang.String grantee,
                                 boolean grant_option)
                                 throws JMSEException
```

このメソッドは、パブリッシュ・サブスクライブ・モデルでトピック権限を付与します。初期設定では、キュー表の所有者のみがこのプロセスを使用してそのトピックの権限を付与できます。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
privilege	ENQUEUE、DEQUEUE または ALL (ALL は両方を意味します。)
grantee	権限受領者(ユーザー、ロールまたは PUBLIC)
grant_option	TRUE に設定すると、権限受領者はこのプロセスを使用して他のユーザーまたはロールにシステム権限を付与できます。

例12-20 パブリッシュ・サブスクライブ・トピック権限の付与

```
TopicSession    t_sess;
Topic            topic;

((AQjmsDestination) topic).grantTopicPrivilege(
    t_sess, "ENQUEUE", "scott", false);
```

パブリッシュ・サブスクライブ・トピック権限の取消し

```
public void revokeTopicPrivilege(javax.jms.Session session,
                                 java.lang.String privilege,
                                 java.lang.String grantee)
    throws JMSEException
```

このメソッドは、パブリッシュ・サブスクライブ・モデルでトピック権限を取り消します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
privilege	ENQUEUE、DEQUEUE または ALL (ALL は両方を意味します。)
grantee	権限が取り消される権限受領者(ユーザー、ロールまたは PUBLIC)

例12-21 パブリッシュ・サブスクライブ・トピック権限の取消し

```
TopicSession    t_sess;
Topic            topic;

((AQjmsDestination) topic).revokeTopicPrivilege(t_sess, "ENQUEUE", "scott");
```

Point-to-Pointキュー権限の付与

```
public void grantQueuePrivilege(javax.jms.Session session,
                                 java.lang.String privilege,
                                 java.lang.String grantee,
```

```
boolean grant_option)
throws JMSEException
```

このメソッドは、Point-to-Pointモデルでキュー権限を付与します。初期設定では、キュー表の所有者のみがこのプロシージャを使用してそのキューの権限を付与できます。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
privilege	ENQUEUE、DEQUEUE または ALL (ALL は両方を意味します。)
grantee	権限受領者(ユーザー、ロールまたは PUBLIC)
grant_option	TRUE に設定すると、権限受領者はこのプロシージャを使用して他のユーザーまたはロールにシステム権限を付与できます。

例12-22 Point-to-Pointキュー権限の付与

```
QueueSession    q_sess;
Queue            queue;

((AQjmsDestination)queue).grantQueuePrivilege(
    q_sess, "ENQUEUE", "scott", false);
```

Point-to-Pointキュー権限の取消し

```
public void revokeQueuePrivilege(javax.jms.Session session,
                                  java.lang.String privilege,
                                  java.lang.String grantee)
    throws JMSEException
```

このメソッドは、Point-to-Pointモデルでキュー権限を取り消します。初期設定では、キュー表の所有者のみがこのプロシージャを使用してそのキューの権限を付与できます。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
privilege	ENQUEUE、DEQUEUE または ALL (ALL は両方を意味します。)
grantee	権限が取り消される権限受領者(ユーザー、ロールまたは PUBLIC)

権限を取り消すには、取消し実行者がその権限の付与者である必要があります。また、GRANTオプションによって伝播された権限は、伝播させた付与者の権限が取り消されたときに取り消されます。

例12-23 Point-to-Pointキュー権限の取消し

```
QueueSession    q_sess;
Queue            queue;

((AQjmsDestination)queue).revokeQueuePrivilege(q_sess, "ENQUEUE", "scott");
```


宛先の管理

この項の内容は次のとおりです。

- [宛先の開始](#)
- [宛先の停止](#)
- [宛先の変更](#)
- [宛先の削除](#)



注意:

現在、JMS シャード・キューは、DBMS_AQADM PL/SQL API を介してのみ管理できます。

宛先の開始

```
public void start(javax.jms.Session session,
                 boolean enqueue,
                 boolean dequeue)
    throws JMSEException
```

このメソッドは、宛先を開始します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
enqueue	TRUE に設定すると、エンキューが使用可能になります。
デキュー	TRUE に設定すると、デキューが使用可能になります。

例12-24 宛先の開始

```
TopicSession t_sess;
QueueSession q_sess;
Topic        topic;
Queue        queue;

(AQjmsDestination) topic.start(t_sess, true, true);
(AQjmsDestination) queue.start(q_sess, true, true);
```

宛先の停止

```
public void stop(javax.jms.Session session,
                 boolean enqueue,
                 boolean dequeue,
                 boolean wait)
    throws JMSEException
```

このメソッドは、宛先を停止します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
enqueue	TRUE に設定すると、エンキューが使用不可になります。
デキュー	TRUE に設定すると、デキューが使用不可になります。
wait	TRUE に設定すると、キュー/トピックの保留トランザクションは、宛先が停止する前に完了できません。

例12-25 宛先の停止

```
TopicSession t_sess;  
Topic topic;  
  
((AQjmsDestination)topic).stop(t_sess, true, false);
```

宛先の変更

```
public void alter(javax.jms.Session session,  
                 oracle.jms.AQjmsDestinationProperty dest_property)  
    throws JMSException
```

このメソッドは、宛先を変更します。これには、次のプロパティがあります。

パラメータ	説明
session	JMS セッション
dest_property	キューまたはトピックの新規プロパティ

例12-26 宛先の変更

```
QueueSession q_sess;  
Queue queue;  
TopicSession t_sess;  
Topic topic;  
AQjmsDestinationProperty dest_prop1, dest_prop2;  
  
((AQjmsDestination)queue).alter(dest_prop1);  
((AQjmsDestination)topic).alter(dest_prop2);
```

宛先の削除

```
public void drop(javax.jms.Session session)  
    throws JMSException
```

このメソッドは、宛先を削除します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション

例12-27 宛先の削除

```
QueueSession q_sess;
Queue queue;
TopicSession t_sess;
Topic topic;

((AQjmsDestination) queue). drop (q_sess);
((AQjmsDestination) topic). drop (t_sess);
```

伝播スケジュール

この項の内容は次のとおりです。

- [伝播のスケジューリング](#)
- [伝播スケジュールの有効化](#)
- [伝播スケジュールの変更](#)
- [伝播スケジュールの無効化](#)
- [伝播スケジュールの解除](#)

注意:



現在、JMS シャード・キューは、DBMS_AQADM PL/SQL API を介してのみ管理され、伝播をサポートしていません。

伝播のスケジューリング

```
public void schedulePropagation(javax.jms.Session session,
                               java.lang.String destination,
                               java.util.Date start_time,
                               java.lang.Double duration,
                               java.lang.String next_time,
                               java.lang.Double latency)
    throws JMSEException
```

このメソッドは、伝播をスケジューリングします。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
destination	伝播がスケジューリングされているリモート・データベースのデータベース・リンク。文字列 NULL は、伝播がトピックのデータベース内のすべてのサブスクライバに対してスケジューリングされていることを示

パラメータ	説明
	します。
start_time	伝播開始時刻。
duration	伝播継続時間。
next_time	次回の伝播開始時刻。
latency	許容可能な待機時間(秒単位)。待機時間は、メッセージがエンキューされた時間と伝播された時間の差異です。

[メッセージ](#)受信者が、同一または異なるキュー内の同じ宛先に複数存在する場合、メッセージはすべての受信者に同時に伝播されます。

例12-28 伝播のスケジューリング

```
TopicSession t_sess;
Topic        topic;

((AQjmsDestination)topic).schedulePropagation(
    t_sess, null, null, null, null, new Double(0));
```

伝播スケジュールの有効化

```
public void enablePropagationSchedule(javax.jms.Session session,
                                     java.lang.String destination)
    throws JMSException
```

このメソッドは、伝播スケジュールを有効化します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
destination	宛先データベースのデータベース・リンク。NULL 文字列は、ローカル・データベースに伝播されることを意味します。

例12-29 伝播スケジュールの有効化

```
TopicSession    t_sess;
Topic            topic;

((AQjmsDestination)topic).enablePropagationSchedule(t_sess, "dbs1");
```

伝播スケジュールの変更

```
public void alterPropagationSchedule(javax.jms.Session session,
                                     java.lang.String destination,
                                     java.lang.Double duration,
```

```
java.lang.String next_time,  
java.lang.Double latency)  
throws JMSException
```

このメソッドは、伝播スケジュールを変更します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
destination	伝播がスケジュールされているリモート・データベースのデータベース・リンク。文字列 NULL は、伝播がトピックのデータベース内のすべてのサブスクライバに対してスケジュールされていることを示します。
duration	伝播継続時間。
next_time	次回の伝播開始時刻。
latency	許容可能な待機時間(秒単位)。待機時間は、メッセージがエンキューされた時間と伝播された時間の差異です。

例12-30 伝播スケジュールの変更

```
TopicSession t_sess;  
Topic topic;  
  
((AQjmsDestination) topic).alterPropagationSchedule(  
    t_sess, null, 30, null, new Double(30));
```

伝播スケジュールの無効化

```
public void disablePropagationSchedule(javax.jms.Session session,  
                                       java.lang.String destination)  
    throws JMSException
```

このメソッドは、伝播スケジュールを無効化します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
destination	宛先データベースのデータベース・リンク。NULL 文字列は、ローカル・データベースに伝播されることを意味します。

例12-31 伝播スケジュールの無効化

```
TopicSession t_sess;  
Topic topic;  
  
((AQjmsDestination) topic).disablePropagationSchedule(t_sess, "dbs1");
```

伝播スケジュールの解除

```
public void unschedulePropagation(javax.jms.Session session,  
                                 java.lang.String destination)  
    throws JMSException
```

このメソッドは、スケジュール済伝播のスケジュールを解除します。次のパラメータがあります。

パラメータ	説明
session	JMS セッション
destination	宛先データベースのデータベース・リンク。NULL 文字列は、ローカル・データベースに伝播されることを意味します。

例12-32 伝播スケジュールの解除

```
TopicSession t_sess;  
Topic topic;  
  
((AQjmsDestination) topic).unschedulePropagation(t_sess, "dbs1");
```

13 Oracle Java Message ServiceのPoint-to-Point

次のトピックでは、Oracle Database Advanced Queuing (AQ)の[Java Message Service \(JMS\)](#)操作インタフェースのうち、Point-to-Point操作固有のコンポーネントについて説明します。Point-to-Pointおよびパブリッシュ/サブスクライブで共有されるコンポーネントについては、[「Oracle Java Message Serviceの共有インタフェース」](#)を参照してください。

- [ユーザー名/パスワードが設定されたConnectionの作成](#)
- [コネクションの確立: デフォルトのConnectionFactoryパラメータの使用](#)
- [ユーザー名/パスワードが設定されたQueueConnectionの作成](#)
- [QueueConnectionの確立: オープンしているJDBCコネクションの使用](#)
- [QueueConnectionの確立: デフォルトのConnectionFactoryパラメータの使用](#)
- [QueueConnectionの確立: オープンしているOracleOCIConnectionPoolの使用](#)
- [セッションの作成](#)
- [QueueSessionの作成](#)
- [QueueSenderの作成](#)
- [メッセージの送信: デフォルトの送信オプションのQueueSenderの使用](#)
- [メッセージの送信: 送信オプションを指定したQueueSenderの使用](#)
- [QueueBrowserの作成: 標準JMS型メッセージ](#)
- [QueueBrowserの作成: 標準JMS型メッセージ、メッセージのロック](#)
- [QueueBrowserの作成: Oracleオブジェクト型メッセージ](#)
- [QueueBrowserの作成: Oracleオブジェクト型メッセージ、メッセージのロック](#)
- [QueueReceiverの作成: 標準JMS型メッセージ](#)
- [QueueReceiverの作成: Oracleオブジェクト型メッセージ](#)

ユーザー名/パスワードが設定されたConnectionの作成

```
public javax.jms.Connection createConnection(  
    java.lang.String username,  
    java.lang.String password)  
    throws JMSException
```

このメソッドは、指定されたユーザー名とパスワードを使用して、Point-to-Point操作とパブリッシュ/サブスクライブ操作の両方をサポートするコネクションを作成します。これは新規メソッドで、JMSバージョン1.1仕様をサポートしています。次のパラメータがあります。

パラメータ	説明
username	キュー用にデータベースに接続するユーザー名
password	サーバーへの接続を作成するためのパスワード

コネクションの確立: デフォルトのConnectionFactoryパラメータの使用

```
public javax.jms.Connection createConnection()
    throws JMSException
```

このメソッドは、デフォルトの[ConnectionFactory](#)パラメータを使用して、Point-to-Point操作とパブリッシュ・サブスクライブ操作の両方をサポートするコネクションを作成します。これは新規メソッドで、JMSバージョン1.1仕様をサポートしています。ConnectionFactoryプロパティにデフォルトのユーザー名およびパスワードを含めないと、JMSExceptionが発生します。

ユーザー名/パスワードが設定されたQueueConnectionの作成

```
public javax.jms.QueueConnection createQueueConnection(
    java.lang.String username,
    java.lang.String password)
    throws JMSException
```

このメソッドは、指定されたユーザー名とパスワードを使用して[キュー](#)・コネクションを作成します。次のパラメータがあります。

パラメータ	説明
username	キュー用にデータベースに接続するユーザー名
password	サーバーへの接続を作成するためのパスワード

例13-1 ユーザー名/パスワードが設定されたQueueConnectionの作成

```
QueueConnectionFactory qc_fact = AQjmsFactory.getQueueConnectionFactory(
    "sun123", "oratest", 5521, "thin");
QueueConnection qc_conn = qc_fact.createQueueConnection("jmsuser", "jmsuser");
```

QueueConnectionの確立: オープンしているJDBCコネクションの使用

```
public static javax.jms.QueueConnection createQueueConnection(
    java.sql.Connection jdbc_connection)
    throws JMSException
```

このメソッドは、オープンしているJDBCコネクションを使用してキュー・コネクションを作成します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
jdbc_connection	データベースへの有効なオープン・コネクション

ユーザーがJMS操作に対して既存の(たとえばコネクション・プールの)JDBCコネクションを使用するときに、[例13-2](#)のメソッドを使用できます。この場合、JMSは新しいコネクションをオープンせずに、提供されたJDBCコネクションを使用してJMS QueueConnectionオブジェクトを作成します。

[例13-3](#)のメソッドは、データベース(JDBCサーバー・ドライバ)内のJavaストア・プロシージャからJMSを使用する場合に、JMS QueueConnectionを確立する唯一の方法です。

例13-2 QueueConnectionの確立: オープンしているJDBCコネクションの使用

```
Connection db_conn; /* previously opened JDBC connection */
```



```
QueueConnection qc_conn = AQjmsQueueConnectionFactory.createQueueConnection(  
    db_conn);
```

例13-3 データベース内のJavaプロシージャからのQueueConnectionの作成

```
OracleDriver ora = new OracleDriver();  
QueueConnection qc_conn = AQjmsQueueConnectionFactory.createQueueConnection(ora.defaultConnection());
```

QueueConnectionの確立: デフォルトのConnectionFactoryパラメータの使用

```
public javax.jms.QueueConnection createQueueConnection()  
    throws JMSEException
```

このメソッドは、デフォルトの[ConnectionFactory](#)パラメータによってキュー・コネクションを確立します。

QueueConnectionFactoryのプロパティにデフォルトのユーザー名およびパスワードを含めないと、JMSEExceptionが発生します。

QueueConnectionの確立: オープンしているOracleOCIConnectionPoolの使用

```
public static javax.jms.QueueConnection createQueueConnection(  
    oracle.jdbc.pool.OracleOCIConnectionPool cpool)  
    throws JMSEException
```

このメソッドは、オープンしているOracleOCIConnectionPoolを使用してキュー・コネクションを作成します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
cpool	データベースに対してオープンな OCI コネクション・プール

ユーザーがJMS操作に対して既存のOracleOCIConnectionPoolインスタンスを使用するとき、[例13-4](#)のメソッドを使用できます。この場合、JMSは新しいOracleOCIConnectionPoolインスタンスをオープンせずに、提供されたOracleOCIConnectionPoolインスタンスを使用してJMS QueueConnectionオブジェクトを作成します。

例13-4 QueueConnectionの確立: オープンしているOracleOCIConnectionPoolの使用

```
OracleOCIConnectionPool cpool; /* previously created OracleOCIConnectionPool */  
QueueConnection qc_conn = AQjmsQueueConnectionFactory.createQueueConnection(cpool);
```

セッションの作成

```
public javax.jms.Session createSession(boolean transacted,  
    int ack_mode)  
    throws JMSEException
```

このメソッドは、Point-to-Point操作とパブリッシュ・サブスクライブ操作の両方をサポートするSessionを作成します。これは新規メソッドで、JMSバージョン1.1仕様をサポートしています。トランザクションおよび[非トランザクション](#)・セッションがサポートされています。次のパラメータがあります。

パラメータ	説明
transacted	TRUE に設定すると、セッションは トランザクション になります。
ack_mode	コンシューマまたはクライアントが受信したメッセージを認識するかどうかを示します。トランザクション処理のセッションの場合は無視されます。有効な値は Session.AUTO_ACKNOWLEDGE、Session.CLIENT_ACKNOWLEDGE および Session.DUPS_OK_ACKNOWLEDGE です。

QueueSessionの作成

```
public javax.jms.QueueSession createQueueSession(
    boolean transacted, int ack_mode)
    throws JMSException
```

このメソッドはQueueSessionを作成します。トランザクションおよび非トランザクション・セッションがサポートされています。次のパラメータがあります。

パラメータ	説明
transacted	TRUE に設定すると、セッションは トランザクション になります。
ack_mode	コンシューマまたはクライアントが受信したメッセージを認識するかどうかを示します。トランザクション処理のセッションの場合は無視されます。有効な値は Session.AUTO_ACKNOWLEDGE、Session.CLIENT_ACKNOWLEDGE および Session.DUPS_OK_ACKNOWLEDGE です。

例13-5 トランザクションQueueSessionの作成

```
QueueConnection qc_conn;
QueueSession q_sess = qc_conn.createQueueSession(true, 0);
```

QueueSenderの作成

```
public javax.jms.QueueSender createSender(javax.jms.Queue queue)
    throws JMSException
```

このメソッドはQueueSenderを作成します。送信者の作成時にデフォルト・キューが設定されていない場合は、すべての[送信](#)操作で宛先キューを指定する必要があります。次のパラメータがあります。

パラメータ	説明
queue	宛先キューの名前

メッセージの送信: デフォルトの送信オプションのQueueSenderの使用

```
public void send(javax.jms.Queue queue,
    javax.jms.Message message)
    throws JMSException
```

このメソッドは、[メッセージ](#)をデフォルトの送信オプションでQueueSenderを使用して送信します。この操作では、メッセージの

priority(1)とtimeToLive(infinite)のデフォルト値を使用します。次のパラメータがあります。

パラメータ	説明
queue	このメッセージを送信するためのキュー
message	送信するメッセージ

QueueSenderがデフォルト・キューで作成されている場合、必ずしもsend()コールにqueueパラメータを指定する必要はありません。キューがsend()操作で指定されている場合、この値はQueueSenderのデフォルト・キューをオーバーライドします。

QueueSenderがデフォルト・キューを使用しないで作成されている場合、send()コールごとにqueueパラメータを指定する必要があります。

例13-6 任意のキューにメッセージを送信するセnderの作成

```
/* Create a sender to send messages to any queue */
QueueSession jms_sess;
QueueSender sender1;
TextMessage message;
sender1 = jms_sess.createSender (null);
sender1.send (queue, message);
```

例13-7 特定のキューにメッセージを送信するセnderの作成

```
/* Create a sender to send messages to a specific queue */
QueueSession jms_sess;
QueueSender sender2;
Queue billed_orders_que;
TextMessage message;
sender2 = jms_sess.createSender (billed_orders_que);
sender2.send (queue, message);
```

メッセージの送信: 送信オプションを指定したQueueSenderの使用

```
public void send(javax.jms.Queue queue,
                javax.jms.Message message,
                int deliveryMode,
                int priority,
                long timeToLive)
    throws JMSEException
```

このメソッドは、送信オプションを指定してQueueSenderを使用してメッセージを送信します。次のパラメータがあります。

パラメータ	説明
queue	このメッセージを送信するためのキュー
message	送信するメッセージ
deliveryMode	使用する配信モード

パラメータ	説明
priority	このメッセージの優先順位
timeToLive	ミリ秒で指定されるメッセージの保存時間(ゼロは無制限)

QueueSenderがデフォルト・キューで作成されている場合、必ずしもsend()コールにqueueパラメータを指定する必要はありません。キューがsend()操作で指定されている場合、この値はQueueSenderのデフォルト・キューをオーバーライドします。

QueueSenderがデフォルト・キューを使用しないで作成されている場合、send()コールごとにqueueパラメータを指定する必要があります。

例13-8 メッセージの送信: 送信オプションを指定したQueueSenderの使用1

```
/* Create a sender to send messages to any queue */
/* Send a message to new_orders_que with priority 2 and timetoLive 100000
   milliseconds */
QueueSession jms_sess;
QueueSender sender1;
TextMessage msg;
Queue new_orders_que
sender1 = jms_sess.createSender(null);
sender1.send(new_orders_que, msg, DeliveryMode.PERSISTENT, 2, 100000);
```

例13-9 メッセージの送信: 送信オプションを指定したQueueSenderの使用2

```
/* Create a sender to send messages to a specific queue */
/* Send a message with priority 1 and timetoLive 400000 milliseconds */
QueueSession jms_sess;
QueueSender sender2;
Queue billed_orders_que;
TextMessage msg;
sender2 = jms_sess.createSender(billed_orders_que);
sender2.send(msg, DeliveryMode.PERSISTENT, 1, 400000);
```

QueueBrowserの作成: 標準JMS型メッセージ

```
public javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,
                                           java.lang.String messageSelector)
                                           throws JMSEException
```

このメソッドは、Text、Stream、Objects、BytesまたはMapMessageメッセージ本体を使用するキューに対するQueueBrowserを作成します。次のパラメータがあります。

パラメータ	説明
queue	アクセス対象のキュー。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。

java.util Enumeration内のメソッドを使用して、メッセージのリストを参照してください。

関連項目:

[「MessageSelector」](#)

例13-10 QueueBrowserの作成: セレクタの指定なし

```
/* Create a browser without a selector */
QueueSession    jms_session;
QueueBrowser    browser;
Queue           queue;
browser = jms_session.createBrowser(queue);
```

例13-11 QueueBrowserの作成: セレクタの指定あり

```
/* Create a browser for queues with a specified selector */
QueueSession    jms_session;
QueueBrowser    browser;
Queue           queue;
/* create a Browser to look at messages with correlationID = RUSH */
browser = jms_session.createBrowser(queue, "JMSCorrelationID = 'RUSH'");
```

QueueBrowserの作成: 標準JMS型メッセージ、メッセージのロック

```
public javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,
                                           java.lang.String messageSelector,
                                           boolean locked)
    throws JMSException
```

このメソッドは、TextMessage、StreamMessage、ObjectMessage、BytesMessageまたはMapMessageメッセージ本体を使用するキューに対して、ブラウザ中にメッセージをロックするQueueBrowserを作成します。ロックされたメッセージは、ブラウザ・セッションによってトランザクションが終了されるまで、他のコンシューマによって削除できません。次のパラメータがあります。

パラメータ	説明
queue	アクセス対象のキュー。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。
locked	TRUE に設定すると、ブラウザ中のメッセージはロックされます(UPDATE の SELECT と同様)。

例13-12 QueueBrowserの作成: セレクタの指定なし、メッセージをロック

```
/* Create a browser without a selector */
QueueSession    jms_session;
QueueBrowser    browser;
Queue           queue;
browser = jms_session.createBrowser(queue, null, true);
```

例13-13 QueueBrowserの作成: セレクタの指定あり、メッセージをロック

```
/* Create a browser for queues with a specified selector */
QueueSession    jms_session;
QueueBrowser    browser;
Queue           queue;
/* create a Browser to look at messages with
```

```
correlationID = RUSH in lock mode */
browser = jms_session.createBrowser(queue, "JMSCorrelationID = 'RUSH'", true);
```

QueueBrowserの作成: Oracleオブジェクト型メッセージ

```
public javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,
                                           java.lang.String messageSelector,
                                           java.lang.Object payload_factory)
    throws JMSEException
```

このメソッドは、Oracle [オブジェクト型](#) メッセージ・キューに対するQueueBrowserを作成します。次のパラメータがあります。

パラメータ	説明
queue	アクセス対象のキュー。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory

SQLオブジェクト型ペイロードにマップされる特定のJavaクラスに対するCustomDatumFactoryは、静的メソッドgetFactoryを使用して取得できます。

注意:



CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。

キューtest_queueは、SCOTT. EMPLOYEE型のペイロードを持ち、このOracleオブジェクト型に対してJPublisherによって生成されるJavaクラスがEmployeeであると仮定します。Employeeクラスは、CustomDatumインターフェースを実装します。このクラスに対するCustomDatumFactoryは、Employee.getFactory()メソッドを使用して取得できます。

注意:



シャード・キューは、オブジェクト型メッセージをサポートしていません。

関連項目:

[「MessageSelector」](#)

例13-14 ADTMessageに対するQueueBrowserの作成

```
/* Create a browser for a Queue with AdtMessage messages of type EMPLOYEE*/
QueueSession jms_session
QueueBrowser browser;
```

```

Queue      test_queue;
browser = ((AQjmsSession) jms_session).createBrowser (test_queue,
                                                    "corrid=' EXPRESS' ",
                                                    Employee.getFactory ());

```

QueueBrowserの作成: Oracleオブジェクト型メッセージ、メッセージのロック

```

public javax.jms.QueueBrowser createBrowser (javax.jms.Queue queue,
                                             java.lang.String messageSelector,
                                             java.lang.Object payload_factory,
                                             boolean locked)
    throws JMSException

```

このメソッドは、Oracleオブジェクト型メッセージ・キューに対して、ブラウズ中にメッセージをロックするQueueBrowserを作成します。次のパラメータがあります。

パラメータ	説明
queue	アクセス対象のキュー。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory
locked	TRUE に設定すると、ブラウズ中のメッセージはロックされます(UPDATE の SELECT と同様)。

注意:



CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。

注意:



シャード・キューは、オブジェクト型メッセージをサポートしていません。

例13-15 AdtMessageキューに対するQueueBrowserの作成、メッセージをロック

```

/* Create a browser for a Queue with AdtMessage messages of type EMPLOYEE* in lock mode/
QueueSession jms_session
QueueBrowser browser;
Queue      test_queue;
browser = ((AQjmsSession) jms_session).createBrowser (test_queue,
                                                    null,
                                                    Employee.getFactory (),
                                                    true);

```

QueueReceiverの作成: 標準JMS型メッセージ

```
public javax.jms.QueueReceiver createReceiver (javax.jms.Queue queue,  
                                              java.lang.String messageSelector)  
    throws JMSException
```

このメソッドは、標準JMS型メッセージ・キューに対するQueueReceiverを作成します。次のパラメータがあります。

パラメータ	説明
queue	アクセス対象のキュー。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。

関連項目:

[「MessageSelector」](#)

例13-16 QueueReceiverの作成: セレクタの指定なし

```
/* Create a receiver without a selector */  
QueueSession    jms_session  
QueueReceiver   receiver;  
Queue           queue;  
receiver = jms_session.createReceiver (queue);
```

例13-17 QueueReceiverの作成: セレクタの指定あり

```
/* Create a receiver for queues with a specified selector */  
QueueSession    jms_session;  
QueueReceiver   receiver;  
Queue           queue;  
/* create Receiver to receive messages with correlationID starting with EXP */  
receiver = jms_session.createReceiver (queue, "JMSCorrelationID LIKE 'EXP%'");
```

QueueReceiverの作成: Oracleオブジェクト型メッセージ

```
public javax.jms.QueueReceiver createReceiver (javax.jms.Queue queue,  
                                              java.lang.String messageSelector,  
                                              java.lang.Object payload_factory)  
    throws JMSException
```

このメソッドは、Oracleオブジェクト型メッセージ・キューに対するQueueReceiverを作成します。次のパラメータがあります。

パラメータ	説明
queue	アクセス対象のキュー。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。

パラメータ	説明
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory

SQLオブジェクト型ペイロードにマップされる特定のJavaクラスに対するCustomDatumFactoryは、静的メソッドgetFactoryを使用して取得できます。

注意:



CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。

キューtest_queueは、SCOTT. EMPLOYEE型のペイロードを持ち、このOracleオブジェクト型に対してJPublisherによって生成されるJavaクラスがEmployeeであると仮定します。Employeeクラスは、CustomDatumインタフェースを実装します。このクラスに対するORADDataFactoryは、Employee.getFactory()メソッドを使用して取得できます。

注意:



シャード・キューは、オブジェクト型メッセージをサポートしていません。

関連項目:

[「MessageSelector」](#)

例13-18 QueueReceiverの作成: AdtMessageメッセージ

```
/* Create a receiver for a Queue with AdtMessage messages of type EMPLOYEE*/
QueueSession jms_session
QueueReceiver receiver;
Queue        test_queue;
browser = ((AQjmsSession) jms_session).createReceiver(
    test_queue,
    "JMSCorrelationID = 'MANAGER',
    Employee.getFactory());
```

14 Oracle Java Message Serviceのパブリッシュ/サブスクライブ

次のトピックでは、Oracle Database Advanced Queuing (AQ)の[Java Message Service \(JMS\)](#)操作インターフェースのうち、[パブリッシュ・サブスクライブ](#)操作固有のコンポーネントについて説明します。Point-to-Pointおよびパブリッシュ/サブスクライブで共有されるコンポーネントについては、[「Oracle Java Message Serviceの共有インターフェース」](#)を参照してください。

- [ユーザー名/パスワードが設定されたConnectionの作成](#)
- [コネクションの確立: デフォルトのConnectionFactoryパラメータの使用](#)
- [ユーザー名/パスワードが設定されたTopicConnectionの作成](#)
- [TopicConnectionの確立: オープンしているJDBCコネクションの使用](#)
- [TopicConnectionの確立: オープンしているOracleOCIConnectionPoolの使用](#)
- [セッションの作成](#)
- [TopicSessionの作成](#)
- [TopicPublisherの作成](#)
- [メッセージのパブリッシュ: 最小限の指定](#)
- [トピック指定によるメッセージのパブリッシュ](#)
- [メッセージのパブリッシュ: 配信モード、優先順位およびTimeToLiveの指定](#)
- [メッセージのパブリッシュ: 受信者リストの指定](#)
- [JMSトピックに対するDurableSubscriberの作成: セレクタの指定なし](#)
- [JMSトピックに対するDurableSubscriberの作成: セレクタの指定あり](#)
- [Oracleオブジェクト型トピックに対するDurableSubscriberの作成: セレクタの指定なし](#)
- [Oracleオブジェクト型トピックに対するDurableSubscriberの作成: セレクタの指定あり](#)
- [トピック・サブスクライバ作成時の変換の指定](#)
- [リモート・サブスクライバの作成: JMSメッセージ](#)
- [リモート・サブスクライバの作成: Oracleオブジェクト型メッセージ](#)
- [リモート・サブスクライバ作成時の変換の指定](#)
- [永続サブスクリプションのサブスクライブの解除: ローカル・サブスクライバ](#)
- [永続サブスクリプションのサブスクライブの解除: リモート・サブスクライバ](#)
- [TopicReceiverの作成: 標準JMS型メッセージ・トピック](#)
- [TopicReceiverの作成: Oracleオブジェクト型メッセージ・トピック](#)
- [TopicBrowserの作成: 標準JMSメッセージ](#)
- [TopicBrowserの作成: 標準JMSメッセージ、メッセージのロック](#)
- [TopicBrowserの作成: Oracleオブジェクト型メッセージ](#)

- [TopicBrowserの作成: Oracleオブジェクト型メッセージ、メッセージのロック](#)
- [TopicBrowserを使用したメッセージのブラウズ](#)

ユーザー名/パスワードが設定されたConnectionの作成

```
public javax.jms.Connection createConnection(
    java.lang.String username,
    java.lang.String password)
    throws JMSException
```

このメソッドは、指定されたユーザー名とパスワードを使用して、Point-to-Point操作とパブリッシュ/サブスクライブ操作の両方をサポートするコネクションを作成します。これは新規メソッドで、JMSバージョン1.1仕様をサポートしています。次のパラメータがあります。

パラメータ	説明
username	キュー用にデータベースに接続するユーザー名
password	サーバーへの接続を作成するためのパスワード

コネクションの確立: デフォルトのConnectionFactoryパラメータの使用

```
public javax.jms.Connection createConnection()
    throws JMSException
```

このメソッドは、デフォルトの[ConnectionFactory](#)パラメータを使用して、Point-to-Point操作とパブリッシュ・サブスクライブ操作の両方をサポートするコネクションを作成します。これは新規メソッドで、JMSバージョン1.1仕様をサポートしています。ConnectionFactoryプロパティにデフォルトのユーザー名およびパスワードを含めないと、JMSExceptionが発生します。

ユーザー名/パスワードが設定されたTopicConnectionの作成

```
public javax.jms.TopicConnection createTopicConnection(
    java.lang.String username,
    java.lang.String password)
    throws JMSException
```

このメソッドは、指定されたユーザー名とパスワードを使用してTopicConnectionを作成します。次のパラメータがあります。

パラメータ	説明
username	キュー用にデータベースに接続するユーザー名
password	サーバーへの接続を作成するためのパスワード

例14-1 ユーザー名/パスワードが設定されたTopicConnectionの作成

```
TopicConnectionFactory tc_fact = AQjmsFactory.getTopicConnectionFactory("sun123", "oratest", 5521,
"thin");
/* Create a TopicConnection using a username/password */
TopicConnection tc_conn = tc_fact.createTopicConnection("jmsuser", "jmsuser");
```

TopicConnectionの確立: オープンしているJDBCコネクションの使用

```
public static javax.jms.TopicConnection createTopicConnection(  
    java.sql.Connection jdbc_connection)  
    throws JMSEException
```

このメソッドは、オープンしているJDBCコネクションを使用してTopicConnectionを作成します。次のパラメータがあります。

パラメータ	説明
jdbc_connection	データベースへの有効なオープン・コネクション

例14-2 TopicConnectionの確立: オープンしているJDBCコネクションの使用

```
Connection db_conn; /*previously opened JDBC connection */  
TopicConnection tc_conn =  
AQjmsTopicConnectionFactory.createTopicConnection(db_conn);
```

例14-3 TopicConnectionの確立: 新規JDBCコネクションの使用

```
OracleDriver ora = new OracleDriver();  
TopicConnection tc_conn =  
AQjmsTopicConnectionFactory.createTopicConnection(ora.defaultConnection());
```

TopicConnectionの確立: オープンしているOracleOCIConnectionPoolの使用

```
public static javax.jms.TopicConnection createTopicConnection(  
    oracle.jdbc.pool.OracleOCIConnectionPool cpool)  
    throws JMSEException
```

このメソッドは、オープンしているOracleOCIConnectionPoolを使用してTopicConnectionを作成します。これは静的であり、次のパラメータを取ります。

パラメータ	説明
cpool	データベースに対してオープンな OCI コネクション・プール

例14-4 TopicConnectionの確立: オープンしているOracleOCIConnectionPoolの使用

```
OracleOCIConnectionPool cpool; /* previously created OracleOCIConnectionPool */  
TopicConnection tc_conn =  
AQjmsTopicConnectionFactory.createTopicConnection(cpool);
```

セッションの作成

```
public javax.jms.Session createSession(boolean transacted,  
    int ack_mode)  
    throws JMSEException
```

このメソッドは、Point-to-Point操作とパブリッシュ・サブスクライブ操作の両方をサポートするSessionを作成します。これは新規で、JMSバージョン1.1仕様をサポートしています。次のパラメータがあります。

パラメータ	説明
transacted	TRUE に設定すると、セッションは トランザクション になります。
ack_mode	コンシューマまたはクライアントが受信したメッセージを認識するかどうかを示します。トランザクション処理のセッションの場合は無視されます。有効な値は <code>Session.AUTO_ACKNOWLEDGE</code> 、 <code>Session.CLIENT_ACKNOWLEDGE</code> および <code>Session.DUPS_OK_ACKNOWLEDGE</code> です。

TopicSessionの作成

```
public javax.jms.TopicSession createTopicSession(boolean transacted,
                                                int ack_mode)
    throws JMSEException
```

このメソッドはTopicSessionを作成します。次のパラメータがあります。

パラメータ	説明
transacted	TRUE に設定すると、セッションは トランザクション になります。
ack_mode	コンシューマまたはクライアントが受信したメッセージを認識するかどうかを示します。トランザクション処理のセッションの場合は無視されます。有効な値は <code>Session.AUTO_ACKNOWLEDGE</code> 、 <code>Session.CLIENT_ACKNOWLEDGE</code> および <code>Session.DUPS_OK_ACKNOWLEDGE</code> です。

例14-5 TopicSessionの作成

```
TopicConnection tc_conn;
TopicSession t_sess = tc_conn.createTopicSession(true, 0);
```

TopicPublisherの作成

```
public javax.jms.TopicPublisher createPublisher(javax.jms.Topic topic)
    throws JMSEException
```

このメソッドはTopicPublisherを作成します。次のパラメータがあります。

パラメータ	説明
topic	パブリッシュ対象のトピック。識別されていない プロデューサ の場合は、NULL。

メッセージのパブリッシュ: 最小限の指定

```
public void publish(javax.jms.Message message)
    throws JMSEException
```

このメソッドは、最小限の指定でメッセージをパブリッシュします。次のパラメータがあります。

パラメータ	説明
-------	----

パラメータ**説明**

message 送信するメッセージ

TopicPublisherでは、メッセージのpriority(1)とtimeToLive(infinite)のデフォルト値を使用します。

例14-6 トピック指定なしのパブリッシュ

```
/* Publish without specifying topic */
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicPublisher publisher1;
Topic shipped_orders;
int myport = 5521;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME",
    "MYSID",
    myport,
    "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
/* create TopicSession */
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
/* get shipped orders topic */
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE",
    "Shipped_Orders_Topic");
publisher1 = jms_sess.createPublisher(shipped_orders);
/* create TextMessage */
TextMessage jms_sess.createTextMessage();
/* publish without specifying the topic */
publisher1.publish(text_message);
```

例14-7 相関と遅延の指定によるパブリッシュ

```
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicPublisher publisher1;
Topic shipped_orders;
int myport = 5521;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME",
    "MYSID",
    myport,
    "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE",
    "Shipped_Orders_Topic");
publisher1 = jms_sess.createPublisher(shipped_orders);
/* Create TextMessage */
TextMessage jms_sess.createTextMessage();
/* Set correlation and delay */
/* Set correlation */
jms_sess.setJMSCorrelationID("F00");
```

```

/* Set delay of 30 seconds */
jms_sess.setLongProperty("JMS_OracleDelay", 30);
/* Publish */
publisher1.publish(text_message);

```

トピック指定によるメッセージのパブリッシュ

```

public void publish(javax.jms.Topic topic, javax.jms.Message message)
    throws JMSException

```

このメソッドは、トピックを指定してメッセージをパブリッシュします。次のパラメータがあります。

パラメータ	説明
topic	パブリッシュするトピック
message	送信するメッセージ

TopicPublisherがデフォルトのトピックで作成されている場合、publish() コールにtopicパラメータを指定することはできません。トピックが指定されている場合、その値はTopicPublisherのデフォルトのトピックをオーバーライドします。TopicPublisherがデフォルトのトピックを使用しないで作成されている場合、publish() コールごとにトピックを指定する必要があります。

例14-8 トピック指定によるパブリッシュ

```

/* Publish specifying topic */
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicPublisher publisher1;
Topic shipped_orders;
int myport = 5521;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    'MYHOSTNAME', 'MYSID', myport, 'oci8');
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
/* create TopicPublisher */
publisher1 = jms_sess.createPublisher(null);
/* get topic object */
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    'WS', 'Shipped_Orders_Topic');
/* create text message */
TextMessage text_message = jms_sess.createTextMessage();
/* publish specifying the topic */
publisher1.publish(shipped_orders, text_message);

```

メッセージのパブリッシュ: 配信モード、優先順位およびTimeToLiveの指定

```

public void publish(javax.jms.Topic topic,
    javax.jms.Message message,
    oracle.jms.AQjmsAgent[] recipient_list,
    int deliveryMode,
    int priority,
    long timeToLive)
    throws JMSException

```

このメソッドは、配信モード、優先順位およびTimeToLiveを指定してメッセージをパブリッシュします。次のパラメータがあります。

パラメータ	説明
topic	メッセージのパブリッシュ先のトピック(MessageProducer のデフォルト・トピックをオーバーライドします)。
message	パブリッシュするメッセージ。
recipient_list	メッセージがパブリッシュされる受信者のリスト。受信者の型は AQjmsAgent です。
deliveryMode	PERSISTENT または NON_PERSISTENT(このリリースでサポートされているのは PERSISTENT のみです)
priority	このメッセージの優先順位
timeToLive	ミリ秒で指定されるメッセージの保存時間(ゼロは無制限)

例14-9 優先順位とTimeToLiveの指定によるパブリッシュ

```
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicPublisher publisher1;
Topic shipped_orders;
int myport = 5521;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
publisher1 = jms_sess.createPublisher(shipped_orders);
/* Create TextMessage */
TextMessage text_message = jms_sess.createTextMessage();
/* Publish message with priority 1 and time to live 200 seconds */
publisher1.publish(text_message, DeliveryMode.PERSISTENT, 1, 200000);
```

メッセージのパブリッシュ: 受信者リストの指定

```
public void publish(javax.jms.Message message,
    oracle.jms.AQjmsAgent[] recipient_list)
    throws JMSEException
```

このメソッドは、トピック・サブスクライバをオーバーライドする[受信者](#)リストを指定して、メッセージをパブリッシュします。次のパラメータがあります。

パラメータ	説明
-------	----

パラメータ	説明
message	パブリッシュするメッセージ。
recipient_list	メッセージがパブリッシュされる受信者のリスト。受信者の型は AQjmsAgent です。

例14-10 トピック・サブスクライバをオーバーライドする受信者リストの指定によるパブリッシュ

```

/* Publish specifying priority and timeToLive */
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicPublisher publisher1;
Topic shipped_orders;
int myport = 5521;
AQjmsAgent[] recipList;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
publisher1 = jms_sess.createPublisher(shipped_orders);
/* create TextMessage */
TextMessage text_message = jms_sess.createTextMessage();
/* create two receivers */
recipList = new AQjmsAgent[2];
recipList[0] = new AQjmsAgent(
    "ES", "ES.shipped_orders_topic", AQAgent.DEFAULT_AGENT_PROTOCOL);
recipList[1] = new AQjmsAgent(
    "WS", "WS.shipped_orders_topic", AQAgent.DEFAULT_AGENT_PROTOCOL);
/* publish message specifying a recipient list */
publisher1.publish(text_message, recipList);

```

JMSトピックに対するDurableSubscriberの作成: セレクタの指定なし

```

public javax.jms.TopicSubscriber createDurableSubscriber(
    javax.jms.Topic topic,
    java.lang.String subs_name)
    throws JMSException

```

このメソッドは、セレクタ指定なしの[JMSトピック](#)にDurableSubscriberを作成します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ先の非一時トピック
subs_name	このサブスクリプションの識別に使用される名前

トピックへの排他アクセス

CreateDurableSubscriber() および Unsubscribe() は、どちらもターゲット・トピックに排他的にアクセスする必要があります。これらのコールが適用されるときに同じトピックに対する保留中の JMS send()、publish() または receive() 操作がある場合、

例外ORA-4020が発生します。この問題を解決するには、次の2つの方法があります。

- 設定またはクリーン・アップのフェーズでのcreateDurableSubscriber () およびUnsubscribe () のコールを、トピックに対して保留中の他のJMS操作がない場合に制限します。これによって、必要なリソースが他のJMS操作のコールによって保持されないようになります。
- createDurableSubscriber () またはUnsubscribe () をコールする前に、TopicSession. commitをコールします。

例14-11 JMSトピックに対する永続サブスクライバの作成: セレクタの指定なし

```
TopicConnectionFactory    tc_fact    = null;
TopicConnection          t_conn     = null;
TopicSession             jms_sess;
TopicSubscriber          subscriber1;
Topic                    shipped_orders;
int                       myport    = 5521;
AQJmsAgent[]             recipList;
/* create connection and session */
tc_fact = AQJmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME",
    "MYSID",
    myport,
    "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQJmsSession) jms_sess).getTopic(
    "OE",
    "Shipped_Orders_Topic");
/* create a durable subscriber on the shipped_orders topic*/
subscriber1 = jms_sess.createDurableSubscriber(
    shipped_orders,
    'WesternShipping');
```

JMSトピックに対するDurableSubscriberの作成: セレクタの指定あり

```
public javax.jms.TopicSubscriber createDurableSubscriber(
    javax.jms.Topic topic,
    java.lang.String subs_name,
    java.lang.String messageSelector,
    boolean noLocal)
    throws JMSException
```

このメソッドは、セレクタを指定して、JMSトピックに対する永続サブスクライバを作成します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ先の非一時トピック
subs_name	このサブスクリプションの識別に使用される名前
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がいないことを表します。
noLocal	TRUE に設定すると、独自の接続によってパブリッシュされたメッセージの配信が禁止され

パラメータ	説明
	ます。

クライアントは、同一の名前および異なるmessageSelectorで永続トピック・サブスクライバを作成することによって、既存の永続サブスクリプションを変更できます。トピックのサブスクリプションを終了するには、unsubscribeコールが必要です。

関連項目:

- [「トピックへの排他アクセス」](#)
- [「MessageSelector」](#)

例14-12 JMSトピックに対する永続サブスクライバの作成: セレクタの指定あり

```

TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicSubscriber subscriber1;
Topic shipped_orders;
int myport = 5521;
AQjmsAgent[] recipList;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
/* create a subscriber */
/* with condition on JMSPriority and user property 'Region' */
subscriber1 = jms_sess.createDurableSubscriber(
    shipped_orders, 'WesternShipping',
    "JMSPriority > 2 and Region like 'Western%'", false);

```

Oracleオブジェクト型トピックに対するDurableSubscriberの作成: セレクタの指定なし

```

public javax.jms.TopicSubscriber createDurableSubscriber(
    javax.jms.Topic topic,
    java.lang.String subs_name,
    java.lang.Object payload_factory)
    throws JMSEException

```

このメソッドは、セレクタを指定せずに、Oracle [オブジェクト型](#)のトピックに対する永続サブスクライバを作成します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ先の非一時トピック

パラメータ	説明
subs_name	このサブスクリプションの識別に使用される名前
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADatumFactory

注意:



- CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADatumFactory ペイロード・ファクトリを使用してください。
- シャード・キューは、オブジェクト型メッセージをサポートしていません。

関連項目:

[「トピックへの排他アクセス」](#)

例14-13 Oracleオブジェクト型トピックに対する永続サブスクリバの作成: セレクタの指定なし

```

/* Subscribe to an ADT queue */
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession t_sess = null;
TopicSession jms_sess;
TopicSubscriber subscriber1;
Topic shipped_orders;
int my[port = 5521;
AQjmsAgent[] recipList;
/* the java mapping of the oracle object type created by J Publisher */
ADTMessage message;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
/* create a subscriber, specifying the correct CustomDatumFactory */
subscriber1 = jms_sess.createDurableSubscriber(
    shipped_orders, 'WesternShipping', AQjmsAgent.getFactory());

```

Oracleオブジェクト型トピックに対するDurableSubscriberの作成: セレクタの指定あり

```

public javax.jms.TopicSubscriber createDurableSubscriber(
    javax.jms.Topic topic,
    java.lang.String subs_name,
    java.lang.String messageSelector,
    boolean noLocal,

```

```
java.lang.Object payload_factory)
throws JMSEException
```

このメソッドは、セレクタを指定して、Oracleオブジェクト型のトピックに対する永続サブスクライバを作成します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ先の非一時トピック
subs_name	このサブスクリプションの識別に使用される名前
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がいないことを表します。
noLocal	TRUE に設定すると、独自の接続によってパブリッシュされたメッセージの配信が禁止されます。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory

注意:



- CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。
- シャード・キューは、オブジェクト型メッセージをサポートしていません。

関連項目:

[「トピックへの排他アクセス」](#)

例14-14 Oracleオブジェクト型トピックに対する永続サブスクライバの作成: セレクタの指定あり

```
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession jms_sess;
TopicSubscriber subscriber1;
Topic shipped_orders;
int myport = 5521;
AQjmsAgent[] recipList;
/* the java mapping of the oracle object type created by J Publisher */
ADTMessage message;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
```

```

        "OE", "Shipped_Orders_Topic");
/* create a subscriber, specifying correct CustomDatumFactory and selector */
subscriber1 = jms_sess.createDurableSubscriber(
    shipped_orders, "WesternShipping",
    "priority > 1 and tab.user_data.region like 'WESTERN %'", false,
    ADTMessage.getFactory());

```

トピック・サブスクライバ作成時の変換の指定

キュー/トピックにメッセージを送信/パブリッシュするときに変換を適用できます。メッセージを変換してからキュー/トピックに送信します。

変換は、AQjmsQueueSenderおよびAQjmsTopicPublisherのsetTransformationインタフェースを使用して指定できます。

CreateDurableSubscriber() コールを使用してトピック・サブスクライバを作成する場合も、変換を指定できます。取り出されたメッセージを変換してから、サブスクライバに戻します。指定されたサブスクライバがすでにCreateDurableSubscriber() コールに存在する場合、その変換は指定した変換に設定されます。

例14-15 変換による宛先へのメッセージの送信

注文入力アプリケーションによって処理された注文情報をWS_bookedorders_topicにパブリッシュする必要があると仮定します。正しいフォーマットでトピックにメッセージを挿入するには、前の項で定義した変換OE2WSを使用します。

```

public void ship_bookedorders(
    TopicSession jms_session,
    AQjmsADTMessage adt_message)
{
    TopicPublisher publisher;
    Topic topic;

    try
    {
        /* get a handle to the WS_bookedorders_topic */
        topic = ((AQjmsSession) jms_session).getTopic("WS", "WS_bookedorders_topic");
publisher = jms_session.createPublisher(topic);

        /* set the transformation in the publisher */
        ((AQjmsTopicPublisher) publisher).setTransformation("OE2WS");
publisher.publish(topic, adt_message);
    }
    catch (JMSEException ex)
    {
        System.out.println("Exception : " + ex);
    }
}

```

例14-16 トピック・サブスクライバ作成時の変換の指定

西部地域向け出荷処理アプリケーションは、変換OE2WSとともにOE_bookedorders_topicにサブスクライブします。この変換はメッセージに適用され、Oracleオブジェクト型WS.WS_ordersのメッセージが戻されます。

WSOrder JavaクラスがJPublisherによって生成され、Oracleオブジェクト型WS.WS_orderにマップされると仮定します。:

```

public AQjmsAdtMessage retrieve_bookedorders(TopicSession jms_session)
{
    TopicSubscriber subscriber;
    Topic topic;
    AQjmsAdtMessage msg = null;

```

```

try
{
    /* get a handle to the OE_bookedorders_topic */
    topic = ((AQjmsSession) jms_session).getTopic("OE", "OE_bookedorders_topic");

    /* create a subscriber with the transformation OE2WS */
    subs = ((AQjmsSession) jms_session).createDurableSubscriber(
        topic, 'WShip', null, false, WSOrder.getFactory(), "OE2WS");
    msg = subscriber.receive(10);
}
catch (JMSEException ex)
{
    System.out.println("Exception : " ex);
}
return (AQjmsAdtMessage)msg;
}

```

リモート・サブスクライバの作成: JMSメッセージ

```

public void createRemoteSubscriber(javax.jms.Topic topic,
                                   oracle.jms.AQjmsAgent remote_subscriber,
                                   java.lang.String messageSelector)
    throws JMSEException

```

このメソッドは、JMSメッセージ・トピックに対するリモート・サブスクライバを作成します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ先のトピック。
remote_subscriber	リモート・サブスクライバを表す AQjmsAgent。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がないことを表します。

Oracle Database Advanced Queuingでは、トピックがリモート・サブスクライバ(同一または異なるデータベース内の他のトピックのサブスクライバ)を持つことができます。リモート・サブスクライバを使用するには、ローカル・トピックとリモート・トピック間の[伝播](#)を設定する必要があります。

リモート・サブスクライバは、リモート・トピックの特定のコンシューマまたはリモート・トピックのすべてのサブスクライバにできます。リモート・サブスクライバは、AQjmsAgent構造を使用して定義されます。AQjmsAgentは、名前およびアドレスで構成されます。名前は、リモート・トピックのconsumer_nameを参照します。アドレスはリモート・トピックを参照します。構文は、schema.topic_name[@db link]です。

リモート・トピックで特定のコンシューマに対してメッセージをパブリッシュするには、リモート・トピックでの受信者のsubscription_nameをAQjmsAgentのnameフィールドに指定し、リモート・トピックをaddressフィールドに指定する必要があります。リモート・トピックのすべてのサブスクライバに対してメッセージをパブリッシュするには、AQjmsAgentのnameフィールドをNULLに設定する必要があります。

注意:



シャード・キューは、リモート・サブスクライバをサポートしていません。

関連項目:

[「MessageSelector」](#)

例14-17 リモート・サブスクライバの作成: JMS型メッセージ・トピック

```
TopicConnectionFactory    tc_fact    = null;
TopicConnection          t_conn     = null;
TopicSession             t_sess     = null;
TopicSession             jms_sess;
TopicSubscriber          subscriber1;
Topic                    shipped_orders;
int                       my[port = 5521;
AQjmsAgent                remoteAgent;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
remoteAgent = new AQjmsAgent("WesternRegion", "WS.shipped_orders_topic", null);
/* create a remote subscriber (selector is null)*/
subscriber1 = ((AQjmsSession) jms_sess).createRemoteSubscriber(
    shipped_orders, remoteAgent, null);
```

リモート・サブスクライバの作成: Oracleオブジェクト型メッセージ

```
public void createRemoteSubscriber(javax.jms.Topic topic,
                                   oracle.jms.AQjmsAgent remote_subscriber,
                                   java.lang.String messageSelector,
                                   java.lang.Object payload_factory)
    throws JMSEException
```

このメソッドは、Oracleオブジェクト型メッセージ・トピックに対するリモート・サブスクライバを作成します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ先のトピック。
remote_subscriber	リモート・サブスクライバを表す AQjmsAgent。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がないことを表します。

パラメータ	説明
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADatFactory

注意:



- CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADatFactory ペイロード・ファクトリを使用してください。
- シャード・キューは、リモート・サブスクライバおよびオブジェクト型メッセージをサポートしていません。

Oracle Database Advanced Queuingでは、トピックがリモート・サブスクライバ(同一または異なるデータベース内の他のトピックのサブスクライバ)を持つことができます。リモート・サブスクライバを使用するには、ローカル・トピックとリモート・トピック間の[伝播](#)を設定する必要があります。

リモート・サブスクライバは、リモート・トピックの特定のコンシューマまたはリモート・トピックのすべてのサブスクライバにできます。リモート・サブスクライバは、AQjmsAgent構造を使用して定義されます。AQjmsAgentは、名前およびアドレスで構成されます。名前は、リモート・トピックのconsumer_nameを参照します。アドレスはリモート・トピックを参照します。構文は、schema.topic_name[@dblink]です。

リモート・トピックで特定のコンシューマに対してメッセージをパブリッシュするには、リモート・トピックでの受信者のsubscription_nameをAQjmsAgentのnameフィールドに指定し、リモート・トピックをaddressフィールドに指定する必要があります。リモート・トピックのすべてのサブスクライバに対してメッセージをパブリッシュするには、AQjmsAgentのnameフィールドをNULLに設定する必要があります。

注意:



AQ は、宛先が同じである複数の dblink の使用をサポートしていません。この問題を解決するには、各宛先に対し 1 つのデータベース・リンクを使用します。

関連項目:

[「MessageSelector」](#)

例14-18 リモート・サブスクライバの作成: Oracleオブジェクト型メッセージ・トピック

```

TopicConnectionFactory    tc_fact    = null;
TopicConnection          t_conn    = null;
TopicSession             t_sess    = null;
TopicSession             jms_sess;
TopicSubscriber          subscriber1;
Topic                    shipped_orders;
int                       my[port = 5521;
AQjmsAgent                remoteAgent;
ADTMessage                message;

```

```

/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
/* create TopicSession */
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
/* get the Shipped order topic */
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
/* create a remote agent */
remoteAgent = new AQjmsAgent("WesternRegion", "WS.shipped_orders_topic", null);
/* create a remote subscriber with null selector*/
subscriber1 = ((AQjmsSession) jms_sess).createRemoteSubscriber(
    shipped_orders, remoteAgent, null, message.getFactory());

```

リモート・サブスクライバ作成時の変換の指定

Oracle Database Advanced Queuingによって、リモート・サブスクライバ(他のデータベースにあるサブスクライバ)をトピックにサブスクライブできます。

`createRemoteSubscriber()` コールを使用してリモート・サブスクライバを作成する場合、変換を指定できます。これによって、異なるフォーマットのトピック間でメッセージを伝播できます。トピックに対してパブリッシュされたメッセージがリモート・サブスクライバの基準を満たしている場合、Oracle Database Advanced Queuingはリモート・サブスクライバに指定されているリモート・データベースにあるキュー/トピックにメッセージを自動的に伝播します。変換が指定される場合も、Oracle Database Advanced Queuingはその変換をメッセージに適用してからリモート・データベース上のキュー/トピックに伝播します。

注意:



シャード・キューは、リモート・サブスクライバをサポートしていません。

例14-19 リモート・サブスクライバ作成時の変換の指定

メッセージが自動的にWS.WS_bookedorders_topicに伝播されるようにリモート・サブスクライバをOE.OE_bookedorders_topicに作成します。変換OE2WSは、WS_bookedorders_topicに到達したメッセージが正しいフォーマットとなるように、リモート・サブスクライバを作成するときに指定します。

WSOrder JavaクラスがJPublisherによって生成され、Oracleオブジェクト型WS.WS_orderにマップされると仮定します。

```

public void create_remote_sub(TopicSession jms_session)
{
    AQjmsAgent      subscriber;
    Topic           topic;

    try
    {
        /* get a handle to the OE_bookedorders_topic */
        topic = ((AQjmsSession) jms_session).getTopic("OE", "OE_bookedorders_topic");
        subscriber = new AQjmsAgent("WSHip", "WS.WS_bookedorders_topic");

        ((AQjmsSession) jms_session).createRemoteSubscriber(
            topic, subscriber, null, WSOrder.getFactory(), "OE2WS");
    }
    catch (JMSEException ex)
    {
        System.out.println("Exception : " + ex);
    }
}

```

```
}  
}
```

永続サブスクリプションのサブスクライブの解除: ローカル・サブスクライバ

```
public void unsubscribe(javax.jms.Topic topic,  
                        java.lang.String subs_name)  
    throws JMSEException
```

このメソッドは、ローカル・サブスクライバについて永続サブスクリプションのサブスクライブを解除します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ解除先の非一時トピック
subs_name	このサブスクリプションの識別に使用される名前

関連項目:

[「トピックへの排他アクセス」](#)

例14-20 永続サブスクリプションのサブスクライブの解除: ローカル・サブスクライバ

```
TopicConnectionFactory tc_fact = null;  
TopicConnection t_conn = null;  
TopicSession jms_sess;  
TopicSubscriber subscriber1;  
Topic shipped_orders;  
int myport = 5521;  
AQjmsAgent[] recipList;  
/* create connection and session */  
tc_fact = AQjmsFactory.getTopicConnectionFactory(  
    "MYHOSTNAME", "MYSID", myport, "oci8");  
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");  
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);  
shipped_orders = ((AQjmsSession) jms_sess).getTopic(  
    "OE", "Shipped_Orders_Topic");  
/* unsubscribe "WesternShipping" from shipped_orders */  
jms_sess.unsubscribe(shipped_orders, "WesternShipping");
```

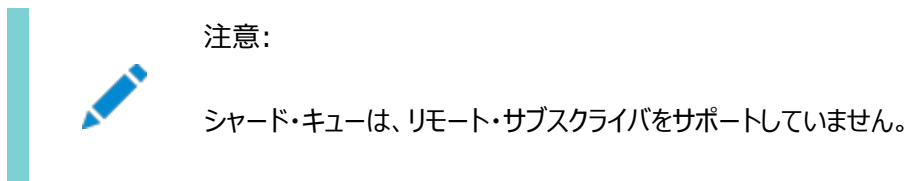
永続サブスクリプションのサブスクライブの解除: リモート・サブスクライバ

```
public void unsubscribe(javax.jms.Topic topic,  
                        oracle.jms.AQjmsAgent remote_subscriber)  
    throws JMSEException
```

このメソッドは、リモート・サブスクライバについて永続サブスクリプションのサブスクライブを解除します。次のパラメータがあります。

パラメータ	説明
topic	サブスクライブ解除先の非一時トピック

パラメータ	説明
remote_subscriber	リモート・サブスクライバを表す AQjmsAgent。AQjmsAgent のアドレス・フィールドは NULL にできません。



関連項目:

[「トピックへの排他アクセス」](#)

例14-21 永続サブスクリプションのサブスクライバの解除: リモート・サブスクライバ

```

TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession t_sess = null;
TopicSession jms_sess;
Topic shipped_orders;
int myport = 5521;
AQjmsAgent remoteAgent;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "OE", "Shipped_Orders_Topic");
remoteAgent = new AQjmsAgent("WS", "WS.Shipped_Orders_Topic", null);
/* unsubscribe the remote agent from shipped_orders */
((AQjmsSession) jms_sess).unsubscribe(shipped_orders, remoteAgent);

```

TopicReceiverの作成: 標準JMS型メッセージ・トピック

```

public oracle.jms.AQjmsTopicReceiver createTopicReceiver(
    javax.jms.Topic topic,
    java.lang.String receiver_name,
    java.lang.String messageSelector)
    throws JMSException

```

このメソッドは、標準JMS型メッセージ・トピックに対するTopicReceiverを作成します。次のパラメータがあります。

パラメータ	説明
topic	アクセスするトピック。
receiver_name	メッセージ・レシーバの名前。

パラメータ	説明
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がいないことを表します。

Oracle Database Advanced Queuingでは、特定の受信者にメッセージを送信できます。これらの受信者は、トピックのサブスクライバである場合とそうでない場合があります。受信者がトピックに対するサブスクライバでない場合、明示的にアドレス指定されているメッセージのみを受信します。このメソッドは、永続サブスクライバ以外のコンシューマに対するTopicReceiverオブジェクトの作成に使用する必要があります。

関連項目:

[「MessageSelector」](#)

例14-22 TopicReceiverの作成: 標準JMS型メッセージ

```

TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession t_sess = null;
TopicSession jms_sess;
Topic shipped_orders;
int myport = 5521;
TopicReceiver receiver;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "WS", "Shipped_Orders_Topic");
receiver = ((AQjmsSession) jms_sess).createTopicReceiver(
    shipped_orders, "WesternRegion", null);

```

TopicReceiverの作成: Oracleオブジェクト型メッセージ・トピック

```

public oracle.jms.AQjmsTopicReceiver createTopicReceiver(
    javax.jms.Topic topic,
    java.lang.String receiver_name,
    java.lang.String messageSelector,
    java.lang.Object payload_factory)
    throws JMSEException

```

このメソッドは、セクタで、Oracleオブジェクト型メッセージ・トピックに対するTopicReceiverを作成します。次のパラメータがあります。

パラメータ	説明
topic	アクセスするトピック。

パラメータ	説明
receiver_name	メッセージ・レシーバの名前。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がないことを表します。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory

注意:



- CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。
- シャード・キューは、オブジェクト型メッセージをサポートしていません。

Oracle Database Advanced Queuingでは、トピックのすべてのサブスクライバ、または特定の受信者にメッセージを送信できます。これらの受信者は、トピックのサブスクライバである場合とそうでない場合があります。受信者がトピックに対するサブスクライバでない場合、明示的にアドレス指定されているメッセージのみを受信します。このメソッドは、永続サブスクライバ以外のコンシューマに対するTopicReceiverオブジェクトの作成に使用する必要があります。

関連項目:

[「MessageSelector」](#)

例14-23 TopicReceiverの作成: Oracleオブジェクト型メッセージ

```

TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession t_sess = null;
TopicSession jms_sess;
Topic shipped_orders;
int myport = 5521;
TopicReceiver receiver;
/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "WS", "Shipped_Orders_Topic");
receiver = ((AQjmsSession) jms_sess).createTopicReceiver(
    shipped_orders, "WesternRegion", null);

```

TopicBrowserの作成: 標準JMSメッセージ

```
public oracle.jms.TopicBrowser createBrowser (javax.jms.Topic topic,
                                             java.lang.String cons_name,
                                             java.lang.String messageSelector)
                                             throws JMSException
```

このメソッドは、TextMessage、StreamMessage、ObjectMessage、BytesMessageまたはMapMessageメッセージ本体を使用するトピックに対するTopicBrowserを作成します。次のパラメータがあります。

パラメータ	説明
topic	アクセスするトピック。
cons_name	永続サブスクライバまたはコンシューマの名前。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がないことを表します。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory

関連項目:

[「MessageSelector」](#)

例14-24 TopicBrowserの作成: セレクタの指定なし

```
/* Create a browser without a selector */
TopicSession jms_session;
TopicBrowser browser;
Topic topic;
browser = ((AQJMSession) jms_session).createBrowser(topic, "SUBS1");
```

例14-25 TopicBrowserの作成: セレクタの指定あり

```
/* Create a browser for topics with a specified selector */
TopicSession jms_session;
TopicBrowser browser;
Topic topic;
/* create a Browser to look at messages with correlationID = RUSH */
browser = ((AQJMSession) jms_session).createBrowser(
    topic, "SUBS1", "JMScorrelationID = 'RUSH'");
```

TopicBrowserの作成: 標準JMSメッセージ、メッセージのロック

```
public oracle.jms.TopicBrowser createBrowser (javax.jms.Topic topic,
                                             java.lang.String cons_name,
                                             java.lang.String messageSelector,
                                             boolean locked)
```

このメソッドは、Text、Stream、Objects、Bytesまたはマップ・メッセージを使用するトピックに対して、ブラウズ中にメッセージをロックするTopicBrowserを作成します。次のパラメータがあります。

パラメータ	説明
topic	アクセスするトピック。
cons_name	永続サブスクライバまたはコンシューマの名前。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector が無いことを表します。
locked	TRUE に設定すると、ブラウズ中のメッセージはロックされます(UPDATE の SELECT と同様)。

例14-26 TopicBrowserの作成: セレクタの指定なし、ブラウズ中にメッセージをロック

```
/* Create a browser without a selector */
TopicSession jms_session;
TopicBrowser browser;
Topic topic;
browser = ((AQJMSession) jms_session).createBrowser(
    topic, "SUBS1", true);
```

例14-27 TopicBrowserの作成: セレクタの指定あり、メッセージをロック

```
/* Create a browser for topics with a specified selector */
TopicSession jms_session;
TopicBrowser browser;
Topic topic;
/* create a Browser to look at messages with correlationID = RUSH in
lock mode */
browser = ((AQJMSession) jms_session).createBrowser(
    topic, "SUBS1", "JMSCorrelationID = 'RUSH'", true);
```

TopicBrowserの作成: Oracleオブジェクト型メッセージ

```
public oracle.jms.TopicBrowser createBrowser (javax.jms.Topic topic,
                                             java.lang.String cons_name,
                                             java.lang.String messageSelector,
                                             java.lang.Object payload_factory)
    throws JMSEException
```

このメソッドは、Oracleオブジェクト型メッセージ・トピックに対するTopicBrowserを作成します。次のパラメータがあります。

パラメータ	説明
topic	アクセスするトピック。

パラメータ	説明
cons_name	永続サブスクライバまたはコンシューマの名前。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がないことを表します。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory

注意:



- CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。
- シャード・キューは、オブジェクト型メッセージをサポートしていません。

SQLオブジェクト型ペイロードにマップされる特定のJavaクラスに対するCustomDatumFactoryは、静的メソッドgetFactoryを使用して取得できます。トピック test_topicは、SCOTT. EMPLOYEE型のペイロードを持ち、このOracleオブジェクト型に対してJpublisherによって生成されるJavaクラスがEmployeeであると仮定します。Employeeクラスは、CustomDatumインタフェースを実装します。このクラスに対するCustomDatumFactoryは、Employee.getFactory()メソッドを使用して取得できます。

関連項目:

[「MessageSelector」](#)

例14-28 TopicBrowserの作成: AdtMessageメッセージ

```
/* Create a browser for a Topic with AdtMessage messages of type EMPLOYEE*/
TopicSession jms_session
TopicBrowser browser;
Topic        test_topic;
browser = ((AQJMSession) jms_session).createBrowser(
    test_topic, "SUBS1", Employee.getFactory());
```

TopicBrowserの作成: Oracleオブジェクト型メッセージ、メッセージのロック

```
public oracle.jms.TopicBrowser createBrowser (javax.jms.Topic topic,
                                             java.lang.String cons_name,
                                             java.lang.String messageSelector,
                                             java.lang.Object payload_factory,
                                             boolean locked)
    throws JMSEException
```

このメソッドは、Oracleオブジェクト型メッセージ・トピックに対して、ブラウズ中にメッセージをロックするTopicBrowserを作成しま

す。次のパラメータがあります。

パラメータ	説明
topic	アクセスするトピック。
cons_name	永続サブスクライバまたはコンシューマの名前。
messageSelector	messageSelector 式と一致したプロパティを持つメッセージのみが配信されます。NULL 値または空の文字列は、メッセージ・コンシューマに対する messageSelector がないことを表します。
payload_factory	Oracle のユーザー定義型にマップする Java クラスの CustomDatumFactory または ORADDataFactory
locked	TRUE に設定すると、ブラウズ中のメッセージはロックされます(UPDATE の SELECT と同様)。

注意:



- CustomDatum は今後のリリースではサポートされなくなります。かわりに、ORADDataFactory ペイロード・ファクトリを使用してください。
- シャード・キューは、オブジェクト型メッセージをサポートしていません。

関連項目:

[「MessageSelector」](#)

例14-29 TopicBrowserの作成: AdtMessageメッセージ、メッセージのロック

```
/* Create a browser for a Topic with AdtMessage messages of type EMPLOYEE* in
lock mode/
TopicSession jms_session
TopicBrowser browser;
Topic        test_topic;
browser = ((AQJmsSession) jms_session).createBrowser(
        test_topic, "SUBS1", Employee.getFactory(), true);
```

TopicBrowserを使用したメッセージのブラウズ

```
public void purgeSeen()
        throws JMSEException
```

このメソッドは、TopicBrowserを使用してメッセージをブラウズします。java.util.Enumeration内のメソッドを使用して、メッセージのリストを参照してください。現行のブラウズ中に参照したメッセージを削除するには、TopicBrowser内のpurgeSeenメ

ソッドを使用してください。

例14-30 TopicBrowserの作成: セレクタの指定あり

```
/* Create a browser for topics with a specified selector */
public void browse_rush_orders(TopicSession jms_session)
TopicBrowser    browser;
Topic            topic;
ObjectMessage   obj_message;
BolOrder        new_order;
Enumeration      messages;
/* get a handle to the new_orders topic */
topic = ((AQJmsSession) jms_session).getTopic("OE", "OE_bookedorders_topic");
/* create a Browser to look at RUSH orders */
browser = ((AQJmsSession) jms_session).createBrowser(
    topic, "SUBS1", "JMSCorrelationID = 'RUSH'");
/* Browse through the messages */
for (messages = browser.elements() ; message.hasMoreElements() ;)
{obj_message = (ObjectMessage)message.nextElement();}
/* Purge messages seen during this browse */
browser.purgeSeen();
```

15 Oracle Java Message Serviceの共有インタフェース

次のトピックでは、Oracle Database Advanced Queuing (AQ)に対する[Java Message Service \(JMS\)操作インタフェース](#)(共有インタフェース)を説明します。

- [Oracle Database Advanced Queuing JMS操作インタフェース: 共有インタフェース](#)
- [JMSメッセージ・プロパティの指定](#)
- [MessageProducerが送信するすべてのメッセージに対するデフォルトのTimeToLiveの設定](#)
- [MessageProducerが送信するすべてのメッセージに対するデフォルトの優先順位の設定](#)
- [AQjmsエージェントの作成](#)
- [メッセージの同期受信](#)
- [メッセージの受信に対するナビゲーション・モードの指定](#)
- [メッセージの非同期受信](#)
- [メッセージIDの取得](#)
- [JMSメッセージ・プロパティの取得](#)
- [クローズおよびシャットダウン](#)
- [トラブルシューティング](#)

Oracle Database Advanced Queuing JMS操作インタフェース: 共有インタフェース

次のトピックでは、JMS操作のためのOracle Database Advanced Queuing共有インタフェースを説明します。

- [JMSコネクションの起動](#)
- [JMSコネクションの取得](#)
- [セッションにおけるすべての操作のコミット](#)
- [セッションにおけるすべての操作のロールバック](#)
- [セッションからのJDBCコネクションの取得](#)
- [JMSコネクションからのOracleOCIConnectionPoolの取得](#)
- [BytesMessageの作成](#)
- [MapMessageの作成](#)
- [StreamMessageの作成](#)
- [ObjectMessageの作成](#)
- [TextMessageの作成](#)
- [JMSメッセージの作成](#)
- [AdtMessageの作成](#)
- [JMS関連識別子の設定](#)

JMSコネクションの起動

```
public void start()
    throws JMSEException
```

AQjmsConnection.start()は、メッセージ受信用の[JMSコネクション](#)を起動します。

JMSコネクションの取得

```
public oracle.jms.AQjmsConnection getJmsConnection()
    throws JMSEException
```

AQjmsSession.getJmsConnection()は、セッションからJMSコネクションを取得します。

セッションにおけるすべての操作のコミット

```
public void commit()
    throws JMSEException
```

AQjmsSession.commit()は、セッションで実行されているすべてのJMSおよびSQL操作をコミットします。

セッションにおけるすべての操作のロールバック

```
public void rollback()
    throws JMSEException
```

AQjmsSession.rollback()は、セッションで実行されているすべてのJMSおよびSQL操作を終了します。

セッションからのJDBCコネクションの取得

```
public java.sql.Connection getDBConnection()
    throws JMSEException
```

AQjmsSession.getDBConnection()は、[JMSセッション](#)から基礎となるJDBCコネクションを取得します。JDBCコネクションは、JMS操作が実行される同一のトランザクションの一部としてSQL操作を実行するために使用される場合があります。

例15-1 JMSセッションからの基礎となるJDBCコネクションの取得

```
java.sql.Connection db_conn;
QueueSession      jms_sess;
db_conn = ((AQjmsSession) jms_sess).getDBConnection();
```

JMSコネクションからのOracleOCIConnectionPoolの取得

```
public oracle.jdbc.pool.OracleOCIConnectionPool getOCIConnectionPool()
```

AQjmsConnection.getOCIConnectionPool()は、JMSコネクションから基礎となるOracleOCIConnectionPoolを取得します。OracleOCIConnectionPoolインスタンスの設定は、コネクションの使用状況(ユーザーが既存のコネクションを使用して確立するセッション数など)に応じてユーザーがチューニングできます。ただし、JMSコネクションが使用しているOracleOCIConnectionPoolインスタンスはクローズしないでください。

例15-2 JMSコネクションからの基礎となるOracleOCIConnectionPoolの取得

```
oracle.jdbc.pool.OracleOCIConnectionPool cpool;
QueueConnection jms_conn;
cpool = ((AQjmsConnection) jms_conn).getOCIConnectionPool();
```

BytesMessageの作成

```
public javax.jms.BytesMessage createBytesMessage()  
    throws JMSEException
```

AQjmsSession.createBytesMessage() は、Bytesメッセージを作成します。これを使用できるのは、宛先キュー/トピックを含むキュー表がペイロード型SYS.AQ\$_JMS_BYTE_MESSAGEまたはAQ\$_JMS_MESSAGEで作成された場合のみです。

MapMessageの作成

```
public javax.jms.MapMessage createMapMessage()  
    throws JMSEException
```

AQjmsSession.createMapMessage() は、Mapメッセージを作成します。これを使用できるのは、宛先キュー/トピックを含むキュー表がペイロード型SYS.AQ\$_JMS_MAP_MESSAGEまたはAQ\$_JMS_MESSAGEで作成された場合のみです。

StreamMessageの作成

```
public javax.jms.StreamMessage createStreamMessage()  
    throws JMSEException
```

AQjmsSession.createStreamMessage() は、StreamMessageを作成します。これを使用できるのは、宛先キュー/トピックを含むキュー表がペイロード型SYS.AQ\$_JMS_STREAM_MESSAGEまたはAQ\$_JMS_MESSAGEで作成された場合のみです。

ObjectMessageの作成

```
public javax.jms.ObjectMessage createObjectMessage(java.io.Serializable object)  
    throws JMSEException
```

AQjmsSession.createObjectMessage() は、Objectメッセージを作成します。これを使用できるのは、宛先キュー/トピックを含むキュー表がペイロード型SYS.AQ\$_JMS_OBJECT_MESSAGEまたはAQ\$_JMS_MESSAGEで作成された場合のみです。

TextMessageの作成

```
public javax.jms.TextMessage createTextMessage()  
    throws JMSEException
```

AQjmsSession.createTextMessage() は、Textメッセージを作成します。これを使用できるのは、宛先キュー/トピックを含むキュー表がペイロード型SYS.AQ\$_JMS_TEXT_MESSAGEまたはAQ\$_JMS_MESSAGEで作成された場合のみです。

JMSメッセージの作成

```
public javax.jms.Message createMessage()  
    throws JMSEException
```

AQjmsSession.createMessage() は、JMSメッセージを作成します。AQ\$_JMS_MESSAGEメッセージを使用して、様々な型のメッセージを構成できます。メッセージ型は、次のいずれかである必要があります。

- DBMS_AQ.JMS_TEXT_MESSAGE
- DBMS_AQ.JMS_OBJECT_MESSAGE
- DBMS_AQ.JMS_MAP_MESSAGE
- DBMS_AQ.JMS_BYTES_MESSAGE

- DBMS_AQ.JMS_STREAM_MESSAGE

このユーザー定義型を使用して、ヘッダーのみのJMSメッセージを作成することもできます。

AdtMessageの作成

```
public oracle.jms.AdtMessage createAdtMessage()  
    throws JMSEException
```

AQjmsSession.createAdtMessage()は、AdtMessageを作成します。これを使用できるのは、宛先キュー/トピックを含むキュー表がペイロード型Oracle ADTで作成された場合のみです。AdtMessageには、CustomDatumインタフェースを実装するオブジェクトが移入される必要があります。このオブジェクトは、キュー/トピックのペイロードとして定義されたSQL ADTのJavaマッピングである必要があります。SQL ADT型に対応するJavaクラスは、Jpublisherツールを使用して生成できます。

JMS関連識別子の設定

```
public void setJMSCorrelationID(java.lang.String correlationID)  
    throws JMSEException
```

AQjmsMessage.setJMSCorrelationID()は、メッセージの関連識別子を指定します。

JMSメッセージ・プロパティの指定

JMSで始まるプロパティ名は、プロバイダ固有です。ユーザー定義のプロパティは、JMSで始めることができません。

次のプロバイダのプロパティは、Text、Stream、Object、BytesまたはMapMessageを使用するクライアントが設定できます。

- JMSXAppID (string)
- JMSXGroupID (string)
- JMSXGroupSeq (int)
- JMS_OracleExcpQ (string)

このメッセージ・プロパティでは[例外キュー](#)を指定します。

- JMS_OracleDelay (int)

このメッセージ・プロパティではメッセージの遅延秒数を指定します。

次のプロパティは、AdtMessageに対して設定できます。

- JMS_OracleExcpQ (String)

このメッセージ・プロパティでは、例外キューを *schema.queue_name* として指定します。

- JMS_OracleDelay (int)

このメッセージ・プロパティではメッセージの遅延秒数を指定します。

この項の内容は次のとおりです。

- [Booleanメッセージ・プロパティの設定](#)
- [Stringメッセージ・プロパティの設定](#)
- [Integerメッセージ・プロパティの設定](#)
- [Doubleメッセージ・プロパティの設定](#)
- [Floatメッセージ・プロパティの設定](#)

- [Byteメッセージ・プロパティの設定](#)
- [Longメッセージ・プロパティの設定](#)
- [Shortメッセージ・プロパティの設定](#)
- [Objectメッセージ・プロパティの取得](#)

Booleanメッセージ・プロパティの設定

```
public void setBooleanProperty(java.lang.String name,
                               boolean value)
                               throws JMSEException
```

AQjmsMessage.setBooleanProperty() は、メッセージ・プロパティをBooleanとして指定します。次のパラメータがあります。

パラメータ	説明
name	Boolean プロパティの名前
value	メッセージに設定する Boolean プロパティの値

Stringメッセージ・プロパティの設定

```
public void setStringProperty(java.lang.String name,
                              java.lang.String value)
                              throws JMSEException
```

AQjmsMessage.setStringProperty() は、メッセージ・プロパティをStringとして指定します。次のパラメータがあります。

パラメータ	説明
name	String プロパティの名前
value	メッセージに設定する String プロパティの値

Integerメッセージ・プロパティの設定

```
public void setIntProperty(java.lang.String name,
                           int value)
                           throws JMSEException
```

AQjmsMessage.setIntProperty() は、メッセージ・プロパティをIntegerとして指定します。次のパラメータがあります。

パラメータ	説明
name	Integer プロパティの名前
value	メッセージに設定する Integer プロパティの値

Doubleメッセージ・プロパティの設定

```
public void setDoubleProperty(java.lang.String name,  
                             double value)  
    throws JMSEException
```

AQjmsMessage.setDoubleProperty() は、メッセージ・プロパティをDoubleとして指定します。次のパラメータがあります。

パラメータ	説明
name	Double プロパティの名前
value	メッセージに設定する Double プロパティの値

Floatメッセージ・プロパティの設定

```
public void setFloatProperty(java.lang.String name,  
                             float value)  
    throws JMSEException
```

AQjmsMessage.setFloatProperty() は、メッセージ・プロパティをFloatとして指定します。次のパラメータがあります。

パラメータ	説明
name	Float プロパティの名前
value	メッセージに設定する Float プロパティの値

Byteメッセージ・プロパティの設定

```
public void setByteProperty(java.lang.String name,  
                            byte value)  
    throws JMSEException
```

AQjmsMessage.setByteProperty() は、メッセージ・プロパティをByteとして指定します。次のパラメータがあります。

パラメータ	説明
name	Byte プロパティの名前
value	メッセージに設定する Byte プロパティの値

Longメッセージ・プロパティの設定

```
public void setLongProperty(java.lang.String name,  
                            long value)  
    throws JMSEException
```

AQjmsMessage.setLongProperty() は、メッセージ・プロパティをLongとして指定します。次のパラメータがあります。

パラメータ	説明
name	Long プロパティの名前
value	メッセージに設定する Long プロパティの値

Shortメッセージ・プロパティの設定

```
public void setShortProperty(java.lang.String name,
                             short value)
    throws JMSEException
```

AQjmsMessage.setShortProperty() は、メッセージ・プロパティを Short として指定します。次のパラメータがあります。

パラメータ	説明
name	Short プロパティの名前
value	メッセージに設定する Short プロパティの値

Objectメッセージ・プロパティの設定

```
public void setObjectProperty(java.lang.String name,
                               java.lang.Object value)
    throws JMSEException
```

AQjmsMessage.setObjectProperty() は、メッセージ・プロパティを Object として指定します。オブジェクト化されたプリミティブ値 (Boolean、Byte、Short、Integer、Long、Float、Double および String) のみサポートされています。次のパラメータがあります。

パラメータ	説明
name	Java の Object プロパティの名前
value	メッセージに設定する Java の Object プロパティの値

MessageProducerが送信するすべてのメッセージに対するデフォルトの TimeToLiveの設定

```
public void setTimeToLive(long timeToLive)
    throws JMSEException
```

このメソッドは、MessageProducerが送信するすべてのメッセージに対するデフォルトの TimeToLive を設定します。メッセージの遅延が発生した後に計算されます。このメソッドには、次のパラメータがあります。

パラメータ	説明
-------	----

パラメータ	説明
timeToLive	ミリ秒で指定されるメッセージの Time-To-Live(ゼロは無制限)

例15-3 MessageProducerが送信するすべてのメッセージに対するデフォルトのTimeToLiveの設定

```
/* Set default timeToLive value to 100000 milliseconds for all messages sent by the QueueSender*/
QueueSender sender;
sender.setTimeToLive(100000);
```

MessageProducerが送信するすべてのメッセージに対するデフォルトの優先順位の設定

```
public void setPriority(int priority)
    throws JMSEException
```

このメソッドは、MessageProducerが送信するすべてのメッセージに対するデフォルトのPriorityを設定します。次のパラメータがあります。

パラメータ	説明
priority	この MessageProducer のメッセージの優先順位。デフォルトは 4 です。

優先順位には、どの整数値でも指定できます。数値が小さいほど高い優先度を示します。優先順位がsend()操作時に明示的に指定されている場合は、このメソッドで設定されたデフォルト値をオーバーライドします。

例15-4 QueueSenderが送信するすべてのメッセージに対するデフォルトの優先順位の設定

```
/* Set default priority value to 2 for all messages sent by the QueueSender*/
QueueSender sender;
sender.setPriority(2);
```

例15-5 TopicPublisherが送信するすべてのメッセージに対するデフォルトの優先順位の設定

```
/* Set default priority value to 2 for all messages sent by the TopicPublisher*/
TopicPublisher publisher;
publisher.setPriority(1);
```

AQjmsエージェントの作成

```
public void createAQAgent(java.lang.String agent_name,
    boolean enable_http,
    throws JMSEException
```

このメソッドはAQjmsAgentを作成します。次のパラメータがあります。

パラメータ	説明
agent_name	AQ エージェントの名前。
enable_http	TRUE に設定すると、このエージェントは HTTP 経由で AQ にアクセスできます。

メッセージの同期受信

タイムアウトを指定するか、待機なしで、メッセージを同期受信できます。また、変換を使用してメッセージを受信することもできます。

- [タイムアウト指定でのメッセージ・コンシューマの使用](#)
- [待機なしでのメッセージ・コンシューマの使用](#)
- [変換による宛先からのメッセージの受信](#)

タイムアウト指定でのメッセージ・コンシューマの使用

```
public javax.jms.Message receive(long timeout)
    throws JMSException
```

このメソッドは、タイムアウトを指定し、メッセージ・[コンシューマ](#)を使用してメッセージを受信します。

パラメータ	説明
timeout	タイムアウト値(ミリ秒単位)

例15-6 タイムアウト指定でのメッセージ・コンシューマの使用

```
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession t_sess = null;
TopicSession jms_sess;
Topic shipped_orders;
int myport = 5521;

/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "WS", "Shipped_Orders_Topic");

/* create a subscriber, specifying the correct CustomDatumFactory and
selector */
subscriber1 = jms_sess.createDurableSubscriber(
    shipped_orders, 'WesternShipping',
    " priority > 1 and tab.user_data.region like 'WESTERN %'",
    false, AQjmsAgent.getFactory());
/* receive, blocking for 30 seconds if there were no messages */
Message = subscriber.receive(30000);
```

例15-7 JMS: メッセージが届くまでブロック

```
public BolOrder get_new_order1(QueueSession jms_session)
{
    Queue queue;
    QueueReceiver qrec;
    ObjectMessage obj_message;
    BolCustomer customer;
    BolOrder new_order = null;
    String state;
```

```

try
{
/* get a handle to the new_orders queue */
queue = ((AQjmsSession) jms_session).getQueue("OE", "OE_neworders_que");
qrec = jms_session.createReceiver(queue);

/* wait for a message to show up in the queue */
obj_message = (ObjectMessage)qrec.receive();
new_order = (BolOrder)obj_message.getObject();
customer = new_order.getCustomer();
state = customer.getState();
System.out.println("Order: for customer " + customer.getName());
}
catch (JMSEException ex)
{
System.out.println("Exception: " + ex);
}
return new_order;
}

```

待機なしでのメッセージ・コンシューマの使用

```

public javax.jms.Message receiveNoWait()
    throws JMSEException

```

このメソッドは、待機なしで、メッセージ・コンシューマを使用してメッセージを受信します。

例15-8 JMS: メッセージを非ブロック

```

public BolOrder poll_new_order3(QueueSession jms_session)
{
Queue queue;
QueueReceiver qrec;
ObjectMessage obj_message;
BolCustomer customer;
BolOrder new_order = null;
String state;

try
{
/* get a handle to the new_orders queue */
queue = ((AQjmsSession) jms_session).getQueue("OE", "OE_neworders_que");
qrec = jms_session.createReceiver(queue);

/* check for a message to show up in the queue */
obj_message = (ObjectMessage)qrec.receiveNoWait();
new_order = (BolOrder)obj_message.getObject();
customer = new_order.getCustomer();
state = customer.getState();

System.out.println("Order: for customer " + customer.getName());
}
catch (JMSEException ex)
{
System.out.println("Exception: " + ex);
}
return new_order;
}

```

変換による宛先からのメッセージの受信

キューまたはトピックからメッセージを受信するときに変換を適用できます。メッセージを変換してから、JMSアプリケーションに戻します。

変換は、AQjmsQueueReceiver、AQjmsTopicSubscriberまたはAQjmsTopicReceiverのsetTransformation() インタフェースを使用して指定できます。

例15-9 JMS: 変換による宛先からのメッセージの受信

Western Shippingアプリケーションが、OE_bookedorders_topicからメッセージを取得すると仮定します。これは、変換OE2WSがメッセージをOracleオブジェクト型WS_orderとして取得するよう指定します。WSOrder JavaクラスがJPublisherによって生成され、Oracleオブジェクト型WS_orderにマップされると仮定します。

```
public AQjmsAdtMessage retrieve_bookedorders(TopicSession jms_session)
    AQjmsTopicReceiver receiver;
    Topic topic;
    Message msg = null;

    try
    {
        /* get a handle to the OE_bookedorders_topic */
        topic = ((AQjmsSession) jms_session).getTopic("OE", "OE_bookedorders_topic");

        /* Create a receiver for WShip */
        receiver = ((AQjmsSession) jms_session).createTopicReceiver(
            topic, "WShip", null, WSOrder.getFactory());

        /* set the transformation in the publisher */
        receiver.setTransformation("OE2WS");
        msg = receiver.receive(10);
    }
    catch (JMSEException ex)
    {
        System.out.println("Exception :", ex);
    }
    return (AQjmsAdtMessage)msg;
}
```

メッセージの受信に対するナビゲーション・モードの指定

```
public void setNavigationMode(int mode)
    throws JMSEException
```

このメソッドは、メッセージの受信に対するナビゲーション・モードを指定します。次のパラメータがあります。

パラメータ	説明
mode	ナビゲーション・モードの新規値

例15-10 メッセージの受信に対するナビゲーション・モードの指定

```
TopicConnectionFactory tc_fact = null;
TopicConnection t_conn = null;
TopicSession t_sess = null;
TopicSession jms_sess;
Topic shipped_orders;
```

```

int                myport = 5521;

/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic("WS", "Shipped_Orders_Topic");

/* create a subscriber, specifying the correct CustomDatumFactory and selector */
subscriber1 = jms_sess.createDurableSubscriber(
    shipped_orders, 'WesternShipping',
    "priority > 1 and tab.user_data.region like 'WESTERN %'", false,
    AQjmsAgent.getFactory());
subscriber1.setNavigationMode(AQjmsConstants.NAVIGATION_FIRST_MESSAGE);

/* get message for the subscriber, returning immediately if there was no message */
Message = subscriber.receive();

```

メッセージの非同期受信

メッセージを非同期に受信するには、次の2つの方法があります。

- [メッセージ・コンシューマに対するメッセージ・リスナーの指定](#)
- [セッションに対するメッセージ・リスナーの指定](#)

メッセージ・コンシューマに対するメッセージ・リスナーの指定

```

public void setMessageListener(javax.jms.MessageListener myListener)
    throws JMSEException

```

このメソッドは、メッセージ・コンシューマに対してメッセージ・リスナーを指定します。次のパラメータがあります。

パラメータ	説明
myListener	コンシューマに対してメッセージ・リスナーを設定します。

例15-11 メッセージ・コンシューマに対するメッセージ・リスナーの指定

```

TopicConnectionFactory tc_fact = null;
TopicConnection        t_conn  = null;
TopicSession           t_sess  = null;
TopicSession           jms_sess;
Topic                  shipped_orders;
int                    myport = 5521;
MessageListener        mLis = null;

/* create connection and session */
tc_fact = AQjmsFactory.getTopicConnectionFactory(
    "MYHOSTNAME", "MYSID", myport, "oci8");
t_conn = tc_fact.createTopicConnection("jmstopic", "jmstopic");
jms_sess = t_conn.createTopicSession(true, Session.CLIENT_ACKNOWLEDGE);
shipped_orders = ((AQjmsSession) jms_sess).getTopic(
    "WS", "Shipped_Orders_Topic");

/* create a subscriber, specifying the correct CustomDatumFactory and selector */
subscriber1 = jms_sess.createDurableSubscriber(

```

```

    shipped_orders, 'WesternShipping',
    "priority > 1 and tab.user_data.region like 'WESTERN %'",
    false, AQjmsAgent.getFactory());
mLis = new myListener(jms_sess, "foo");

/* get message for the subscriber, returning immediately if there was no message */
subscriber.setMessageListener(mLis);
The definition of the myListener class
import oracle.AQ.*;
import oracle.jms.*;
import javax.jms.*;
import java.lang.*;
import java.util.*;
public class myListener implements MessageListener
{
    TopicSession    mySess;
    String          myName;
    /* constructor */
    myListener(TopicSession t_sess, String t_name)
    {
        mySess = t_sess;
        myName = t_name;
    }
    public onMessage(Message m)
    {
        System.out.println("Retrieved message with correlation: " ||
m.getJMSCorrelationID());
        try{
            /* commit the dequeue */
            mySession.commit();
        } catch (java.sql.SQLException e)
        {System.out.println("SQL Exception on commit"); }
    }
}

```

セッションに対するメッセージ・リスナーの指定

```

public void setMessageListener(javax.jms.MessageListener listener)
    throws JMSEException

```

このメソッドは、セッションに対してメッセージ・リスナーを指定します。

パラメータ	説明
listener	このセッションに関連付けするメッセージ・リスナー

メッセージIDの取得

この項の内容は次のとおりです。

- [相関識別子の取得](#)
- [メッセージ識別子の取得](#)

相関識別子の取得

```

public java.lang.String getJMSCorrelationID()

```


throws JMSEException

AQjmsMessage.getJMSCorrelationID() は、メッセージの相関識別子を取得します。

メッセージ識別子の取得

```
public byte[] getJMSCorrelationIDAsBytes()
    throws JMSEException
```

AQjmsMessage.getJMSMessageID() は、メッセージのメッセージ識別子をバイトまたは文字列として取得します。

JMSメッセージ・プロパティの取得

この項の内容は次のとおりです。

- [Booleanメッセージ・プロパティの取得](#)
- [Stringメッセージ・プロパティの取得](#)
- [Integerメッセージ・プロパティの取得](#)
- [Doubleメッセージ・プロパティの取得](#)
- [Floatメッセージ・プロパティの取得](#)
- [Byteメッセージ・プロパティの取得](#)
- [Longメッセージ・プロパティの取得](#)
- [Shortメッセージ・プロパティの取得](#)
- [Objectメッセージ・プロパティの取得](#)

Booleanメッセージ・プロパティの取得

```
public boolean getBooleanProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getBooleanProperty() は、メッセージ・プロパティをBooleanとして取得します。次のパラメータがあります。

パラメータ	説明
name	Boolean プロパティの名前

Stringメッセージ・プロパティの取得

```
public java.lang.String getStringProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getStringProperty() は、メッセージ・プロパティをStringとして取得します。次のパラメータがあります。

パラメータ	説明
name	String プロパティの名前

Integerメッセージ・プロパティの取得

```
public int getIntProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getIntProperty() は、メッセージ・プロパティをIntegerとして取得します。次のパラメータがあります。

パラメータ	説明
name	Integer プロパティの名前

Doubleメッセージ・プロパティの取得

```
public double getDoubleProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getDoubleProperty() は、メッセージ・プロパティをDoubleとして取得します。次のパラメータがあります。

パラメータ	説明
name	Double プロパティの名前

Floatメッセージ・プロパティの取得

```
public float getFloatProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getFloatProperty() は、メッセージ・プロパティをFloatとして取得します。次のパラメータがあります。

パラメータ	説明
name	Float プロパティの名前

Byteメッセージ・プロパティの取得

```
public byte getByteProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getByteProperty() は、メッセージ・プロパティをByteとして取得します。次のパラメータがあります。

パラメータ	説明
name	Byte プロパティの名前

Longメッセージ・プロパティの取得

```
public long getLongProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getLongProperty() は、メッセージ・プロパティをLongとして取得します。次のパラメータがあります。

パラメータ	説明
name	Long プロパティの名前

Shortメッセージ・プロパティの取得

```
public short getShortProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getShortProperty()は、メッセージ・プロパティをShortとして取得します。次のパラメータがあります。

パラメータ	説明
name	Short プロパティの名前

Objectメッセージ・プロパティの取得

```
public java.lang.Object getObjectProperty(java.lang.String name)
    throws JMSEException
```

AQjmsMessage.getObjectProperty()は、メッセージ・プロパティをObjectとして取得します。次のパラメータがあります。

パラメータ	説明
name	Object プロパティの名前

例15-12 Objectメッセージ・プロパティの取得

```
TextMessage message;
message.getObjectProperty("empid", new Integer(1000));
```

クローズおよびシャットダウン

この項の内容は次のとおりです。

- [MessageProducerのクローズ](#)
- [メッセージ・コンシューマのクローズ](#)
- [JMSコネクションの停止](#)
- [JMSセッションのクローズ](#)
- [JMSコネクションのクローズ](#)

MessageProducerのクローズ

```
public void close()
    throws JMSEException
```

AQjmsProducer.close()はMessageProducerをクローズします。

メッセージ・コンシューマのクローズ

```
public void close()  
    throws JMSException
```

`AQjmsConsumer.close()` は、メッセージ・コンシューマをクローズします。

JMSコネクションの停止

```
public void stop()  
    throws JMSException
```

`AQjmsConnection.stop()` はJMSコネクションを停止します。

JMSセッションのクローズ

```
public void close()  
    throws JMSException
```

`AQjmsSession.close()` はJMSセッションをクローズします。

JMSコネクションのクローズ

```
public void close()  
    throws JMSException
```

`AQjmsConnection.close()` はJMSコネクションをクローズし、コネクションのために割り当てられているすべてのリソースを解放します。通常、JMSのプロバイダは重要なリソースをコネクションのためにJVMの外部に割り当てるため、クライアントはこれらのリソースが必要ない場合はクローズする必要があります。ガベージ・コレクションによる最終的なリソースの解放を待つ必要はありません。

トラブルシューティング

この項の内容は次のとおりです。

- [JMSエラー・コードの取得](#)
- [JMSエラー番号の取得](#)
- [JMS例外にリンクされた例外の取得](#)
- [JMS例外のスタック・トレースの出力](#)
- [例外リスナーの設定](#)
- [例外リスナーの取得](#)

JMSエラー・コードの取得

```
public java.lang.String getErrorCode()
```

`AQjmsException.getErrorCode()` は、JMS例外のエラー・コードを取得します。

JMSエラー番号の取得

```
public int getErrorNumber()
```

AQjmsException.getErrorNumber() は、JMS例外のエラー番号を取得します。



注意:

このメソッドは今後のリリースでは使用されなくなります。かわりに、getErrorCode() を使用してください。

JMS例外にリンクされた例外の取得

```
public java.lang.String getLinkString()
```

AQjmsException.getLinkString() は、JMS例外にリンクされた例外を取得します。一般に、この例外にはデータベースで発行されるSQL例外が含まれます。

JMS例外のスタック・トレースの出力

```
public void printStackTrace(java.io.PrintStream s)
```

AQjmsException.printStackTrace() は、JMS例外のスタック・トレースを出力します。

例外リスナーの設定

```
public void setExceptionListener(javax.jms.ExceptionListener listener)
    throws JMSEException
```

AQjmsConnection.setExceptionListener() は、コネクション用の例外リスナーを指定します。次のパラメータがあります。

パラメータ	説明
listener	例外リスナー

例外リスナーの登録が完了している場合は、コネクションに重大な問題が検出されると通知されます。これは、リスナーのonException() メソッドをコールして、問題を説明するJMS例外を渡すことによって実行されます。これによって、問題がJMSクライアントに非同期に通知されます。メッセージを処理するのみのコネクションもあるため、コネクションが失敗したことを知るための他の方法がありません。

例15-13 コネクション用の例外リスナーの指定

```
//register an exception listener
Connection jms_connection;
jms_connection.setExceptionListener(
    new ExceptionListener() {
        public void onException (JMSEException jmsException) {
            System.out.println("JMS-EXCEPTION: " + jmsException.toString());
        }
    });
);
```

例外リスナーの取得

```
public javax.jms.ExceptionListener getExceptionListener()
    throws JMSEException
```

AQjmsConnection. getExceptionListener () は、コネクション用の例外リスナーを取得します。

例15-14に、ExceptionListenerとMessageListenerの組合せ使用の方法を示します。次の条件が満たされていることを確認してください。

- ユーザー jmsuser (パスワード jmsuser) が、適切な権限を所有してデータベースに作成されます。
- キュー demoQueue が作成され起動されます。

この例は、コネクション再起動があり、例外リスナーがJMSオブジェクトを再作成する場合に、MessageListenerが非同期にメッセージを受信する方法を示します。

例15-14 ExceptionListenerとMessageListenerの組合せ使用

```
import java.util.Enumeration;
import java.util.Properties;

import javax.jms.Connection;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageConsumer;
import javax.jms.MessageProducer;
import javax.jms.Queue;
import javax.jms.QueueBrowser;
import javax.jms.Session;
import javax.jms.TextMessage;

import oracle.jms.AQjmsConnectionFactory;
import oracle.jms.AQjmsFactory;
import oracle.jms.AQjmsSession;

public class JMSDemo {

    static String queueName = "demoQueue";

    static String queueOwner = "jmsuser";

    static String queueOwnerPassword = "jmsuser";

    static Connection connection = null;

    static int numberOfMessages = 25000;

    static int messageCount = 0;

    static String jdbcURL = "";

    public static void main(String args[]) {
        try {
            jdbcURL = System.getProperty("JDBC_URL");

            if (jdbcURL == null)
                System.out
                    .println("The system property JDBC_URL has not been set, " +
                        "usage: java -DJDBC_URL=xxx filename ");
        }
        else {
            JMSDemo demo = new JMSDemo();
            demo.performJmsOperations();
        }
    }
}
```

```

    }
} catch (Exception exception) {
    System.out.println("Exception : " + exception);
    exception.printStackTrace();
} finally {
    try {
        if (connection != null)
            connection.close();
    } catch (Exception exc) {
        exc.printStackTrace();
    }
}
}
System.out.println("\nEnd of Demo aqjmsdemo11.");
}

public void performJmsOperations() {
    try {
        connection = getConnection(jdbcURL);
        Session session = connection.createSession(false,
            Session.AUTO_ACKNOWLEDGE);
        Queue queue = session.createQueue(queueName);

        // remove the messages from the Queue
        drainQueue(queueName, queueOwner, jdbcURL, true);

        // set the exception listener on the Connection
        connection.setExceptionListener(new DemoExceptionListener());

        MessageProducer producer = session.createProducer(queue);
        TextMessage textMessage = null;

        System.out.println("Sending " + numberOfMessages + " messages to queue "
            + queueName);
        for (int i = 0; i < numberOfMessages; i++) {
            textMessage = session.createTextMessage();
            textMessage.setText("Sample message text");
            producer.send(textMessage);
        }

        MessageConsumer consumer = session.createConsumer(queue);
        System.out.println("Setting the message listener ...");
        consumer.setMessageListener(new DemoMessageListener());
        connection.start();

        // Introduce a long wait to allow the listener to receive all the messages
        while (messageCount < numberOfMessages) {
            try {
                Thread.sleep(5000);
            } catch (InterruptedException interruptedException) {
            }
        }
    } catch (JMSEException jmsException) {
        jmsException.printStackTrace();
    }
}

// Sample message listener
static class DemoMessageListener implements javax.jms.MessageListener {

    public void onMessage(Message message) {

```

```

    try {
        System.out.println("Message listener received message with JMSMessageID "
            + message.getJMSMessageID());
        messageCount++;
    } catch (JMSEException jmsException) {
        System.out.println("JMSEException " + jmsException.getMessage());
    }
}

// sample exception listener
static class DemoExceptionListener implements javax.jms.ExceptionListener {

    public void onException(JMSEException jmsException) {
        try {
            // As a first step close the connection
            if (connection != null)
                connection.close();
        } catch (JMSEException exception) {}

        try {
            System.out.println("Re-create the necessary JMS objects ...");
            connection = getConnection(jdbcURL);
            connection.start();
            Session session = connection.createSession(false,
                Session.AUTO_ACKNOWLEDGE);
            Queue queue = session.createQueue(queueName);
            MessageConsumer consumer = session.createConsumer(queue);
            consumer.setMessageListener(new DemoMessageListener());
        } catch (JMSEException newJmsException) {
            newJmsException.printStackTrace();
        }
    }
}

// Utility method to get a connection
static Connection getConnection(String jdbcUrl) throws JMSEException {
    Properties prop = new Properties();
    prop.put("user", queueOwner);
    prop.put("password", queueOwnerPassword);

    AQjmsConnectionFactory fact = (AQjmsConnectionFactory) AQjmsFactory
        .getConnectionFactory(jdbcUrl, prop);
    Connection conn = fact.createConnection();
    return conn;
}

// Utility method to remove the messages from the queue
static void drainQueue(String queueName, String queueOwner, String jdbcUrl,
    boolean debugInfo) {
    Connection connection = null;
    Session session = null;
    long timeout = 10000;
    int count = 0;
    Message message = null;
    try {
        connection = getConnection(jdbcUrl);
        connection.start();
        session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
        Queue queue = ((AQjmsSession) session).getQueue(queueOwner, queueName);

```


16 Oracle Java Message Service型の例

次の例は、Oracle JMSタイプを使用してOracle Database Advanced Queuing (AQ)メッセージを[デキュー](#)および[エンキュー](#)する方法を示しています。

- [Oracle Database Advanced QueuingのJMS型の設定方法の例](#)
- [JMS BytesMessageの例](#)
- [JMS StreamMessageの例](#)
- [JMS MapMessageの例](#)
- [その他のOracle Database Advanced Queuing JMSの例](#)

Oracle Database Advanced QueuingのJMS型の設定方法の例

[例16-3](#)から[例16-8](#)を実行するステップは、次のとおりです。

1. [例16-1](#)をコピーし、setup.sqlとして保存します。
2. 次のようにsetup.sqlを実行します。

```
sqlplus /nolog @setup.sql
```

3. SQL*Plusにjmsuser/jmsuserとしてログインします。
4. [メッセージ](#)の型ごとに、対応するSQLスクリプトのペアを実行します。

たとえば、JMS BytesMessageの場合は[例16-3](#)および[例16-4](#)を実行します。

5. データベース・パラメータjava_pool_sizeの容量が十分であることを確認してください。たとえば、java_pool_size=20Mを使用できます。

例16-1 JMS型の例を実行する環境の設定

```
connect sys;
enter password: password

Rem
Rem Create the JMS user: jmsuser
Rem

DROP USER jmsuser CASCADE;
CREATE USER jmsuser IDENTIFIED BY jmsuser;
GRANT EXECUTE ON DBMS_AQADM TO jmsuser;
GRANT EXECUTE ON DBMS_AQ TO jmsuser;
GRANT EXECUTE ON DBMS_LOB TO jmsuser;
GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;

set echo offset verify offconnect sysDROP USER jmsuser CASCADE:ACCEPT password CHAR PROMPT 'Enter the
password for JMSUSER: ' HIDECREATE USER jmsuser IDENTIFIED BY &password;GRANT DBA,
AQ_ADMINISTRATOR_ROLE, AQ_USER_ROLE to jmsuser;GRANT EXECUTE ON DBMS_AQADM TO jmsuser;GRANT EXECUTE ON
DBMS_AQ TO jmsuser;GRANT EXECUTE ON DBMS_LOB TO jmsuser;GRANT EXECUTE ON DBMS_JMS_PLSQL TO
jmsuser;connect jmsuser/&password

Rem
Rem Creating five AQ queue tables and five queues for five payloads:
Rem SYS.AQ$_JMS_TEXT_MESSAGE
```

```

Rem SYS.AQ$_JMS_BYTES_MESSAGE
Rem SYS.AQ$_JMS_STREAM_MESSAG
Rem SYS.AQ$_JMS_MAP_MESSAGE
Rem SYS.AQ$_JMS_MESSAGE
Rem

EXECUTE DBMS_AQADM.CREATE_QUEUE_TABLE (Queue_table => 'jmsuser.jms_qtt_text',
    Queue_payload_type => 'SYS.AQ$_JMS_TEXT_MESSAGE', compatible => '8.1.0');
EXECUTE DBMS_AQADM.CREATE_QUEUE_TABLE (Queue_table => 'jmsuser.jms_qtt_bytes',
    Queue_payload_type => 'SYS.AQ$_JMS_BYTES_MESSAGE', compatible => '8.1.0');
EXECUTE DBMS_AQADM.CREATE_QUEUE_TABLE (Queue_table => 'jmsuser.jms_qtt_stream',
    Queue_payload_type => 'SYS.AQ$_JMS_STREAM_MESSAGE', compatible => '8.1.0');
EXECUTE DBMS_AQADM.CREATE_QUEUE_TABLE (Queue_table => 'jmsuser.jms_qtt_map',
    Queue_payload_type => 'SYS.AQ$_JMS_MAP_MESSAGE', compatible => '8.1.0');
EXECUTE DBMS_AQADM.CREATE_QUEUE_TABLE (Queue_table => 'jmsuser.jms_qtt_general',
    Queue_payload_type => 'SYS.AQ$_JMS_MESSAGE', compatible => '8.1.0');
EXECUTE DBMS_AQADM.CREATE_QUEUE (Queue_name => 'jmsuser.jms_text_que',
    Queue_table => 'jmsuser.jms_qtt_text');
EXECUTE DBMS_AQADM.CREATE_QUEUE (Queue_name => 'jmsuser.jms_bytes_que',
    Queue_table => 'jmsuser.jms_qtt_bytes');
EXECUTE DBMS_AQADM.CREATE_QUEUE (Queue_name => 'jmsuser.jms_stream_que',
    Queue_table => 'jmsuser.jms_qtt_stream');
EXECUTE DBMS_AQADM.CREATE_QUEUE (Queue_name => 'jmsuser.jms_map_que',
    Queue_table => 'jmsuser.jms_qtt_map');
EXECUTE DBMS_AQADM.CREATE_QUEUE (Queue_name => 'jmsuser.jms_general_que',
    Queue_table => 'jmsuser.jms_qtt_general');

Rem
Rem Starting the queues and enable both enqueue and dequeue
Rem
EXECUTE DBMS_AQADM.START_QUEUE (Queue_name => 'jmsuser.jms_text_que');
EXECUTE DBMS_AQADM.START_QUEUE (Queue_name => 'jmsuser.jms_bytes_que');
EXECUTE DBMS_AQADM.START_QUEUE (Queue_name => 'jmsuser.jms_stream_que');
EXECUTE DBMS_AQADM.START_QUEUE (Queue_name => 'jmsuser.jms_map_que');
EXECUTE DBMS_AQADM.START_QUEUE (Queue_name => 'jmsuser.jms_general_que');

Rem The supporting utility used in the example to help display results in SQLPLUS enviroment

Rem
Rem Display a RAW data in SQLPLUS
Rem
create or replace procedure display_raw(rdata raw)
IS
    pos                pls_integer;
    length             pls_integer;
BEGIN
    pos := 1;
    length := UTL_RAW.LENGTH(rdata);

    WHILE pos <= length LOOP
        IF pos+20 > length+1 THEN
            dbms_output.put_line(UTL_RAW.SUBSTR(rdata, pos, length-pos+1));
        ELSE
            dbms_output.put_line(UTL_RAW.SUBSTR(rdata, pos, 20));
        END IF;
        pos := pos+20;
    END LOOP;

END display_raw;
/

```

```
show errors;
```

```
Rem
Rem Display a BLOB data in SQLPLUS
Rem
create or replace procedure display_blob(bdata blob)
IS
    pos            pls_integer;
    length         pls_integer;
BEGIN
    length := dbms_lob.getlength(bdata);
    pos := 1;
    WHILE pos <= length LOOP
        display_raw(DBMS_LOB.SUBSTR(bdata, 2000, pos));
        pos := pos+2000;
    END LOOP;
END display_blob;
/
```

```
show errors;
```

```
Rem
Rem Display a VARCHAR data in SQLPLUS
Rem
create or replace procedure display_varchar(vdata varchar)
IS
    pos            pls_integer;
    text_len      pls_integer;
BEGIN
    text_len := length(vdata);
    pos := 1;

    WHILE pos <= text_len LOOP
        IF pos+20 > text_len+1 THEN
            dbms_output.put_line(SUBSTR(vdata, pos, text_len-pos+1));
        ELSE
            dbms_output.put_line(SUBSTR(vdata, pos, 20));
        END IF;
        pos := pos+20;
    END LOOP;
END display_varchar;
/
```

```
show errors;
```

```
Rem
Rem Display a CLOB data in SQLPLUS
Rem
create or replace procedure display_clob(cdata clob)
IS
    pos            pls_integer;
    length         pls_integer;
BEGIN
    length := dbms_lob.getlength(cdata);
    pos := 1;
    WHILE pos <= length LOOP
        display_varchar(DBMS_LOB.SUBSTR(cdata, 2000, pos));
        pos := pos+2000;
    END LOOP;
END display_clob;
/
```

```

    END LOOP;
END display_clob;
/

show errors;

Rem
Rem Display a SYS.AQ$_JMS_EXCEPTION data in SQLPLUS
Rem
Rem When application receives an ORA-24197 error, It means the JAVA stored
Rem procedures has thrown some exceptions that could not be catergorized. The
Rem user can use GET_EXCEPTION procedure of SYS.AQ$_JMS_BYTES_MESSAGE,
Rem SYS.AQ$_JMS_STREAM_MESSAG or SYS.AQ$_JMS_MAP_MESSAGE
Rem to retrieve a SYS.AQ$_JMS_EXCEPTION object which contains more detailed
Rem information on this JAVA exception including the exception name, JAVA error
Rem message and stack trace.
Rem
Rem This utility function is to help display the SYS.AQ$_JMS_EXCEPTION object in
Rem SQLPLUS
Rem
create or replace procedure display_exp(exp SYS.AQ$_JMS_EXCEPTION)
IS
    pos1          pls_integer;
    pos2          pls_integer;
    text_data     varchar(2000);
BEGIN
    dbms_output.put_line(' exception:' || exp.exp_name);
    dbms_output.put_line(' err_msg:' || exp.err_msg);
    dbms_output.put_line(' stack:' || length(exp.stack));
    pos1 := 1;
    LOOP
        pos2 := INSTR(exp.stack, chr(10), pos1);
        IF pos2 = 0 THEN
            pos2 := length(exp.stack)+1;
        END IF;

        dbms_output.put_line(SUBSTR(exp.stack, pos1, pos2-pos1));

        IF pos2 > length(exp.stack) THEN
            EXIT;
        END IF;

        pos1 := pos2+1;
    END LOOP;

END display_exp;
/

show errors;

EXIT;

```

例16-2 例の設定

[例16-1](#)では、JMS型の例に必要な設定を実行します。この例をコピーし、setup.sqlとして保存します。

JMS BytesMessageの例

この項では、JMS BytesMessageのエンキューおよびデキューの説明の例を示します。

[例16-3](#)では、JMS型メンバー関数とDBMS_AQ関数を使用して、データベース内でsys.aq\$_jms_bytes_message型として表示されるJMS BytesMessageを移入およびエンキューする方法を紹介します。このメッセージは、後でJAVA [Oracle Java Message Service](#) (Oracle JMS)クライアントによってデキューできます。

[例16-4](#)では、JMS型メンバー関数とDBMS_AQ関数を使用して、データベース内でsys.aq\$_jms_bytes_message型として表示されるJMS BytesMessageからデータをデキューおよび取り出す方法を紹介합니다。このメッセージは、Oracle JMSクライアントによってエンキューされる場合もあります。

例16-3 BytesMessageの移入とエンキュー

```
set echo off set verify off connect sysDROP USER jmsuser CASCADE;ACCEPT password CHAR PROMPT 'Enter the password for JMSUSER: ' HIDECREATE USER jmsuser IDENTIFIED BY &password;GRANT DBA, AQ_ADMINISTRATOR_ROLE, AQ_USER_ROLE to jmsuser;GRANT EXECUTE ON DBMS_AQADM TO jmsuser;GRANT EXECUTE ON DBMS_AQ TO jmsuser;GRANT EXECUTE ON DBMS_LOB TO jmsuser;GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;connect jmsuser/&password
```

```
SET ECHO ON
set serveroutput on
```

```
DECLARE
```

```
    id                pls_integer;
    agent             sys.aq$_agent := sys.aq$_agent(' ', null, 0);
    message           sys.aq$_jms_bytes_message;
    enqueue_options   dbms_aq.enqueue_options_t;
    message_properties dbms_aq.message_properties_t;
    msgid raw(16);
```

```
    java_exp          exception;
    pragma EXCEPTION_INIT(java_exp, -24197);
```

```
BEGIN
```

```
    -- Construct a empty BytesMessage object
    message := sys.aq$_jms_bytes_message.construct;
```

```
    -- Shows how to set the JMS header
    message.set_replyto(agent);
    message.set_type(' tkaqpet1');
    message.set_userid(' jmsuser');
    message.set_appid(' plsql_enq');
    message.set_groupid(' st');
    message.set_groupseq(1);
```

```
    -- Shows how to set JMS user properties
    message.set_string_property(' color', ' RED');
    message.set_int_property(' year', 1999);
    message.set_float_property(' price', 16999.99);
    message.set_long_property(' mileage', 300000);
    message.set_boolean_property(' import', True);
    message.set_byte_property(' password', -127);
```

```
    -- Shows how to populate the message payload of aq$_jms_bytes_message
```

```
    -- Passing -1 reserve a new slot within the message store of sys.aq$_jms_bytes_message.
    -- The maximum number of sys.aq$_jms_bytes_message type of messages to be operated at
    -- the same time within a session is 20. Calling clean_body function with parameter -1
    -- might result a ORA-24199 error if the messages currently operated is already 20.
    -- The user is responsible to call clean or clean_all function to clean up message store.
    id := message.clear_body(-1);
```

```

-- Write data into the BytesMessage payload. These functions are analogy of JMS JAVA api's.
-- See the document for detail.

-- Write a byte to the BytesMessage payload
message.write_byte(id, 10);

-- Write a RAW data as byte array to the BytesMessage payload
message.write_bytes(id, UTL_RAW.XRANGE(HEXTORAW('00'), HEXTORAW('FF')));

-- Write a portion of the RAW data as byte array to BytesMessage payload
-- Note the offset follows JAVA convention, starting from 0
message.write_bytes(id, UTL_RAW.XRANGE(HEXTORAW('00'), HEXTORAW('FF')), 0, 16);

-- Write a char to the BytesMessage payload
message.write_char(id, 'A');

-- Write a double to the BytesMessage payload
message.write_double(id, 9999.99);

-- Write a float to the BytesMessage payload
message.write_float(id, 99.99);

-- Write a int to the BytesMessage payload
message.write_int(id, 12345);

-- Write a long to the BytesMessage payload
message.write_long(id, 1234567);

-- Write a short to the BytesMessage payload
message.write_short(id, 123);

-- Write a String to the BytesMessage payload,
-- the String is encoded in UTF8 in the message payload
message.write_utf(id, 'Hello World!');

-- Flush the data from JAVA stored procedure (JServ) to PL/SQL side
-- Without doing this, the PL/SQL message is still empty.
message.flush(id);

-- Use either clean_all or clean to clean up the message store when the user
-- do not plan to do payload population on this message anymore
sys.aq$_jms_bytes_message.clean_all();
--message.clean(id);

-- Enqueue this message into AQ queue using DBMS_AQ package
dbms_aq.enqueue(queue_name => 'jmsuser.jms_bytes_que',
               enqueue_options => enqueue_options,
               message_properties => message_properties,
               payload => message,
               msgid => msgid);

EXCEPTION
WHEN java_exp THEN
    dbms_output.put_line('exception information:');
    display_exp(sys.aq$_jms_stream_message.get_exception());

```

```

END;
/

```

```
commit;
```

例16-4 JMS BytesMessageデータのデキューおよび取出し

```
set echo off
set verify off

DROP USER jmsuser CASCADE;

ACCEPT password CHAR PROMPT 'Enter the password for JMSUSER: ' HIDE

CREATE USER jmsuser IDENTIFIED BY &password;
GRANT EXECUTE ON DBMS_AQADM TO jmsuser;
GRANT EXECUTE ON DBMS_AQ TO jmsuser;
GRANT EXECUTE ON DBMS_LOB TO jmsuser;
GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;
connect jmsuser/&password
set echo on
set serveroutput on size 20000

DECLARE

    id                pls_integer;
    blob_data         blob;
    clob_data         clob;
    blob_len          pls_integer;
    message           sys.aq$_jms_bytes_message;
    agent             sys.aq$_agent;
    dequeue_options   dbms_aq.dequeue_options_t;
    message_properties dbms_aq.message_properties_t;
    msgid raw(16);
    gdata             sys.aq$_jms_value;

    java_exp          exception;
    pragma EXCEPTION_INIT(java_exp, -24197);
BEGIN
    DBMS_OUTPUT.ENABLE (20000);

    -- Dequeue this message from AQ queue using DBMS_AQ package
    dbms_aq.dequeue(queue_name => 'jmsuser.jms_bytes_que',
                    dequeue_options => dequeue_options,
                    message_properties => message_properties,
                    payload => message,
                    msgid => msgid);

    -- Retrieve the header
    agent := message.get_replyto;

    dbms_output.put_line(' Type: ' || message.get_type ||
                          ' UserId: ' || message.get_userid ||
                          ' AppId: ' || message.get_appid ||
                          ' GroupId: ' || message.get_groupid ||
                          ' GroupSeq: ' || message.get_groupseq);

    -- Retrieve the user properties
    dbms_output.put_line(' price: ' || message.get_float_property('price'));
    dbms_output.put_line(' color: ' || message.get_string_property('color'));
    IF message.get_boolean_property('import') = TRUE THEN
        dbms_output.put_line(' import: Yes' );
    END IF;
END;
```



```

ELSIF message.get_boolean_property('import') = FALSE THEN
    dbms_output.put_line('import: No ');
END IF;
dbms_output.put_line('year: ' || message.get_int_property('year'));
dbms_output.put_line('mileage: ' || message.get_long_property('mileage'));
dbms_output.put_line('password: ' || message.get_byte_property('password'));

-- Shows how to retrieve the message payload of aq$_jms_bytes_message

-- Prepare call, send the content in the PL/SQL aq$_jms_bytes_message object to
-- Java stored procedure(Jserv) in the form of a byte array.
-- Passing -1 reserves a new slot in msg store of sys.aq$_jms_bytes_message.
-- Max number of sys.aq$_jms_bytes_message type of messages to be operated at
-- the same time in a session is 20. Call clean_body fn. with parameter -1
-- might result in ORA-24199 error if messages operated on are already 20.
-- You must call clean or clean_all function to clean up message store.
    id := message.prepare(-1);

-- Read data from BytesMessage payload. These fns. are analogy of JMS Java
-- API's. See the JMS Types chapter for detail.
    dbms_output.put_line('Payload:');

    -- read a byte from the BytesMessage payload
    dbms_output.put_line('read_byte:' || message.read_byte(id));

    -- read a byte array into a blob object from the BytesMessage payload
    dbms_output.put_line('read_bytes:');
    blob_len := message.read_bytes(id, blob_data, 272);
    display_blob(blob_data);

    -- read a char from the BytesMessage payload
    dbms_output.put_line('read_char:' || message.read_char(id));

    -- read a double from the BytesMessage payload
    dbms_output.put_line('read_double:' || message.read_double(id));

    -- read a float from the BytesMessage payload
    dbms_output.put_line('read_float:' || message.read_float(id));

    -- read a int from the BytesMessage payload
    dbms_output.put_line('read_int:' || message.read_int(id));

    -- read a long from the BytesMessage payload
    dbms_output.put_line('read_long:' || message.read_long(id));

    -- read a short from the BytesMessage payload
    dbms_output.put_line('read_short:' || message.read_short(id));

    -- read a String from the BytesMessage payload.
    -- the String is in UTF8 encoding in the message payload
    dbms_output.put_line('read_utf:');
    message.read_utf(id, clob_data);
    display_clob(clob_data);

    -- Use either clean_all or clean to clean up the message store when the user
    -- do not plan to do payload retrieving on this message anymore
    message.clean(id);
    -- sys.aq$_jms_bytes_message.clean_all();

```

EXCEPTION

```

    WHEN java_exp THEN
        dbms_output.put_line('exception information:');
        display_exp(sys.aq$_jms_bytes_message.get_exception());

END;
/

commit;

```

JMS StreamMessageの例

この項では、JMS StreamMessageのエンキューおよびデキューの説明の例を示します。

[例16-5](#)では、JMS型メンバー関数とDBMS_AQ関数を使用して、データベース内でsys.aq\$_jms_stream_message型として表示されるJMS StreamMessageを移入およびエンキューする方法を紹介します。このメッセージは、後でOracle JMSクライアントによってデキューできます。

[例16-6](#)では、JMS型メンバー関数とDBMS_AQ関数を使用して、データベース内でsys.aq\$_jms_stream_message型として表示されるJMS StreamMessageからデータをデキューおよび取り出す方法を紹介しします。このメッセージは、Oracle JMSクライアントによってエンキューされる場合もあります。

例16-5 JMS StreamMessageの移入とエンキュー

```

set echo off
set verify off

DROP USER jmsuser CASCADE;

ACCEPT password CHAR PROMPT 'Enter the password for JMSUSER: ' HIDE

CREATE USER jmsuser IDENTIFIED BY &password;
GRANT EXECUTE ON DBMS_AQADM TO jmsuser;
GRANT EXECUTE ON DBMS_AQ TO jmsuser;
GRANT EXECUTE ON DBMS_LOB TO jmsuser;
GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;
connect jmsuser/&password
SET ECHO ON
set serveroutput on

DECLARE

    id                pls_integer;
    agent             sys.aq$_agent := sys.aq$_agent(' ', null, 0);
    message           sys.aq$_jms_stream_message;
    enqueue_options  dbms_aq.enqueue_options_t;
    message_properties dbms_aq.message_properties_t;
    msgid raw(16);

    java_exp          exception;
    pragma EXCEPTION_INIT(java_exp, -24197);
BEGIN

    -- Construct a empty StreamMessage object
    message := sys.aq$_jms_stream_message.construct;

    -- Shows how to set the JMS header
    message.set_replyto(agent);

```

```

message.set_type('tkaqpet1');
message.set_userid('jmsuser');
message.set_appid('plsql_enq');
message.set_groupid('st');
message.set_groupseq(1);

-- Shows how to set JMS user properties
message.set_string_property('color', 'RED');
message.set_int_property('year', 1999);
message.set_float_property('price', 16999.99);
message.set_long_property('mileage', 300000);
message.set_boolean_property('import', True);
message.set_byte_property('password', -127);

-- Shows how to populate the message payload of aq$_jms_stream_message

-- Passing -1 reserve a new slot within the message store of sys.aq$_jms_stream_message.
-- The maximum number of sys.aq$_jms_stream_message type of messages to be operated at
-- the same time within a session is 20. Calling clean_body function with parameter -1
-- might result a ORA-24199 error if the messages currently operated is already 20.
-- The user is responsible to call clean or clean_all function to clean up message store.
id := message.clear_body(-1);

-- Write data into the message payload. These functions are analogy of JMS JAVA api's.
-- See the document for detail.

-- Write a byte to the StreamMessage payload
message.write_byte(id, 10);

-- Write a RAW data as byte array to the StreamMessage payload
message.write_bytes(id, UTL_RAW.XRANGE(HEXTORAW('00'), HEXTORAW('FF')));

-- Write a portion of the RAW data as byte array to the StreamMessage payload
-- Note the offset follows JAVA convention, starting from 0
message.write_bytes(id, UTL_RAW.XRANGE(HEXTORAW('00'), HEXTORAW('FF')), 0, 16);

-- Write a char to the StreamMessage payload
message.write_char(id, 'A');

-- Write a double to the StreamMessage payload
message.write_double(id, 9999.99);

-- Write a float to the StreamMessage payload
message.write_float(id, 99.99);

-- Write a int to the StreamMessage payload
message.write_int(id, 12345);

-- Write a long to the StreamMessage payload
message.write_long(id, 1234567);

-- Write a short to the StreamMessage payload
message.write_short(id, 123);

-- Write a String to the StreamMessage payload
message.write_string(id, 'Hello World!');

-- Flush the data from JAVA stored procedure (JServ) to PL/SQL side
-- Without doing this, the PL/SQL message is still empty.
message.flush(id);

```

```

-- Use either clean_all or clean to clean up the message store when the user
-- do not plan to do payload population on this message anymore
sys.aq$_jms_stream_message.clean_all();
--message.clean(id);

-- Enqueue this message into AQ queue using DBMS_AQ package
dbms_aq.enqueue(queue_name => 'jmsuser.jms_stream_que',
                enqueue_options => enqueue_options,
                message_properties => message_properties,
                payload => message,
                msgid => msgid);

EXCEPTION
WHEN java_exp THEN
    dbms_output.put_line('exception information:');
    display_exp(sys.aq$_jms_stream_message.get_exception());

END;
/

commit;

```

例16-6 JMS StreamMessageからのデータのデキューおよび取出し

```

set echo off
set verify off

DROP USER jmsuser CASCADE;

ACCEPT password CHAR PROMPT 'Enter the password for JMSUSER: ' HIDE

CREATE USER jmsuser IDENTIFIED BY &password;
GRANT EXECUTE ON DBMS_AQADM TO jmsuser;
GRANT EXECUTE ON DBMS_AQ TO jmsuser;
GRANT EXECUTE ON DBMS_LOB TO jmsuser;
GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;
connect jmsuser/&password
set echo on
set serveroutput on

DECLARE

    id                pls_integer;
    blob_data         blob;
    clob_data         clob;
    message           sys.aq$_jms_stream_message;
    agent             sys.aq$_agent;
    dequeue_options   dbms_aq.dequeue_options_t;
    message_properties dbms_aq.message_properties_t;
    msgid raw(16);
    gdata             sys.aq$_jms_value;

    java_exp          exception;
    pragma EXCEPTION_INIT(java_exp, -24197);

BEGIN
    DBMS_OUTPUT.ENABLE (20000);

```

```

-- Dequeue this message from AQ queue using DBMS_AQ package
dbms_aq.dequeue(queue_name => 'jmsuser.jms_stream_que',
                dequeue_options => dequeue_options,
                message_properties => message_properties,
                payload => message,
                msgid => msgid);

-- Retrieve the header
agent := message.get_replyto;

dbms_output.put_line(' Type: ' || message.get_type ||
                    ' UserId: ' || message.get_userid ||
                    ' AppId: ' || message.get_appid ||
                    ' GroupId: ' || message.get_groupid ||
                    ' GroupSeq: ' || message.get_groupseq);

-- Retrieve the user properties
dbms_output.put_line(' price: ' || message.get_float_property('price'));
dbms_output.put_line(' color: ' || message.get_string_property('color'));
IF message.get_boolean_property('import') = TRUE THEN
    dbms_output.put_line(' import: Yes' );
ELSIF message.get_boolean_property('import') = FALSE THEN
    dbms_output.put_line(' import: No' );
END IF;
dbms_output.put_line(' year: ' || message.get_int_property('year'));
dbms_output.put_line(' mileage: ' || message.get_long_property('mileage'));
dbms_output.put_line(' password: ' || message.get_byte_property('password'));

-- Shows how to retrieve the message payload of aq$_jms_stream_message

-- The prepare call send the content in the PL/SQL aq$_jms_stream_message object to
-- JAVA stored procedure(Jserv) in the form of byte array.
-- Passing -1 reserve a new slot within the message store of sys.aq$_jms_stream_message.
-- The maximum number of sys.aq$_jms_stream_message type of messages to be operated at
-- the same time within a session is 20. Calling clean_body function with parameter -1
-- might result a ORA-24199 error if the messages currently operated is already 20.
-- The user is responsible to call clean or clean_all function to clean up message store.
id := message.prepare(-1);

-- Assume the users know the types of data in the StreamMessage payload.
-- The user can use the specific read function corresponding with the data type.
-- These functions are analogy of JMS JAVA api's. See the document for detail.
dbms_output.put_line(' Retrieve payload by Type:');

-- Read a byte from the StreamMessage payload
dbms_output.put_line(' read_byte:' || message.read_byte(id));

-- Read a byte array into a blob object from the StreamMessage payload
dbms_output.put_line(' read_bytes:');
message.read_bytes(id, blob_data);
display_blob(blob_data);

-- Read another byte array into a blob object from the StreamMessage payload
dbms_output.put_line(' read_bytes:');
message.read_bytes(id, blob_data);
display_blob(blob_data);

-- Read a char from the StreamMessage payload
dbms_output.put_line(' read_char:' || message.read_char(id));

```

```

-- Read a double from the StreamMessage payload
dbms_output.put_line(' read_double:' || message.read_double(id));

-- Read a float from the StreamMessage payload
dbms_output.put_line(' read_float:' || message.read_float(id));

-- Read a int from the StreamMessage payload
dbms_output.put_line(' read_int:' || message.read_int(id));

-- Read a long from the StreamMessage payload
dbms_output.put_line(' read_long:' || message.read_long(id));

-- Read a short from the StreamMessage payload
dbms_output.put_line(' read_short:' || message.read_short(id));

-- Read a String into a clob data from the StreamMessage payload
dbms_output.put_line(' read_string:');
message.read_string(id, clob_data);
display_clob(clob_data);

-- Assume the users do not know the types of data in the StreamMessage payload.
-- The user can use read_object method to read the data into a sys.aq$_jms_value object
-- These functions are analogy of JMS JAVA api's. See the document for detail.

-- Reset the stream pointer to the beginning of the message so that we can read throught
-- the message payload again.
message.reset(id);

```

LOOP

```

message.read_object(id, gdata);
IF gdata IS NULL THEN
    EXIT;
END IF;

```

CASE gdata.type

```

WHEN sys.dbms_jms_plsql.DATA_TYPE_BYTE THEN
    dbms_output.put_line(' read_object/byte:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_SHORT THEN
    dbms_output.put_line(' read_object/short:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_INTEGER THEN
    dbms_output.put_line(' read_object/int:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_LONG THEN
    dbms_output.put_line(' read_object/long:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_FLOAT THEN
    dbms_output.put_line(' read_object/float:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_DOUBLE THEN
    dbms_output.put_line(' read_object/double:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_BOOLEAN THEN
    dbms_output.put_line(' read_object/boolean:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_CHARACTER THEN
    dbms_output.put_line(' read_object/char:' || gdata.char_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_STRING THEN
    dbms_output.put_line(' read_object/string:');
    display_clob(gdata.text_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_BYTES THEN
    dbms_output.put_line(' read_object/bytes:');
    display_blob(gdata.bytes_val);
ELSE dbms_output.put_line(' No such data type');

```

```

END CASE;

END LOOP;

-- Use either clean_all or clean to clean up the message store when the user
-- do not plan to do payload retrieving on this message anymore
message.clean(id);
-- sys.aq$_jms_stream_message.clean_all();

EXCEPTION
WHEN java_exp THEN
    dbms_output.put_line('exception information:');
    display_exp(sys.aq$_jms_stream_message.get_exception());

END;
/

commit;

```

JMS MapMessageの例

この項では、JMS MapMessageのエンキューおよびデキューの説明の例を示します。

[例16-7](#)では、JMS型メンバー関数とDBMS_AQ関数を使用して、データベース内でsys.aq\$_jms_map_message型として表示されるJMS MapMessageを移入およびエンキューする方法を紹介します。このメッセージは、後でOracle JMSクライアントによってデキューできます。

[例16-8](#)では、JMS型メンバー関数とDBMS_AQ関数を使用して、データベース内でsys.aq\$_jms_map_message型として表示されるJMS MapMessageからデータをデキューおよび取り出す方法を紹介します。このメッセージは、Oracle JMSクライアントによってエンキューされる場合もあります。

例16-7 JMS MapMessageの移入とエンキュー

```

set echo off
set verify off

DROP USER jmsuser CASCADE;

ACCEPT password CHAR PROMPT 'Enter the password for JMSUSER: ' HIDE

CREATE USER jmsuser IDENTIFIED BY &password;
GRANT EXECUTE ON DBMS_AQADM TO jmsuser;
GRANT EXECUTE ON DBMS_AQ TO jmsuser;
GRANT EXECUTE ON DBMS_LOB TO jmsuser;
GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;
connect jmsuser/&password

SET ECHO ON
set serveroutput on

DECLARE

    id                pls_integer;
    agent             sys.aq$_agent := sys.aq$_agent(' ', null, 0);
    message           sys.aq$_jms_map_message;
    enqueue_options  dbms_aq.enqueue_options_t;
    message_properties dbms_aq.message_properties_t;

```

```

msgid raw(16);

java_exp          exception;
pragma EXCEPTION_INIT(java_exp, -24197);
BEGIN

-- Construct a empty map message object
message := sys.aq$_jms_map_message.construct;

-- Shows how to set the JMS header
message.set_replyto(agent);
message.set_type('tkaqpet1');
message.set_userid('jmsuser');
message.set_appid('plsql_enq');
message.set_groupid('st');
message.set_groupseq(1);

-- Shows how to set JMS user properties
message.set_string_property('color', 'RED');
message.set_int_property('year', 1999);
message.set_float_property('price', 16999.99);
message.set_long_property('mileage', 300000);
message.set_boolean_property('import', True);
message.set_byte_property('password', -127);

-- Shows how to populate the message payload of aq$_jms_map_message

-- Passing -1 reserve a new slot within the message store of sys.aq$_jms_map_message.
-- The maximum number of sys.aq$_jms_map_message type of messages to be operated at
-- the same time within a session is 20. Calling clean_body function with parameter -1
-- might result a ORA-24199 error if the messages currently operated is already 20.
-- The user is responsible to call clean or clean_all function to clean up message store.
id := message.clear_body(-1);

-- Write data into the message payload. These functions are analogy of JMS JAVA api's.
-- See the document for detail.

-- Set a byte entry in map message payload
message.set_byte(id, 'BYTE', 10);

-- Set a byte array entry using RAW data in map message payload
message.set_bytes(id, 'BYTES', UTL_RAW.XRANGE(HEXTORAW('00'), HEXTORAW('FF')));

-- Set a byte array entry using only a portion of the RAW data in map message payload
-- Note the offset follows JAVA convention, starting from 0
message.set_bytes(id, 'BYTES_PART', UTL_RAW.XRANGE(HEXTORAW('00'), HEXTORAW('FF')), 0, 16);

-- Set a char entry in map message payload
message.set_char(id, 'CHAR', 'A');

-- Set a double entry in map message payload
message.set_double(id, 'DOUBLE', 9999.99);

-- Set a float entry in map message payload
message.set_float(id, 'FLOAT', 99.99);

-- Set a int entry in map message payload
message.set_int(id, 'INT', 12345);

-- Set a long entry in map message payload

```



```

message.set_long(id, 'LONG', 1234567);

-- Set a short entry in map message payload
message.set_short(id, 'SHORT', 123);

-- Set a String entry in map message payload
message.set_string(id, 'STRING', 'Hello World!');

-- Flush the data from JAVA stored procedure (JServ) to PL/SQL side
-- Without doing this, the PL/SQL message is still empty.
message.flush(id);

-- Use either clean_all or clean to clean up the message store when the user
-- do not plan to do payload population on this message anymore
sys.aq$_jms_map_message.clean_all();
--message.clean(id);

-- Enqueue this message into AQ queue using DBMS_AQ package
dbms_aq.enqueue(queue_name => 'jmsuser.jms_map_que',
                enqueue_options => enqueue_options,
                message_properties => message_properties,
                payload => message,
                msgid => msgid);

END;
/

commit;

```

例16-8 JMS MapMessageからのデータのデキューおよび取出し

```

set echo off
set verify off

DROP USER jmsuser CASCADE;

ACCEPT password CHAR PROMPT 'Enter the password for JMSUSER: ' HIDE

CREATE USER jmsuser IDENTIFIED BY &password;
GRANT EXECUTE ON DBMS_AQADM TO jmsuser;
GRANT EXECUTE ON DBMS_AQ TO jmsuser;
GRANT EXECUTE ON DBMS_LOB TO jmsuser;
GRANT EXECUTE ON DBMS_JMS_PLSQL TO jmsuser;
connect jmsuser/&password

set echo on
set serveroutput on

DECLARE

    id                pls_integer;
    blob_data         blob;
    clob_data         clob;
    message           sys.aq$_jms_map_message;
    agent             sys.aq$_agent;
    dequeue_options   dbms_aq.dequeue_options_t;
    message_properties dbms_aq.message_properties_t;
    msgid             raw(16);
    name_arr          sys.aq$_jms_namearray;

```

```

gdata          sys.aq$_jms_value;

java_exp      exception;
pragma EXCEPTION_INIT(java_exp, -24197);
BEGIN
  DBMS_OUTPUT.ENABLE (20000);

  -- Dequeue this message from AQ queue using DBMS_AQ package
  dbms_aq.dequeue(queue_name => 'jmsuser.jms_map_que',
                 dequeue_options => dequeue_options,
                 message_properties => message_properties,
                 payload => message,
                 msgid => msgid);

  -- Retrieve the header
  agent := message.get_replyto;

  dbms_output.put_line(' Type: ' || message.get_type ||
                      ' UserId: ' || message.get_userid ||
                      ' AppId: ' || message.get_appid ||
                      ' GroupId: ' || message.get_groupid ||
                      ' GroupSeq: ' || message.get_groupseq);

  -- Retrieve the user properties
  dbms_output.put_line(' price: ' || message.get_float_property('price'));
  dbms_output.put_line(' color: ' || message.get_string_property('color'));
  IF message.get_boolean_property('import') = TRUE THEN
    dbms_output.put_line(' import: Yes' );
  ELSIF message.get_boolean_property('import') = FALSE THEN
    dbms_output.put_line(' import: No' );
  END IF;
  dbms_output.put_line(' year: ' || message.get_int_property('year'));
  dbms_output.put_line(' mileage: ' || message.get_long_property('mileage'));
  dbms_output.put_line(' password: ' || message.get_byte_property('password'));

  -- Shows how to retrieve the message payload of aq$_jms_map_message

  -- 'Prepare' sends the content in the PL/SQL aq$_jms_map_message object to
  -- Java stored procedure(Jserv) in the form of byte array.
  -- Passing -1 reserve a new slot within the message store of
  -- sys.aq$_jms_map_message. The maximum number of sys.aq$_jms_map_message
  -- type of messages to be operated at the same time within a session is 20.
  -- Calling clean_body function with parameter -1
  -- might result a ORA-24199 error if the messages currently operated is
  -- already 20. The user is responsible to call clean or clean_all function
  -- to clean up message store.
  id := message.prepare(-1);

  -- Assume the users know the names and types in the map message payload.
  -- The user can use names to get the corresponding values.
  -- These functions are analogous to JMS Java API's. See JMS Types chapter
  -- for detail.
  dbms_output.put_line(' Retrieve payload by Name:');

  -- Get a byte entry from the map message payload
  dbms_output.put_line(' get_byte:' || message.get_byte(id, 'BYTE'));

  -- Get a byte array entry from the map message payload
  dbms_output.put_line(' get_bytes:');

```

```

message.get_bytes(id, 'BYTES', blob_data);
display_blob(blob_data);

-- Get another byte array entry from the map message payload
dbms_output.put_line('get_bytes:');
message.get_bytes(id, 'BYTES_PART', blob_data);
display_blob(blob_data);

-- Get a char entry from the map message payload
dbms_output.put_line('get_char:' || message.get_char(id, 'CHAR'));

-- get a double entry from the map message payload
dbms_output.put_line('get_double:' || message.get_double(id, 'DOUBLE'));

-- Get a float entry from the map message payload
dbms_output.put_line('get_float:' || message.get_float(id, 'FLOAT'));

-- Get a int entry from the map message payload
dbms_output.put_line('get_int:' || message.get_int(id, 'INT'));

-- Get a long entry from the map message payload
dbms_output.put_line('get_long:' || message.get_long(id, 'LONG'));

-- Get a short entry from the map message payload
dbms_output.put_line('get_short:' || message.get_short(id, 'SHORT'));

-- Get a String entry from the map message payload
dbms_output.put_line('get_string:');
message.get_string(id, 'STRING', clob_data);
display_clob(clob_data);

-- Assume users do not know names and types in map message payload.
-- User can first retrieve the name array containing all names in the
-- payload and iterate through the name list and get the corresponding
-- value. These functions are analogous to JMS Java API's.
-- See JMS Type chapter for detail.
dbms_output.put_line('Retrieve payload by iteration:');

-- Get the name array from the map message payload
name_arr := message.get_names(id);

-- Iterate through the name array to retrieve the value for each of the name.
FOR i IN name_arr.FIRST..name_arr.LAST LOOP

-- Test if a name exist in the map message payload
-- (It is not necessary in this case, just a demonstration on how to use it)
IF message.item_exists(id, name_arr(i)) THEN
    dbms_output.put_line('item exists:' || name_arr(i));

-- Because we do not know the type of entry, we must use sys.aq$_jms_value
-- type object for the data returned
message.get_object(id, name_arr(i), gdata);
IF gdata IS NOT NULL THEN
    CASE gdata.type
    WHEN sys.dbms_jms_plsql.DATA_TYPE_BYTE
        THEN dbms_output.put_line('get_object/byte:' || gdata.num_val);
    WHEN sys.dbms_jms_plsql.DATA_TYPE_SHORT
        THEN dbms_output.put_line('get_object/short:' || gdata.num_val);
    WHEN sys.dbms_jms_plsql.DATA_TYPE_INTEGER
        THEN dbms_output.put_line('get_object/int:' || gdata.num_val);
    
```

```

WHEN sys.dbms_jms_plsql.DATA_TYPE_LONG
  THEN dbms_output.put_line('get_object/long:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_FLOAT
  THEN dbms_output.put_line('get_object/float:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_DOUBLE
  THEN dbms_output.put_line('get_object/double:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_BOOLEAN
  THEN dbms_output.put_line('get_object/boolean:' || gdata.num_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_CHARACTER
  THEN dbms_output.put_line('get_object/char:' || gdata.char_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_STRING
  THEN dbms_output.put_line('get_object/string:');
  display_clob(gdata.text_val);
WHEN sys.dbms_jms_plsql.DATA_TYPE_BYTES
  THEN
    dbms_output.put_line('get_object/bytes:');
    display_blob(gdata.bytes_val);
  ELSE dbms_output.put_line('No such data type');
  END CASE;
END IF;
ELSE
  dbms_output.put_line('item not exists:' || name_arr(i));
END IF;

END LOOP;

-- Use either clean_all or clean to clean up the message store when the user
-- do not plan to do payload population on this message anymore
message.clean(id);
-- sys.aq$_jms_map_message.clean_all();

EXCEPTION
WHEN java_exp THEN
  dbms_output.put_line('exception information:');
  display_exp(sys.aq$_jms_stream_message.get_exception());

END;
/

commit;

```

その他のOracle Database Advanced Queuing JMSの例

[例16-9](#)のサンプル・プログラムでは、JMS TEXTメッセージを保持するために、Oracle JMS管理インタフェースによって作成されるOracle Streams Advanced Queuingキュー内に、大きなTextMessageを(JMSユーザー・プロパティとともに)エンキューしています。この例でエンキューされるTextMessageおよびBytesMessageは、どちらもOracle JMSクライアントでデキューできます。

[例16-10](#)のサンプル・プログラムでは、大きなBytesMessageをエンキューします。

例16-9 大きなTextMessageのエンキュー

```

DECLARE

  text          varchar2(32767);
  agent         sys.aq$_agent := sys.aq$_agent(' ', null, 0);
  message       sys.aq$_jms_text_message;

```

```

enqueue_options    dbms_aq.enqueue_options_t;
message_properties dbms_aq.message_properties_t;
msgid              raw(16);

```

BEGIN

```

message := sys.aq$_jms_text_message.construct;

message.set_replyto(agent);
message.set_type('tkaqpet2');
message.set_userid('jmsuser');
message.set_appid('plsql_enq');
message.set_groupid('st');
message.set_groupseq(1);

message.set_boolean_property('import', True);
message.set_string_property('color', 'RED');
message.set_short_property('year', 1999);
message.set_long_property('mileage', 300000);
message.set_double_property('price', 16999.99);
message.set_byte_property('password', 127);

FOR i IN 1..500 LOOP
    text := CONCAT (text, '1234567890');
END LOOP;

message.set_text(text);

dbms_aq.enqueue(queue_name => 'jmsuser.jms_text_t1',
                enqueue_options => enqueue_options,
                message_properties => message_properties,
                payload => message,
                msgid => msgid);

```

END;

例16-10 大きなBytesMessageのエンキュー

DECLARE

```

text          VARCHAR2(32767);
bytes         RAW(32767);
agent         sys.aq$_agent := sys.aq$_agent(' ', null, 0);
message       sys.aq$_jms_bytes_message;
body          BLOB;
position      INT;

```

```

enqueue_options    dbms_aq.enqueue_options_t;
message_properties dbms_aq.message_properties_t;
msgid raw(16);

```

BEGIN

```

message := sys.aq$_jms_bytes_message.construct;

message.set_replyto(agent);
message.set_type('tkaqper4');
message.set_userid('jmsuser');
message.set_appid('plsql_enq_raw');
message.set_groupid('st');

```

```

message.set_groupseq(1);

message.set_boolean_property('import', True);
message.set_string_property('color', 'RED');
message.set_short_property('year', 1999);
message.set_long_property('mileage', 300000);
message.set_double_property('price', 16999.99);

-- prepare a huge payload into a blob

FOR i IN 1..1000 LOOP
    text := CONCAT (text, '0123456789ABCDEF');
END LOOP;

bytes := HEXTORAW(text);

dbms_lob.createtemporary(lob_loc => body, cache => TRUE);
dbms_lob.open (body, DBMS_LOB.LOB_READWRITE);
position := 1;
FOR i IN 1..10 LOOP
    dbms_lob.write ( lob_loc => body,
                    amount => FLOOR((LENGTH(bytes)+1)/2),
                    offset => position,
                    buffer => bytes);
    position := position + FLOOR((LENGTH(bytes)+1)/2);
END LOOP;

-- end of the preparation

message.set_bytes(body);
dbms_aq.enqueue(queue_name => 'jmsuser.jms_bytes_t1',
               enqueue_options => enqueue_options,
               message_properties => message_properties,
               payload => message,
               msgid => msgid);

dbms_lob.freetemporary(lob_loc => body);
END;
```

17 Oracle Messaging Gatewayの概要

メッセージ・ゲートウェイ管理パッケージDBMS_MGWADMでは、メッセージ・ゲートウェイ・エージェントの作成、エージェントの管理、メッセージ・システム・リンクの作成、Oracle以外のキューの登録および伝播ジョブの設定のためのインターフェースを提供しています。

トピック:

- [Oracle Messaging Gatewayの概要](#)
- [Oracle Messaging Gatewayの機能](#)
- [Oracle Messaging Gatewayのアーキテクチャ](#)
- [伝播処理の概要](#)
- [Oracle Database AQのバッファ済メッセージとメッセージ・ゲートウェイ](#)

Oracle Messaging Gatewayの概要

メッセージ・ゲートウェイを使用すると、Oracle以外のメッセージ・システムとOracle Database Advanced Queuingに基づくアプリケーション間の通信が可能になります。

Oracle Database Advanced Queuingでは、2つのOracle Database Advanced Queuingキュー間の[伝播](#)機能が提供されており、E-Business([IDAP](#)を介したHTTP)が可能です。メッセージ・ゲートウェイでは、この機能をOracle以外のメッセージ・システムに基づくアプリケーションに拡張しています。

メッセージ・ゲートウェイはOracle Database Advanced QueuingおよびOracle Databaseと統合されているため、信頼性のある[メッセージ](#)配信が可能です。メッセージ・ゲートウェイを使用すると、永続性がサポートされているOracle以外のメッセージ・システムとOracle Database Advanced Queuing間で、メッセージの配信が1回のみ行われることが保証されます。PL/SQLインターフェースでは、特にOracle Database Advanced Queuingの使用方法をすでによく理解している開発者を対象に、簡単に使用できる管理[API](#)が提供されます。

今回のリリースのメッセージ・ゲートウェイでは、Oracle Database Advanced QueuingとWebSphere MQ 9.0およびTIB/Rendezvous 8.2ベースのアプリケーションの統合がサポートされています。

Oracle Messaging Gatewayの機能

メッセージ・ゲートウェイの機能は次のとおりです。

- Oracle Database Advanced Queuingメッセージ伝播の拡張
メッセージ・ゲートウェイでは、Oracle Database Advanced QueuingとOracle以外のメッセージ・システム間でメッセージが伝播されます。Oracle Database Advanced Queuingアプリケーションによって送信されたメッセージは、Oracle以外のメッセージ・システム・アプリケーションで受信できます。逆に、Oracle以外のメッセージ・システム・アプリケーションによってパブリッシュされたメッセージは、Oracle Database Advanced Queuingアプリケーションで使用できます。
- [Java Message Service](#)(JMS)メッセージ・システムに対するサポート
メッセージ・ゲートウェイでは、Oracle Java Message Service(Oracle JMS)およびWebSphere MQ Java Message Service(WebSphere MQ JMS)間でメッセージを伝播します。
- システム固有のメッセージ・フォーマットのサポート

メッセージ・ゲートウェイでは、メッセージ・システム固有のメッセージ・フォーマットがサポートされています。Oracle Database Advanced Queuingメッセージは、RAWまたは任意のOracle [オブジェクト型](#)のペイロードを持てます。WebSphere MQメッセージには、テキスト・メッセージまたはバイト・メッセージが有効です。TIB/Rendezvousメッセージは、ネスト化されたデータ型MSGおよび符号のない整数以外の任意のTIB/Rendezvousワイヤ・フォーマットのデータ型にできます。

- **メッセージの変換**

メッセージ・ゲートウェイを使用すると、Oracle Database Advanced QueuingメッセージとOracle以外のメッセージ・システムのメッセージ間でメッセージを簡単に変換できます。メッセージは、メッセージ・ゲートウェイによって提供される自動メッセージ変換ルーチンまたはユーザーによって提供されたメッセージ [変換](#)関数のいずれかを介して変換されます。

注意:



- メッセージ・ゲートウェイでは、JMS メッセージ・システムと JMS 以外のメッセージ・システム間のメッセージ伝播はサポートされていません。
- Oracle Database AQ シャード・キューは、Oracle Database 12c リリース 2 (12.2)のMGW ではサポートされません。

- **Oracle Databaseとの統合**

メッセージ・ゲートウェイは、Oracle Database Advanced Queuingと類似したPL/SQLインタフェースによって管理されています。構成情報は、Oracle Database表に格納されています。メッセージの伝播は、Oracle Databaseサーバーの外部プロセスによって実行されます。

- **メッセージ配信の保証**

トランザクションが伝播元のメッセージ・システムと伝播先のメッセージ・システムの両方でサポートされている場合、メッセージ・ゲートウェイでは永続メッセージが1回のみ伝播されることが保証されます。メッセージが永続メッセージではないか、伝播元または伝播先のメッセージ・システムでトランザクションがサポートされていない場合は、伝播が行われることが保証されるのみです。

- **セキュリティ・サポート**

メッセージ・ゲートウェイでは、Oracle DatabaseおよびOracle以外のメッセージ・システムのクライアント認証がサポートされています。

メッセージ・ゲートウェイでは、メッセージ・ゲートウェイ・エージェントによるIBM WebSphere MQおよびWebSphere MQ JMS接続においてSecure Socket Layer(SSL)もサポートされています。

- **複数エージェントのサポート**

メッセージ・ゲートウェイでは、1つのデータベースに対して複数のエージェントがサポートされています。伝播ジョブを機能、編成または作業負荷別にパーティション化し、異なるメッセージ・ゲートウェイ・エージェントに割り当てることができます。これにより、メッセージ・ゲートウェイがOracle RAC環境で拡張され、伝播ジョブのグループ化および分離が可能になります。

関連項目:

- [伝播処理の概要](#)
- [Oracle Messaging Gatewayのメッセージの変換](#)
- [Oracle Messaging GatewayのJMS以外のメッセージの変換](#)

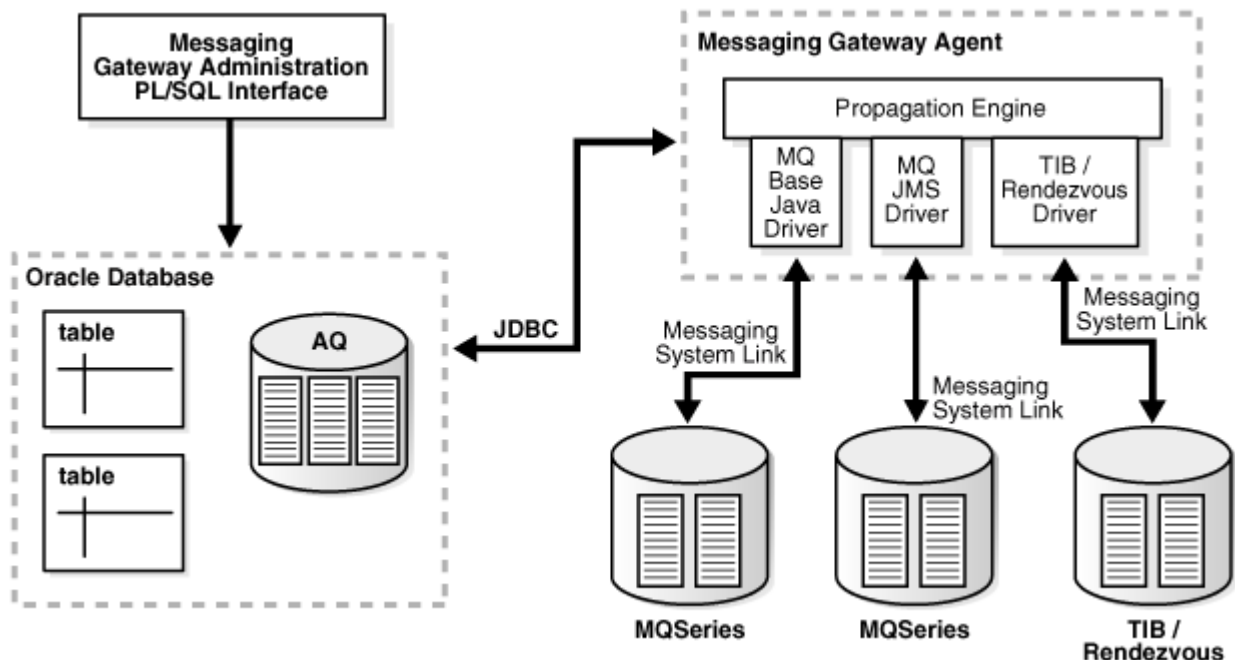
Oracle Messaging Gatewayのアーキテクチャ

メッセージ・ゲートウェイには、次の2つの主要コンポーネントがあります。

- 管理パッケージDBMS_MGWADM
- メッセージ・ゲートウェイ・エージェント

[図17-1](#)は、これらのコンポーネントがOracle DatabaseとOracle以外のメッセージ・システム間で連動する方法を示したものです。

図17-1 メッセージ・ゲートウェイのアーキテクチャ



管理パッケージDBMS_MGWADM

メッセージ・ゲートウェイ管理パッケージDBMS_MGWADMでは、指定されたメッセージ・ゲートウェイ・エージェントの作成、エージェントの管理、メッセージ・システム・リンクの作成、Oracle以外のキューの登録および伝播ジョブの設定のためのインターフェースを提供しています。

メッセージ・ゲートウェイ・エージェントが稼働中かどうかにかかわらず、構成を変更するには、ユーザーはパッケージのプロシージャをコールします。メッセージ・ゲートウェイ・エージェントが稼働中の場合は、パッケージ内のプロシージャが構成の変更に関する通知をエージェントに送信します。大部分の構成変更では、エージェントは動的に構成を変更しますが、一部、変更を反映するために、エージェントを停止して再起動する必要がある場合があります。パッケージ内のすべてのプロシージャはシリアライズされており、メッセージ・ゲートウェイ・エージェントが通知を作成時と同じ順序で受信し、処理できることが保証されています。

関連項目:

DBMS_MGWADMパッケージの詳細は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)を参照し

てください。

Oracle Messaging Gatewayのエージェント

メッセージ・ゲートウェイ・エージェントは、Oracle Databaseサーバーの外部プロセスとして実行され、伝播ジョブを処理します。起動および停止するには、DBMS_MGWADMパッケージのSTARTUPおよびSHUTDOWNプロシジャをそれぞれコールします。

メッセージ・ゲートウェイ・エージェントには、マルチスレッド伝播エンジンおよびメッセージ・システム用のドライバ・セットが含まれます。伝播エンジンによって伝播ジョブが適切にスケジューリングされ、伝播処理が同時に処理されます。エージェントのポーリング・スレッドによって、有効な伝播ジョブのソース・キューが定期的にポーリングされ、メッセージが使用可能な場合は、ワーカー・スレッドが起動して伝播ジョブが処理されます。Oracle以外のメッセージ・システムのドライバは、すべてのメッセージ操作においてメッセージ・システムのクライアントとして実行されます。

Oracle Database

メッセージ・ゲートウェイでは、Oracle Databaseとして、Oracle Database Advanced QueuingとOracle以外のメッセージ・システム間のメッセージ伝播用メカニズムを提供しています。Oracle Database Advanced Queuingは、伝播元または伝播先のいずれかの立場ですべての伝播ジョブに関与しています。

メッセージ・ゲートウェイは、PL/SQL管理パッケージDBMS_MGWADMを介して管理されています。すべての構成情報とメッセージ・ゲートウェイの実行状態情報はOracle Databaseに格納され、データベース・ビューからアクセスできます。

メッセージ・ゲートウェイ・エージェントは、Oracle Databaseサーバーの外部プロシジャとして実行されます。したがって、関連付けられたデータベース・サーバーが実行中の場合のみ実行されます。

Oracle以外のメッセージ・システム

メッセージ・ゲートウェイ・エージェントは、メッセージ・システム・リンクを介してOracle以外のメッセージ・システムに接続されます。メッセージ・システム・リンクとは、メッセージ・ゲートウェイ・エージェントとOracle以外のメッセージ・システム間の通信チャネルです。ユーザーは、管理パッケージDBMS_MGWADMを使用して、同じまたは異なるOracle以外のメッセージ・システムへの複数のリンクを構成できます。

WebSphere MQキュー、TIB/Rendezvousサブジェクト、WebSphere MQ JMS宛先(キューおよびトピック)などのOracle以外のメッセージ・システム内のキューは、すべてメッセージ・ゲートウェイの伝播元および伝播先として機能できます。これらのキューを外部キューと呼びます。ソース・キュー、宛先キューまたは例外キューとしてメッセージ伝播に関与するすべての外部キューは、管理パッケージを経由して登録する必要があります。外部[キュー](#)を登録しても、Oracle以外のメッセージ・システムに物理的なキューは作成されず、キューにアクセスするためのメッセージ・システム・リンク、システム固有の名前、ドメイン(キューまたはトピック)など、キューに関する情報が記録されるだけです。物理的なキューは、Oracle以外のメッセージ・システムの管理インタフェースを介して作成する必要があります。

関連項目:

[Oracle以外のキューの登録](#)

伝播処理の概要

メッセージが1つのメッセージ・システムから別のメッセージ・システムに伝播されるようにするには、伝播ジョブを定義する必要があります。

ります。伝播ジョブは、ソース・キュー、宛先キュー、および伝播ジョブの処理に影響する他の各種属性を定義します。

伝播ソースがPoint-to-Pointキューの場合は、メッセージ・ゲートウェイ・エージェントによってキュー内のすべてのメッセージが宛先に移動されます。伝播ソースがトピック([パブリッシュ・サブスクライブ](#))の場合は、メッセージ・ゲートウェイ・エージェントによって伝播ソース・トピックのサブスクリプションが作成されます。サブスクリプションを作成すると、エージェントによってパブリッシュされたすべてのメッセージがトピックに移動されます。

伝播ジョブはスケジュールが有効になったときに処理されます。伝播ジョブを無効にすると伝播処理が停止されますが、メッセージ・サブスクリプションは停止されません。

メッセージ・ゲートウェイ・エージェントが伝播ジョブを処理すると、ソース・キューからメッセージがデキューされ、宛先キューへメッセージがエンキューされます。メッセージは、各メッセージの伝播時に、ソース・メッセージ・システム固有のフォーマットから宛先メッセージ・システム固有のフォーマットに変換されます。メッセージ・ゲートウェイでは、一般的に使用されている単純なメッセージ・フォーマット間のメッセージ変換は自動的に実行されます。独自のメッセージ変換機能を提供することによって、メッセージ変換をカスタマイズできます。

メッセージ・ゲートウェイ・エージェントがソース・フォーマットから宛先フォーマットへのメッセージ変換に失敗すると、[例外キュー](#)がある場合は、エージェントによってメッセージがソース・キューから例外キューに移動され、伝播ジョブの処理が続行されます。

伝播ジョブの処理中に障害が発生した場合、メッセージ・ゲートウェイ・エージェントはジョブを停止する前に、指数バックオフ・スキームで最大16回ジョブを再試行します(2秒から最大30分)。

信頼できるメッセージ配信を保証するために、メッセージ・ゲートウェイではトランザクションと永続メッセージをサポートしているメッセージ・システムでロギング・キューが必要になります。メッセージ・ゲートウェイ・エージェントでは、ロギング・キューを使用して伝播ジョブの処理状態を格納し、障害から伝播処理を復元できるようにしています。

関連項目:

[Oracle Messaging Gatewayの伝播ジョブの構成](#)

Oracle Database AQのバッファ済メッセージとメッセージ・ゲートウェイ

メッセージ・ゲートウェイでは、バッファ済メッセージの伝播はサポートされていません。発信伝播の場合、メッセージ・ゲートウェイ・エージェントはAQキューから永続メッセージのみをデキューします。着信伝播の場合、メッセージ・ゲートウェイ・エージェントは常に永続メッセージをAQキューにエンキューします。

18 Oracle Messaging Gateway スタート・ガイド

次のトピックでは、Oracle Messaging Gateway (MGW)の前提条件と、メッセージ・ゲートウェイのロード、設定およびアンロード手順を説明します。また、mgw.ora初期化ファイルの設定および変更方法も説明します。

- [Oracle Messaging Gatewayの前提条件](#)
- [Oracle Messaging Gatewayのロードおよび設定](#)
- [Oracle以外のメッセージ・システムの設定](#)
- [Oracle Messaging Gateway設定の検証](#)
- [Oracle Messaging Gatewayのアンロード](#)
- [mgw.ora初期化ファイルについて](#)

Oracle Messaging Gatewayの前提条件

メッセージ・ゲートウェイでは、メッセージ・ゲートウェイ・エージェントごとに1つのOracle Schedulerジョブを使用します。JOB_QUEUE_PROCESSESデータベース初期化パラメータの値がゼロの場合、Oracle Schedulerジョブは実行されません。値がゼロ以外の場合、事実上、この値は同時に実行できるSchedulerジョブおよびジョブ・キューのジョブの最大数です。

ゼロではない値が設定され、各メッセージ・ゲートウェイ・エージェントの起動をSchedulerジョブが処理するのに十分な大きさであることを確認してください。

Oracle Messaging Gatewayのロードおよび設定

メッセージ・ゲートウェイを実行する前に、次の手順を実行してください。

- [データベースへのデータベース・オブジェクトのロード](#)
- [外部プロシージャのlistener.oraの変更](#)
- [外部プロシージャのtnsnames.oraの変更](#)
- [mgw.ora初期化ファイルの設定](#)
- [Oracle Messaging Gateway管理者ユーザーの作成](#)
- [Oracle Messaging Gatewayのエージェント・ユーザーの作成](#)
- [Oracle Messaging Gateway接続情報の構成](#)
- [Oracle RAC環境におけるOracle Messaging Gatewayの構成](#)

注意:



この設定手順は、Windows の 32 ビットおよび 64 ビット・バージョンと UNIX ベースのオペレーティング・システムの 64 ビット・バージョン固有のもので、タスクは、「Windows オペレーティング・システムのみ」または「Linux Operating System のみ」と表示されている以外は、Windows および UNIX ベースのオペレーティング・システム両方に適用されます。その他のオペレーティング・システムについては、オペレーティング・システム固有のドキュメントを参照してください。

データベースへのデータベース・オブジェクトのロード

SQL*Plusを使用して、ユーザーSYSまたはSYSDBAとして `ORACLE_HOME/mgw/admin/catmgw.sql` を実行します。このスクリプトによって、ロール、表、ビュー、オブジェクト型およびPL/SQLパッケージを含む、メッセージ・ゲートウェイに必要なデータベース・オブジェクトがロードされます。これによって、メッセージ・ゲートウェイのPL/SQLパッケージのパブリック・シノニムが作成されます。MGW_ADMINISTRATOR_ROLEおよびMGW_AGENT_ROLE、という2つのロールが作成され、特定の権限が付与されます。すべてのオブジェクトは、SYSに所有されます。

注意:



CDB 環境では、`ORACLE_HOME/mgw/admin/catmgw.sql` を PDB root として実行します。

外部プロシージャのlistener.oraの変更

この手順は、UNIXベースのオペレーティング・システム専用です。Windowsオペレーティング・システムでは、リスナーについての静的サービス情報は必要ありません。

メッセージ・ゲートウェイのPL/SQLパッケージで外部プロシージャがコールされるように、`listener.ora` を変更する必要があります。

1. 外部プロシージャのデフォルトの[プロセス間通信](#)(IPC)プロトコル・アドレスが設定されていることを確認します。

```
LISTENER = (ADDRESS_LIST=  
(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC)))
```

2. ステップ1のリスナーの静的サービス情報を追加します。リスナーのSID_DESCを設定する必要があります。SID_DESCのうち、[表18-1](#)で説明されているパラメータはメッセージ・ゲートウェイで重要なため、使用環境にあわせて指定する必要があります。

例18-1 リスナーの静的サービス情報の追加

```
# Add a SID_DESC  
SID_LIST_LISTENER= (SID_LIST=  
(SID_DESC =  
  (SID_NAME= mgwextproc)  
  (ENVS=  
    "LD_LIBRARY_PATH=JRE_HOME/lib/amd64:JRE_HOME/lib/amd64/server:ORACLE_HOME/lib")  
  (ORACLE_HOME=ORACLE_HOME)  
  (PROGRAM = extproc))
```

表18-1 SID_DESCパラメータ

パラメータ	説明
SID_NAME	tnsnames.ora のネット・サービス名で指定されている SID。次の例では、SID_NAME は mgwextproc です。
ENVS	外部プロシージャを実行するために必要な環境変数 LD_LIBRARY_PATH を設定します。

パラメータ	説明
	<p>LD_LIBRARY_PATH には次のパスを含める必要があります。</p> <p><i>JRE_HOME/lib/PLATFORM_TYPE JRE_HOME/lib/PLATFORM_TYPE/server ORACLE_HOME/lib</i></p> <p>サード・パーティのメッセージ・システムに必要な追加ライブラリも含まれる必要があります。「Oracle 以外のメッセージ・システムの設定」を参照してください。</p> <p>ENVS 値は、LD_LIBRARY_PATH_64 (Oracle Solaris on SPARC (64-Bit)および Oracle Solaris on x86-64 (64-Bit)の場合)、LIBPATH (IBM AIX on POWER Systems の場合)、および LD_LIBRARY_PATH (他のプラットフォームの場合)です。</p>
ORACLE_HOME	ユーザーの Oracle ホーム・ディレクトリ。\$ORACLE_HOME を使用しても機能しません。
PROGRAM	外部プロシージャのエージェントの名前(extproc)。

注意:



ORACLE_HOME が Oracle インストールのルート・ディレクトリを表すように、*JRE_HOME* は JRE インストールのルート・ディレクトリを表します。Oracle Database とともにインストールされた JRE の使用をお勧めします。

[例18-1](#)では、SID_NAME mgwextprocをLinux x86用のlistener.oraファイルに追加します。

外部プロシージャのtnsnames.oraの変更

外部プロシージャに対して、tnsnames.oraで、[例18-2](#)で示したように、接続記述子がlistener.oraで構成された情報に一致するネット・サービス名MGW_AGENTを構成します。ネット・サービス名は、MGW_AGENTである必要があります(この値は固定)。KEYの値は、listener.oraでIPCプロトコルに対して指定されたKEYの値に一致する必要があります。SIDの値は、listener.oraでSID_DESCエントリのSID_NAMEに対して指定された値に一致する必要があります。

注意:



sqlnet.ora の names.default_domain パラメータを使用してデフォルト・ドメインを設定した場合は、そのドメインを tnsnames.ora のネット・サービス名 MGW_AGENT に追加する必要があります。たとえば、sqlnet.ora にエントリ names.default_domain=acme.com が含まれる場合、tnsnames.ora のネット・サービス名は MGW_AGENT.acme.com である必要があります。

例18-2 MGW_AGENTの構成

```
MGW_AGENT =
(DESCRIPTION=
  (ADDRESS_LIST= (ADDRESS= (PROTOCOL=IPC) (KEY=EXTPROC)))
  (CONNECT_DATA= (SID=mgwextproc)))
```

mgw.ora初期化ファイルの設定

メッセージ・ゲートウェイのデフォルトの初期化ファイル *ORACLE_HOME*/mgw/admin/mgw.ora は、テキスト・ファイルです。メッセージ・ゲートウェイの外部プロシージャは、その初期化ファイルを使用して初期化パラメータを取得し、メッセージ・ゲートウェイ・エージェントを起動します。*ORACLE_HOME*/mgw/admin/sample_mgw.ora を mgw.ora にコピーし、状況に応じて変更します。

次のプロシージャでは、メッセージ・ゲートウェイのすべてのアプリケーションに必要な環境変数およびその他のパラメータを設定します。

1. **Windowsオペレーティング・システムのみ**: MGW_PRE_PATH変数を設定します。この値は、次のjvm.dllライブラリへのパスです。

```
set MGW_PRE_PATH = JRE_HOME\bin\client
```

この変数は、メッセージ・ゲートウェイ・エージェントのプロセスによって継承されたパスの前に付けられます。

2. CLASSPATHには、少なくとも次のクラスが入っている必要があります。

- JREランタイム・クラス

```
JRE_HOME/lib/rt.jar
```

- Oracle JDBCクラス

```
ORACLE_HOME/jdbc/lib/ojdbc6.jar
```

- Oracle国際化クラス

```
ORACLE_HOME/jlib/orai18n.jar
```

- SQLJランタイム

```
ORACLE_HOME/sqlj/lib/runtime12.jar
```

- [Java Message Service\(JMS\)](#)インタフェース

```
ORACLE_HOME/rdbms/jlib/jmscommon.jar
```

- Oracle JMS実装クラス

```
ORACLE_HOME/rdbms/jlib/aqapi.jar
```

- Javaトランザクション[API](#)

```
ORACLE_HOME/jlib/jta.jar
```

- メッセージ・ゲートウェイがOracle以外のメッセージ・システムにアクセスするために必要な追加のクラス

注意:



*ORACLE_HOME*には、該当する具体的な値を代入します。たとえば、*\$ORACLE_HOME* を使用しても機能しません。

Windows オペレーティング・システムのユーザーは、Windows オペレーティング・システムのパス構文を使用して CLASSPATH を設定する必要があります。

関連項目:

[Oracle以外のメッセージ・システムの設定](#)

Oracle Messaging Gateway管理者ユーザーの作成

メッセージ・ゲートウェイの管理作業を実行するには、[例18-3](#)に示したように、MGW_ADMINISTRATOR_ROLE権限を所有するデータベース・ユーザーを作成する必要があります。

例18-3 メッセージ・ゲートウェイ管理者ユーザーの作成

```
CREATE USER admin_user IDENTIFIED BY admin_password;  
GRANT CREATE SESSION to admin_user;  
GRANT MGW_ADMINISTRATOR_ROLE to admin_user;
```

Oracle Messaging Gatewayのエージェント・ユーザーの作成

メッセージ・ゲートウェイ・エージェントからデータベースへの接続を確立するには、[例18-4](#)に示したように、MGW_AGENT_ROLE権限を所有するデータベース・ユーザーを作成する必要があります。

例18-4 メッセージ・ゲートウェイ・エージェント・ユーザーの作成

```
CREATE USER agent_user IDENTIFIED BY agent_password;  
GRANT CREATE SESSION to agent_user;  
GRANT MGW_AGENT_ROLE to agent_user;
```

Oracle Messaging Gateway接続情報の構成

メッセージ・ゲートウェイ・エージェント・ユーザーの作成後、管理ユーザーとしてDBMS_MGWADM.ALTER_AGENTを使用し、[例18-5](#)に示したように、メッセージ・ゲートウェイ・エージェントによるデータベースへの接続に使用されるユーザー名、パスワードおよびデータベース接続文字列でメッセージ・ゲートウェイを構成します。[\[Oracle Messaging Gatewayのエージェント・ユーザーの作成\]](#)で作成したメッセージ・ゲートウェイのユーザー名とパスワードを使用します。データベース接続文字列パラメータは、tnsnames.oraの新しいサービス名(パフォーマンスを向上させるにはIPCプロトコルを使用)またはNULLのいずれかに設定できます。NULLに設定する場合は、mgw.oraにoracle_sidパラメータを設定する必要があります。

このリリースの場合は、DBMS_MGWADM.ALTER_AGENTをコールするとき、常にデータベース接続文字列パラメータに対してNULL以外の値を指定してください。

例18-5 メッセージ・ゲートウェイの接続情報の構成

```
set echo off  
set verify off  
connect admin_user  
  
ACCEPT password CHAR PROMPT 'Enter the password for AGENT_USER: ' HIDE  
  
EXEC DBMS_MGWADM.ALTER_AGENT(  
    agent_name => 'default_agent',  
    username   => 'agent_user',  
    password   => '&password',  
    database   => 'agent_database');
```


Oracle RAC環境におけるOracle Messaging Gatewayの構成

この項の内容は次のとおりです。

- [メッセージ・ゲートウェイ・エージェント接続に使用する接続情報の構成](#)
- [メッセージ・ゲートウェイ・エージェントに対するOracle RACインスタンスの設定](#)

メッセージ・ゲートウェイ・エージェント接続に使用する接続情報の構成

メッセージ・ゲートウェイ・エージェントによるすべてのデータベース接続先を、メッセージ・ゲートウェイ・エージェントのプロセスが実行中のインスタンスにする必要があります。これにより、Oracle RAC環境で確実に適切なフェイルオーバー動作が得られます。このように接続を構成するには、インスタンスで少し異なる方法でtnsnames. oraファイルを使用させます。各ファイルには同じネット・サービス名のエントリがありますが、接続データはそのtnsnames. oraファイルに関連付けられているインスタンスのみを参照します。共通のネット・サービス名は、DBMS_MGWADM.ALTER_AGENTを使用してメッセージ・ゲートウェイ・エージェントのデータベース接続情報を構成するときに、データベース・パラメータに使用されます。

たとえば、OraDB1およびOraDB2という2つのインスタンスで構成される2インスタンスOracle RAC環境で、ネット・サービス名AGENT_DBが使用される場合、インスタンスOraDB1のtnsnames. oraは次のようになります。

```
AGENT_DB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = myhost1. mycorp. com) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = OraDB10. mycorp. com)
      (INSTANCE_NAME = OraDB1)
    )
  )
```

OraDB2のtnsnames. oraは次のようになります。

```
AGENT_DB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = myhost2. mycorp. com) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = OraDB10. mycorp. com)
      (INSTANCE_NAME = OraDB2)
    )
  )
```

その後、次のコマンドを実行して、メッセージ・ゲートウェイ・エージェント・ユーザーの接続情報を構成します。

```
EXEC DBMS_MGWADM.ALTER_AGENT (
  agent_name => 'default_agent',
  username   => 'agent_user',
  password   => 'agent_password',
  database   => 'agent_db');
```

メッセージ・ゲートウェイ・エージェントに対するOracle RACインスタンスの設定

メッセージ・ゲートウェイは、Oracle Schedulerのデータベース・サービス・サポートを活用して、メッセージ・ゲートウェイ・エージェントの外部プロセスのサービス・アフィニティを指定します。デフォルトでは、メッセージ・ゲートウェイ・エージェントは、すべてのインスタンスにマップされているデフォルトのデータベース・サービスを使用します。メッセージ・ゲートウェイ・エージェントが、特定のグループのデータベース・インスタンスで起動するようにするには、これらのインスタンスにデータベース・サービスを作成し、

DBMS_MGWADM.CREATE_AGENTまたはDBMS_MGWADM.ALTER_AGENTプロシージャのSERVICEパラメータを使用して、メッセージ・ゲートウェイ・エージェントに割り当てる必要があります。DBMS_MGWADM.STARTUPプロシージャによって、Oracle Schedulerジョブの実行時にメッセージ・ゲートウェイ・エージェントの外部プロセスを開始するSchedulerジョブが送られます。Schedulerジョブは、メッセージ・ゲートウェイ・エージェント用に構成されたデータベース・サービスを使用します。

SERVICEパラメータで指定されるデータベース・サービスは、Oracle Schedulerジョブのサービス・アフィニティ(メッセージ・ゲートウェイの外部プロセスのサービス・アフィニティを含む)にのみ使用されます。メッセージ・ゲートウェイ・エージェント・ユーザーによって作成されたデータベース接続には使用されません。それらのJDBCクライアント接続は、DATABASEおよびCONNTYPEパラメータに指定された値に基づきます。

関連項目:

[Oracle RAC上でのOracle Messaging Gatewayのエージェントの実行](#)

Oracle以外のメッセージ・システムの設定

この項の内容は次のとおりです。

- [TIB/Rendezvousの設定](#)
- [WebSphere MQ Base JavaまたはJMSの設定](#)

TIB/Rendezvousの設定

TIB/Rendezvous Javaクライアント・アプリケーションとして実行する場合、メッセージ・ゲートウェイ・エージェントには、メッセージ・ゲートウェイ・エージェントが実行されるコンピュータ上にTIB/Rendezvousソフトウェアをインストールする必要があります。この項では、*TIBRV_HOME*はインストールされたTIB/Rendezvousソフトウェアの場所を参照します。

listener.oraの変更

Linux Operating Systemの場合は、メッセージ・ゲートウェイのエントリのLD_LIBRARY_PATHには、TIB/Rendezvous共有ライブラリ・ファイルにアクセスするためのエージェントの*TIBRV_HOME/lib*を含める必要があります。

関連項目:

[外部プロシージャのlistener.oraの変更](#)

Windowsオペレーティング・システムの場合は、listener.oraを変更する必要はありません。ただし、システム環境変数PATHに*TIBRV_HOME\bin*を含める必要があります。

mgw.oraの変更

MGW_PRE_PATHには、通常*TIBRV_HOME/bin*にあるTIB/Rendezvousライセンス・チケット・ファイル(*tibrv.tkt*)が入っているディレクトリを含める必要があります。

CLASSPATHには、TIB/Rendezvous jarファイル*TIBRV_HOME/lib/tibrvj.jar*を含める必要があります。独自にカスタマイズされたTIB/Rendezvousアドバイザー・メッセージ・コールバックを使用する場合は、コールバック・クラスの場所も含める必要があります。

次のJavaプロパティを設定して、デフォルトの設定を変更します。

- oracle.mgw.tibrv.encoding
- oracle.mgw.tibrv.intraProcAdvSubjects
- oracle.mgw.tibrv.advMsgCallback

関連項目:

[mgw.ora初期化ファイルについて](#)

例18-6 Javaプロパティの設定

```
setJavaProp oracle.mgw.tibrv.encoding=ISO8859_1
setJavaProp oracle.mgw.tibrv.intraProcAdvSubjects=_RV.>
setJavaProp oracle.mgw.tibrv.advMsgCallback=MyadvCallback
```

WebSphere MQ Base JavaまたはJMSの設定

JavaおよびJMSのWebSphere MQクライアントとWebSphere MQクラスは、メッセージ・ゲートウェイ・エージェントが作動するコンピュータ上にインストールする必要があります。この項では、*MQ_HOME*はインストール済クライアントの場所を参照します。Linux Operating Systemの場合は、この場所は常に/opt/mqmです。Windowsオペレーティング・システムの場合は、インストール場所を変更できます。

listener.oraの変更

メッセージ・ゲートウェイによるWebSphere MQへのアクセスには、listener.oraの変更は必要ありません。

mgw.oraの変更

WebSphere MQ Base Java (JMS以外)インタフェースを使用する場合は、[「mgw.ora初期化ファイルの設定」](#)の設定以外に少なくとも次のパスを含むようにCLASSPATHを設定する必要があります。

- *MQ_HOME*/java/lib/com.ibm.mq.jar
- *MQ_HOME*/java/lib/connector.jar

WebSphere MQ JMSインタフェースを使用する場合は、CLASSPATHを設定して、[「mgw.ora初期化ファイルの設定」](#)の設定以外に少なくとも次のパスを含める必要があります。

- *MQ_HOME*/java/lib/com.ibm.mqjms.jar
- *MQ_HOME*/java/lib/com.ibm.mq.jar
- *MQ_HOME*/java/lib/connector.jar

Oracle Messaging Gateway設定の検証

次の手順を実行して、設定を検証します。この手順には、メッセージ・ゲートウェイ・エージェントの単純な起動および停止が含まれます。

1. データベース・リスナーを起動します。

外部プロシージャに対するリスナーおよび通常のデータベース接続に対する他のリスナーを起動します。

2. メッセージ・ゲートウェイ・エージェント・ユーザーに対してデータベース接続文字列をテストします。

sqlplus *agent_user/agent_password@agent_database*を実行します。

正常に終了した場合、メッセージ・ゲートウェイ・エージェントをデータベースに接続できます。

3. **Linux Operating Systemのみ:** 外部プロシージャのコールに使用するネット・サービス・エントリをテストします。

`sqlplus agent_user/agent_password@MGW_AGENT`を実行します。

これは「ORA-28547: サーバーへの接続が失敗しました。Oracle Net管理エラーの可能性あります」を表示して失敗します。これ以外のエラーは、`tnsnames.ora`、`listener.ora`あるいはその両方が正しくないことを示します。

4. `admin_user`として接続し、`DBMS_MGWADM.STARTUP`をコールしてメッセージ・ゲートウェイ・エージェントを起動します。
5. `MGW_GATEWAY`ビューを使用して、`AGENT_STATUS`が`RUNNING`に、`AGENT_PING`が`REACHABLE`に変わるまで待ちます。
6. `admin_user`として接続し、`DBMS_MGWADM.SHUTDOWN`をコールしてメッセージ・ゲートウェイ・エージェントを停止します。
7. `MGW_GATEWAY`ビューを使用して、`AGENT_STATUS`が`NOT_STARTED`に変わるまで待ちます。

Oracle Messaging Gatewayのアンロード

メッセージ・ゲートウェイをアンロードする手順は、次のとおりです。

1. メッセージ・ゲートウェイを停止します。
2. ペイロードがメッセージ・ゲートウェイ標準型(`SYS.MGW_BASIC_MSG_T`など)であるユーザー作成のキューを削除します。
3. `SQL*Plus`を使用して、ユーザー`SYS`または`SYSDBA`として`ORACLE_HOME/mgw/admin/catnomgw.sql`を実行します。
これによって、ロール、表、ビュー、パッケージ、オブジェクト型およびシノニムを含む、メッセージ・ゲートウェイによって使用されるデータベース・オブジェクトが削除されます。
4. `listener.ora`および`tnsnames.ora`で作成したメッセージ・ゲートウェイのエントリを削除します。

mgw.ora初期化ファイルについて

メッセージ・ゲートウェイ・エージェントを起動すると、メッセージ・ゲートウェイによって、テキスト・ファイルから初期化情報が読み込まれます。初期化ファイルには、初期化パラメータ、環境変数およびJavaプロパティを設定するための行が含まれています。各エントリは1行に指定する必要があります。先頭の空白は、すべての場合に切り捨てられます。

メッセージ・ゲートウェイの管理者は、`DBMS_MGWADM.CREATE_AGENT`および`DBMS_MGWADM.ALTER_AGENT`により、メッセージ・ゲートウェイ・エージェントに使用される初期化ファイルを指定できます。初期化ファイルが指定されない場合、デフォルトの初期化ファイルが使用されます。

デフォルトのエージェントのデフォルト初期化ファイルは、`ORACLE_HOME/mgw/admin/mgw.ora`にあります。

指定されたエージェントのデフォルト初期化ファイルは`ORACLE_HOME/mgw/admin/mgw_AGENTNAME.ora`で、`AGENTNAME`は、メッセージ・ゲートウェイ・エージェントの名前を大文字で表記したものです。たとえば、エージェント名が`my_agent`の場合、エージェントのデフォルト初期化ファイルは`ORACLE_HOME/mgw/admin/mgw_MY_AGENT.ora`です。指定されたエージェントのデフォルトの初期化ファイルが検出されなかった場合、`ORACLE_HOME/mgw/admin/mgw.ora`が使用されます。

mgw.oraの初期化パラメータ

通常、初期化パラメータは、「`name=value<NL>`」という書式の行で指定されます。ここで、`name`はパラメータ名、`value`はパラメータの値、`<NL>`は改行を表します。

`log_directory`

用途:

メッセージ・ゲートウェイのログ・ファイルまたはトレース・ファイルが作成されるディレクトリを指定します。

形式:

```
log_directory = value
```

デフォルト:

```
ORACLE_HOME/mgw/log
```

例:

```
log_directory = /private/mgwlog
```

log_level

用途:

メッセージ・ゲートウェイ・エージェントによって記録されるロギングの詳細レベルを指定します。ロギング・レベルは、メッセージ・ゲートウェイ・エージェントの実行中にDBMS_MGWADM.SET_LOG_LEVELをコールすることで動的に変更できます。Oracleでは、常にログレベル0(デフォルト値)を使用することをお勧めします。

形式:

```
log_level = value
```

値は次のとおりです。

0: 基本的なロギングの場合に使用します。これは、DBMS_MGWADM.BASIC_LOGGINGと同じです。

1: 低レベルのトレースの場合に使用します。これは、DBMS_MGWADM.TRACE_LITE_LOGGINGと同じです。

2: 高レベルのトレースの場合に使用します。これは、DBMS_MGWADM.TRACE_HIGH_LOGGINGと同じです。

3: デバッグ・トレースの場合に使用します。これは、DBMS_MGWADM.TRACE_DEBUG_LOGGINGと同じです。

4: 高レベルのデバッグ・トレースの場合に使用します。これは、DBMS_MGWADM.TRACE_DEBUG_HIGH_LOGGINGと同じです。

例:

```
log_level = 0
```

mgw.oraの環境変数

ユーザーは、メッセージ・ゲートウェイの処理環境を直接制御できないため、初期化ファイルを使用して特定の環境変数を設定する必要があります。現在、メッセージ・ゲートウェイ・エージェントによって使用されている環境変数は、CLASSPATH、MGW_PRE_PATHおよびORACLE_SIDです。

CLASSPATHおよびMGW_PRE_PATHなどの環境変数は、メッセージ・ゲートウェイ・エージェントが必要な共有オブジェクトやJavaクラスなどを検出できるように設定されます。環境変数は、「set env_var=value<NL>」または「setenv env_var=value<NL>」という書式の行で指定されます。ここで、env_varは設定する環境変数の名前、valueは環境変数の値、<NL>は改行を表します。

CLASSPATH

用途:

Oracle Database Advanced QueuingとOracle以外のメッセージ・システム間の[伝播](#)のために、メッセージ・ゲートウェイ・エージェントに必要なJavaクラスを検索する[Java Virtual Machine](#)によって使用されます。

形式:

```
set CLASSPATH=value
```

例:

```
set CLASSPATH=ORACLE_HOME/jdbc/lib/ojdbc6.jar:JRE_HOME/lib/rt.jar:  
ORACLE_HOME/sqlj/lib/runtime12.jar:ORACLE_HOME/jlib/orai18n.jar:ORACLE_HOME/rdbms/  
jlib/jmscommon.jar:ORACLE_HOME/rdbms/jlib/aqapi.jar:ORACLE_HOME/jlib/jta.jar:  
/opt/mqm/java/lib/com.ibm.mq.jar:/opt/mqm/java/lib/com.ibm.mqjms.jar:/opt/mqm/java  
/lib/connector.jar
```

MGW_PRE_PATH

用途:

メッセージ・ゲートウェイのプロセスによって継承されたパスの先頭に追加されます。Windowsオペレーティング・システムの場合、この変数を設定してライブラリjvm.dllが存在する位置を指定する必要があります。

形式:

```
set MGW_PRE_PATH=value
```

例:

```
set MGW_PRE_PATH=JRE_HOME\bin\client
```

ORACLE_SID

用途:

メッセージ・ゲートウェイの構成時にサービス名が指定されていない場合に使用できます。

形式:

```
set ORACLE_SID=value
```

例:

```
set ORACLE_SID=my_sid
```

mgw.ora Javaプロパティ

TIB/Rendezvousサブジェクトを使用する場合、メッセージ・ゲートウェイJVMのJavaシステム・プロパティを指定する必要があります。これには、メッセージ・ゲートウェイ初期化ファイルのsetJavaPropパラメータを使用できます。Javaプロパティは、「setJavaProp *prop_name*=*value*<NL>」という書式の行で指定されます。ここで、*prop_name*は設定するJavaプロパティの名前、*value*はJavaプロパティの値、<NL>は改行記号を表します。

oracle.mgw.batch_size

用途:

このJavaプロパティは、1つのトランザクションで伝播されるメッセージの最大数を表します。メッセージ・ゲートウェイのジョブ・オプションMsgBatchSizeが指定されていない場合のデフォルト値として機能します。デフォルトから変更された場合、予期される[メッセージ](#)のサイズとメッセージ・ゲートウェイ・エージェントのメモリーを考慮する必要があります(DBMS_MGWADM.ALTER_AGENTのmax_memoryパラメータを参照)。このJavaプロパティの最小値は1、最大値は100で、デフォルトは30です。

関連項目:

[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_MGWADMに関する項

構文:

```
setJavaProp oracle.mgw.batch_size=va/ue
```

例:

```
setJavaProp oracle.mgw.batch_size=10
```

oracle.mgw.polling_interval

用途:

このパラメータは、伝播元 [キュー](#) の使用可能メッセージのポーリング間の経過時間(1000分の1秒単位)を指定します。メッセージ・ゲートウェイで 사용되는デフォルトのポーリング間隔は5000ミリ秒(5秒)です。このJavaプロパティの最小値は100ミリ秒で、最大値は60秒です。

構文:

```
setJavaProp oracle.mgw.polling_interval=va/ue
```

例:

```
setJavaProp oracle.mgw.polling_interval=1000
```

oracle.mgw.tibrv.encoding

用途:

このパラメータでは、TIB/Rendezvousメッセージ・システム・リンクで使用される文字エンコードを指定します。TIB/Rendezvous制約により、すべての構成済TIB/Rendezvousリンクに許可されているキャラクタ・セットは1つのみです。デフォルトは、ISO 8859-1またはJavaシステム・プロパティ `file.encoding` によって指定されたキャラクタ・セットです。

構文:

```
setJavaProp oracle.mgw.tibrv.encoding=va/ue
```

例:

```
setJavaProp oracle.mgw.tibrv.encoding=ISO8859_1
```

oracle.mgw.tibrv.intraProcAdvSubjects

用途

このパラメータはすべてのTIB/Rendezvousメッセージ・システム・リンクに使用され、プロセス内転送で表示されるシステム勧告サブジェクトの名前を指定します。

構文

```
setJavaProp oracle.mgw.tibrv.intraProcAdvSubjects=  
advisorySubjectName[: advisorySubjectName]
```

例:

```
setJavaProp oracle.mgw.tibrv.intraProcAdvSubjects=_RV.>
```

oracle.mgw.tibrv.advMsgCallback

用途:

このパラメータはすべてのTIB/Rendezvousメッセージ・システム・リンクに使用され、TibrvMsgCallbackインタフェースを実装してシステム勧告メッセージを処理するJavaクラスの名前を指定します。指定されていない場合は、メッセージ・ゲートウェイに

よって提供されるデフォルトのシステム警告メッセージ・ハンドラが使用され、メッセージ・ゲートウェイのログ・ファイルにシステム警告メッセージが書き込まれます。指定されている場合は、クラス・ファイルが格納されているディレクトリがmgw. oraのCLASSPATHに含まれている必要があります。

構文:

```
setJavaProp oracle.mgw.tibrv.advMsgCallback=c/assName
```

例:

```
setJavaProp oracle.mgw.tibrv.advMsgCallback=MyAdvCallback
```

oracle.net.tns_admin

用途:

このパラメータはtnsnames.oraファイルのディレクトリを指定します。メッセージ・ゲートウェイ・エージェントがJDBC Thinドライバを使用するよう構成されており、エージェントの接続情報のデータベース指定子がTNSNamesの別名の場合、設定する必要があります。JDBC OCIドライバが使用されている場合や、データベース指定子がTNSNamesの別名ではない場合は、設定する必要はありません。

構文:

```
setJavaProp oracle.net.tns_admin=va/ue
```

例:

```
setJavaProp oracle.net.tns_admin=/myoraclehome/network/admin
```

mgw.oraコメント行

コメント行は、行の先頭に#という文字を使用して指定されます。

19 Oracle Messaging Gatewayの操作

Oracle Messaging Gateway(MGW)をロードおよび設定すると、構成および実行が可能になります。

DBMS_MGWADM.ALTER_AGENTを使用して、メッセージ・ゲートウェイ・エージェントがデータベース接続の確立に使用するユーザー名、パスワード、データベース指定子および接続タイプを設定できます。

トピック:

- [Oracle Messaging Gatewayのエージェントの構成](#)
- [Oracle Messaging Gatewayのエージェントの起動と停止](#)
- [メッセージ・システム・リンクの構成](#)
- [Oracle以外のメッセージ・システムのキューの構成](#)
- [Oracle Messaging Gatewayの伝播ジョブの構成](#)
- [伝播ジョブ、サブスクリバおよびスケジュール](#)
- [構成プロパティ](#)

注意:



例で示すすべてのコマンドは、MGW_ADMINISTRATOR_ROLE が付与されたユーザーが実行する必要があります。

関連項目:

[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_MGWADMおよびDBMS_MGWMSGに関する項

Oracle Messaging Gatewayのエージェントの構成

メッセージは、Oracle Database Advanced QueuingとOracle以外のメッセージ・システム間でメッセージ・ゲートウェイによって伝播されます。メッセージ・ゲートウェイ・エージェントは、Oracle Databaseサーバーの外部プロセスとして実行されます。

メッセージ・ゲートウェイでは、1つのデータベースに対して複数のエージェントがサポートされています。デフォルトのエージェントは、DEFAULT_AGENTという名前で自動的に作成されます。伝播ジョブの分離およびグループ化、Oracle RAC環境での拡張を可能にする他の名前エージェントも作成できます。通常、Oracle RAC環境でない1つのインスタンスの場合、デフォルトのエージェントのみで十分です。

この項の内容は次のとおりです。

- [メッセージ・ゲートウェイ・エージェントの作成](#)
- [メッセージ・ゲートウェイ・エージェントの削除](#)
- [データベース接続の設定](#)
- [リソース制限の設定](#)

メッセージ・ゲートウェイ・エージェントの作成

DBMS_MGWADM.CREATE_AGENTを使用して、追加のメッセージ・ゲートウェイ・エージェントを作成できます。メッセージ・エージェントのデフォルトのエージェントであるDEFAULT_AGENTは、メッセージ・エージェントがインストールされると自動的に作成され、常駐するようになります。

エージェントは、作成時または作成後に、DBMS_MGWADM.ALTER_AGENTを使用して、エージェント・ユーザー、接続情報、データベース・サービスおよびリソース制限で構成できます。メッセージ・ゲートウェイ・エージェントは、起動前に、MGW_AGENT_ROLEロールが付与されているデータベース・ユーザーで構成される必要があります。

[例19-1](#)では、myagentという名前のエージェントを作成し、エージェント・ユーザーのデータベース接続情報を指定します。デフォルト値は、その他すべてのパラメータに対して使用されます。

例19-1 メッセージ・ゲートウェイ・エージェントの作成

```
SQL> exec DBMS_MGWADM.CREATE_AGENT (
    agent_name => 'myagent',
    username   => 'mgwagent',
    password   => 'mgwagent_password',
    database   => 'mydatabase');
```

メッセージ・ゲートウェイ・エージェントの削除

メッセージ・ゲートウェイ・エージェントは、DBMS_MGWADM.REMOVE_AGENTをコールすることによって削除できます。エージェントを削除する前に、そのエージェントに関連付けられているすべてのメッセージ・ゲートウェイのリンクを削除し、そのエージェントを停止する必要があります。デフォルトのエージェントのDEFAULT_AGENTは削除できません。[例19-2](#)では、myagentというエージェントを削除します。

例19-2 メッセージ・ゲートウェイ・エージェントの削除

```
SQL> exec DBMS_MGWADM.REMOVE_AGENT (agent_name => 'myagent');
```

データベース接続の設定

メッセージ・ゲートウェイ・エージェントは、データベースの外部プロセスとして実行されます。Oracle Database Advanced Queuingおよびメッセージ・ゲートウェイ・パッケージにアクセスするには、メッセージ・ゲートウェイ・エージェントからデータベースへの接続を確立する必要があります。DBMS_MGWADM.ALTER_AGENTを使用して、メッセージ・ゲートウェイ・エージェントがデータベース接続の確立に使用するユーザー名、パスワードおよびデータベース接続文字列を設定できます。メッセージ・ゲートウェイ・エージェントを起動するには、ユーザーにMGW_AGENT_ROLEが付与されている必要があります。

[例19-3](#)は、ネット・サービス名mydatabaseを使用して、パスワードmgwagent_passwordのユーザーmgwagentに対して構成されているメッセージ・ゲートウェイのデフォルトのエージェントを示したものです。

例19-3 データベース接続情報の設定

```
SQL> exec DBMS_MGWADM.ALTER_AGENT (
    agent_name => 'default_agent',
    username   => 'mgwagent',
    password   => 'mgwagent_password',
    database   => 'mydatabase');
```

リソース制限の設定

DBMS_MGWADM.ALTER_AGENTを使用して、メッセージ・ゲートウェイ・エージェントのリソース制限を設定できます。たとえば、メッ

セッション・エージェント・プロセスのヒープ・サイズ、およびそのエージェント・プロセスで使用される伝播スレッド数を設定できます。デフォルト値は、メモリー・ヒープ64MBおよび伝播スレッド1です。特定のエージェントの場合、エージェントがDBMS_MGWADM.CREATE_AGENTによって作成されるときにこれらの値を指定することもできます。

[例19-4](#)では、myagentというエージェントについて、ヒープ・サイズ96MB、伝播スレッド2を設定します。

メッセージ・ゲートウェイ・エージェントを実行中に、メモリー・ヒープ・サイズと伝播スレッド数は変更できません。

例19-4 リソース制限の設定

```
SQL> exec DBMS_MGWADM.ALTER_AGENT (
            agent_name => 'myagent',
            max_memory  => 96,
            max_threads => 2);
```

Oracle Messaging Gatewayのエージェントの起動と停止

この項の内容は次のとおりです。

- [Oracle Messaging Gatewayのエージェントの起動](#)
- [Oracle Messaging Gatewayのエージェントの停止](#)
- [Oracle Messaging Gatewayのエージェントのスケジューラのジョブ](#)
- [Oracle RAC上でのOracle Messaging Gatewayのエージェントの実行](#)

Oracle Messaging Gatewayのエージェントの起動

メッセージ・ゲートウェイ・エージェントが構成されたら、DBMS_MGWADM.STARTUPを使用して起動できます。[例19-5](#)に、デフォルトのエージェントおよびmyagentエージェントの起動方法を示します。

[「Oracle Messaging Gatewayの監視」](#)で説明されているように、MGW_GATEWAYビューを使用してメッセージ・ゲートウェイ・エージェントのステータスをチェックできます。

例19-5 メッセージ・ゲートウェイ・エージェントの起動

```
SQL> exec DBMS_MGWADM.STARTUP;
SQL> exec DBMS_MGWADM.STARTUP ('myagent');
```

Oracle Messaging Gatewayのエージェントの停止

DBMS_MGWADM.SHUTDOWNを使用して、メッセージ・ゲートウェイ・エージェントを停止できます。[例19-6](#)に、Messaging Gatewayのデフォルトのエージェントおよびmyagentエージェントの停止方法を示します。

[「Oracle Messaging Gatewayの監視」](#)で説明されているように、MGW_GATEWAYビューを使用してメッセージ・ゲートウェイ・エージェントが正常に停止したかどうかをチェックできます。

例19-6 メッセージ・ゲートウェイ・エージェントの停止

```
SQL> exec DBMS_MGWADM.SHUTDOWN;
SQL> exec DBMS_MGWADM.SHUTDOWN ('myagent');
```

Oracle Messaging Gatewayのエージェントのスケジューラのジョブ

メッセージ・ゲートウェイでは、Schedulerジョブを使用して、メッセージ・ゲートウェイ・エージェントを起動します。このジョブは、DBMS_MGWADM.STARTUPプロシージャをコールすると作成されます。ジョブを実行すると、外部プロセスでメッセージ・ゲートウェイ・

エージェントを作成する外部プロシージャがコールされます。ジョブは、次の場合に削除されます。

- DBMS_MGWADM.SHUTDOWNがコールされたためにエージェントが停止する場合
- 再起動不能エラーが発生したためにエージェントが終了する場合

メッセージ・ゲートウェイは、DBMS_SCHEDULERを使用して、繰返し間隔1分で繰返し可能なSchedulerジョブを作成します。ジョブの所有者はSYSです。データベースの停止、データベースの障害または再起動可能なエラーによって指定したジョブ・インスタンスが終了したとき、繰返しジョブによってメッセージ・ゲートウェイ・エージェントは、自動的に再起動が可能になります。指定した時間に実行できるメッセージ・ゲートウェイ・エージェント・ジョブのインスタンスは1つのみです。

各エージェントは、Schedulerジョブ・クラスを使用して、エージェントのSchedulerジョブのサービス・アフィニティを指定します。このジョブ・クラスは、DBMS_MGWADM.CREATE_AGENTまたはDBMS_MGWADM.ALTER_AGENTで指定されるデータベース・サービスで構成されます。データベース・サービスの設定は、データベース管理者が担当します。いずれのデータベース・サービスも指定されない場合、各インスタンスにマップされているデフォルトのデータベース・サービスが使用されます。

メッセージ・ゲートウェイのデフォルトのエージェントで使用されるSchedulerジョブ・クラスの名前は、SYS.MGW_JOBCLS_DEFAULT_AGENTです。デフォルトのエージェントによって使用されるSchedulerジョブの名前は、SYS.MGW_JOB_DEFAULT_AGENTです。

メッセージ・ゲートウェイの指定エージェントで使用されるSchedulerジョブ・クラスの名前は、SYS.MGW_JOBCLS_<agent_name>です。指定エージェントによって使用されるSchedulerジョブの名前は、SYS.MGW_JOB_<agent_name>です。

エージェント・ジョブにエラーが発生すると、エラーは再起動可能エラーと再起動不能エラーに分類されます。再起動可能エラーとは、エージェント・ジョブを再起動すると解決する問題を指します。再起動不能エラーとは、永続的なものでエージェント・ジョブを再起動しても、再度発生する問題を指します。ORA-01089(即時シャットダウン処理中)およびORA-28576(外部プロシージャ・エージェントへのRPC接続が失われました)は、再起動可能エラー例です。ORA-06520(外部ライブラリのロード中にエラーが発生しました)は、再起動不能エラーの例です。

メッセージ・ゲートウェイでは、データベース停止トリガーを使用します。メッセージ・ゲートウェイ・エージェントが停止中のインスタンス上で起動されると、トリガーによってエージェントの停止が通知され、通知を受け取るとエージェントは現在の実行を終了します。ジョブ・スケジューラによって、後で自動的にジョブの再実行がスケジュールされます。

データベース機能障害またはエージェント・ジョブによって検出された再起動不能エラーのため、メッセージ・ゲートウェイ・エージェント・ジョブが終了する場合は、ジョブは削除されず、後でジョブ・スケジューラによって自動的にジョブの再実行がスケジュールされます。

MGW_GATEWAYビューには、メッセージ・ゲートウェイ・エージェントを実行中のエージェント・ステータス、データベース・サービスおよびデータベース・インスタンスが表示されます。Oracle Schedulerビューで、Schedulerジョブ、ジョブ・クラスおよびジョブ実行の詳細に関する情報を確認できます。

関連項目:

- [『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_SCHEDULERに関する項
- [Oracle Messaging Gatewayの監視](#)

Oracle RAC上でのOracle Messaging Gatewayのエージェントの実行

Oracle Real Application Clusters (Oracle RAC)環境とOracle RAC以外の環境で、メッセージ・ゲートウェイ・ジョブの起動原理と停止原理が同じ場合は、Oracle RAC環境に対して、いくつかの点を考慮する必要があります。

Oracle RAC環境でも、構成済のメッセージ・ゲートウェイ・エージェントごとに1つのプロセスを実行できます。たとえば、デフォルトのエージェントと2つの指定エージェントがエージェント・ユーザーで構成されている場合、これら3つのエージェントはすべて同時に実行できます。各エージェントに関連付けられているデータベース・サービスが、エージェントのSchedulerジョブのサービス・アフィニティ、およびエージェント・プロセスが実行されるデータベースのインスタンスを決定します。

Oracle RAC環境でデータベース・インスタンスを停止すると、メッセージ・ゲートウェイ・エージェントが停止中のインスタンスで実行中の場合のみ、メッセージ・ゲートウェイ停止トリガーによってエージェントに停止が通知されます。インスタンスが再起動されると、ジョブ・スケジューラによって自動的に別のインスタンスでジョブの今後の再実行がスケジュールされるか、または停止時のインスタンスでのみジョブを実行できる場合は、そのインスタンスで再実行するジョブがスケジュールされます。

メッセージ・ゲートウェイ・エージェントによるすべてのデータベース接続先を、メッセージ・ゲートウェイ・エージェントのプロセスが実行中のインスタンスにすることをお勧めします。これにより、Oracle RAC環境で確実に適切なフェイルオーバー動作が得られます。

メッセージ・ゲートウェイ・エージェントがデータベース・サービスと関連付けられている場合、サービスが実行中のインスタンスで有効化されるまで、そのエージェントのSchedulerジョブは実行されません。データベースを停止すると、そのデータベースへのすべてのサービスが停止し、データベースの起動時にそのサービスを手動で再起動する必要がある場合があります。

関連項目:

- [Oracle RAC環境におけるOracle Messaging Gatewayの構成](#)
- 『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のDBMS_MGWADMおよびDBMS_SCHEDULERに関する項

メッセージ・システム・リンクの構成

Oracle以外のメッセージ・システムのクライアントとして実行すると、メッセージ・ゲートウェイ・エージェントはメッセージ・システム・リンクを介してOracle以外のメッセージ・システムと通信できます。メッセージ・システム・リンクとは、メッセージ・ゲートウェイ・エージェントとOracle以外のメッセージ・システム間の一連の接続です。

Oracle以外のメッセージ・システムのメッセージ・システム・リンクを構成するには、ユーザーはエージェント情報を指定してOracle以外のメッセージ・システムへの接続を確立する必要があります。ユーザーは、メッセージ・コネクションの最大数を指定できます。

エージェント名は、各メッセージ・システム・リンクに関連付けられます。これは、リンクの作成時に設定され、変更できません。リンクに関連付けられているエージェントは、そのリンクと関連付けられている登録済のキューを使用するすべての伝播ジョブを処理する必要があります。メッセージ・システム・リンクの作成時にエージェント名が指定されていない場合、メッセージ・ゲートウェイのエージェントが使用されます。

トランザクションと永続メッセージをサポートするOracle以外のメッセージ・システムのメッセージ・システム・リンクを構成する場合は、着信伝播および発信伝播のログ・キューの固有の名前を指定して1回のみメッセージが配信されるようにする必要があります。ログ・キューは、メッセージ・ゲートウェイ・エージェントによってのみ使用される必要があります。他のプログラムでは、ログ・キューのメッセージをインキューまたはデキューできません。着信ログ・キューと発信ログ・キューは同じ物理的キューを参照できますが、別の物理的キューを参照した方がパフォーマンスは向上します。

1つのメッセージ・ゲートウェイ・エージェントにつき、1つの伝播ログ・キューにアクセスできます。これにより、指定のログ・キューに、そのエージェントによって処理される伝播ジョブのみに関するログ記録が含まれ、そのエージェントは、発生する可能性のある他のログ記録の破棄が可能であることが保証されます。

メッセージ・システム・リンクを構成する際に、options引数を指定することもできます。options引数とは、SYS.MGW_PROPERTY型の組{名前、値}のセットです。

この項の内容は次のとおりです。

- [WebSphere MQ Base Javaリンクの作成](#)
- [WebSphere MQ JMSリンクの作成](#)
- [SSLを使用するためのWebSphere MQリンクの作成](#)
- [TIB/Rendezvousリンクの作成](#)
- [メッセージ・システム・リンクの変更](#)
- [メッセージ・システム・リンクの削除](#)
- [メッセージ・システム・リンクのビュー](#)

WebSphere MQ Base Javaリンクの作成

WebSphere MQ Base Javaリンクは、次の情報を指定し、DBMS_MGWADM.CREATE_MSGSYSTEM_LINKをコールすることによって作成されます。

- インタフェース型: DBMS_MGWADM.MQSERIES_BASE_JAVA_INTERFACE
- WebSphere MQ接続情報
 - WebSphere MQサーバーのホスト名およびポート番号
 - キュー・マネージャ名
 - チャンネル名
 - ユーザー名とパスワード
- 許可されるメッセージ・コネクションの最大数
- 着信伝播および発信伝播のログ・キュー名
- オプション情報の例
 - 送信、受信およびセキュリティの終了
 - キャラクタ・セット

[例19-7](#)では、WebSphere MQ Base Javaリンクmqlinkを構成します。このリンクは、WebSphere MQチャンネルmychannelを使用して、ホストmyhost.mydomainおよびポート1414上のWebSphere MQキュー・マネージャmy.queue.managerを使用するように構成されます。

この例では、WebSphere MQ SendExitクラスを登録するオプションも設定します。クラスmySendExitは、mgw.oraのCLASSPATHセットにある必要があります。メッセージ・ゲートウェイのデフォルトのエージェント(DEFAULT_AGENT)は、リンク、およびそのリンクを使用するすべての伝播ジョブを処理する必要があります。

関連項目:

- メッセージ・ゲートウェイ・エージェントのCLASSPATHの設定の詳細は、[「mgw.ora初期化ファイルについて」](#)を参照してください。
- [WebSphere MQシステム・プロパティ](#)

例19-7 WebSphere MQ Base Javaリンクの構成

```

DECLARE
  v_options sys.mgw_properties;
  v_prop sys.mgw_mqseries_properties;
BEGIN
  v_prop := sys.mgw_mqseries_properties.construct();

  v_prop.interface_type := dbms_mgwadm.MQSERIES_BASE_JAVA_INTERFACE;
  v_prop.max_connections := 1;
  v_prop.username := 'mqm';
  v_prop.password := 'mqm';
  v_prop.hostname := 'myhost.mydomain';
  v_prop.port := 1414;
  v_prop.channel := 'mychannel';
  v_prop.queue_manager := 'my.queue.manager';
  v_prop.outbound_log_queue := 'mylogq';

  -- Specify a WebSphere MQ send exit class 'mySendExit' to be associated with
  -- the queue.
  -- Note that this is used as an example of how to use the options parameter,
  -- but is not an option that is usually set.
  v_options := sys.mgw_properties(sys.mgw_property('MQ_SendExit',
                                                    'mySendExit'));

  dbms_mgwadm.create_msgsystem_link(
    linkname => 'mqlink', agent_name=>'default_agent', properties => v_prop,
    options => v_options );
END;

```

WebSphere MQ JMSリンクの作成

WebSphere MQ JMSリンクは、次の情報を指定し、DBMS_MGWADM.CREATE_MSGSYSTEM_LINKをコールすることによって作成されます。

- インタフェース型

[Java Message Service](#)(JMS)は、キュー・コネクションとトピック・コネクションを識別します。Sun MicrosystemsのJMS 1.1標準は、単一のJMSコネクションでJMSキューおよびトピックの両方へのアクセスを可能にするドメイン統一をサポートしています。

- インタフェース型DBMS_MGWADM.JMS_CONNECTIONで作成されたWebSphere MQ JMSリンクは、JMSキューおよびトピックの両方へのアクセスに使用できます。これは、WebSphere MQ JMSリンクの推奨インタフェースです。
 - インタフェース型DBMS_MGWADM.JMS_QUEUE_CONNECTIONで作成されたWebSphere MQ JMSリンクは、JMSキューへのアクセスにのみ使用できます。
 - インタフェース型DBMS_MGWADM.JMS_TOPIC_CONNECTIONで作成されたWebSphere MQ JMSリンクは、JMSトピックへのアクセスにのみ使用できます。
- WebSphere MQ接続情報

- WebSphere MQサーバーのホスト名およびポート番号
 - キュー・マネージャ名
 - チャネル名
 - ユーザー名とパスワード
- 許可されるメッセージ・コネクションの最大数
メッセージ・コネクションは[JMSセッション](#)にマップされます。
 - 着信伝播および発信伝播のログ宛先(JMSキューまたは[JMSトピック](#))
ログの宛先タイプはリンクの型に有効である必要があります。JMS統一リンクとJMSキュー・リンクではログの宛先にJMSキューを使用し、JMSトピック・リンクではトピックを使用する必要があります。
 - WebSphere MQ JMS統一リンクまたはキュー・リンクの場合は、ログ・キュー名はWebSphere MQ管理ツールを使用して作成される物理的WebSphere MQ JMSキューの名前である必要があります。
 - WebSphere MQ JMSトピック・リンクの場合は、ログ・キュー名はWebSphere MQ JMSトピックの名前である必要があります。そのトピックによって使用される物理的WebSphere MQキューは、WebSphere MQ管理ツールを使用して作成される必要があります。デフォルトでは、使用される物理的キューはSYSTEM. JMS. D. SUBSCRIBER. QUEUEです。リンク・オプションを使用して、別の物理的キューを指定できます。
 - オプション情報の例
 - 送信、受信およびセキュリティの終了
 - キャラクタ・セット
 - JMSトピックに使用されるWebSphere MQ[パブリッシュ・サブスクライブ](#)構成

[例19-8](#)では、JMSトピック・インタフェースを使用してWebSphere MQキュー・マネージャに対するメッセージ・ゲートウェイ・リンクを構成します。このリンクは、mqjmslinkという名前、WebSphere MQチャネルmychannelを使用して、ホストmyhost.mydomainおよびポート1414上のWebSphere MQキュー・マネージャmy.queue.managerを使用するように構成されます。

この例ではオプション・パラメータを使用して、ログ・トピックで使用されるデフォルト以外の永続的なサブスクライバ・キューを指定します。メッセージ・ゲートウェイ・エージェントmyagentは、リンク、およびそのリンクを使用するすべての伝播ジョブを処理する必要があります。

関連項目:

- JMSキューおよびトピックの詳細は、[「WebSphere MQ JMSキューまたはトピックの登録」](#)を参照してください。
- [WebSphere MQシステム・プロパティ](#)

例19-8 WebSphere MQ JMSリンクの構成

```

DECLARE
  v_options sys.mgw_properties;
  v_prop sys.mgw_mqseries_properties;
BEGIN
  v_prop := sys.mgw_mqseries_properties.construct();
  v_prop.max_connections := 1;

  v_prop.interface_type := DBMS_MGWADM.JMS_TOPIC_CONNECTION;

```



```

v_prop.username := 'mqm';
v_prop.password := 'mqm';
v_prop.hostname := 'myhost.mydomain';
v_prop.port      := 1414;
v_prop.channel   := 'mychannel';
v_prop.queue_manager := 'my.queue.manager';

v_prop.outbound_log_queue := 'mylogtopic'

-- Specify a WebSphere MQ durable subscriber queue to be used with the
-- log topic.
v_options := sys.mgw_properties(
sys.mgw_property('MQ_JMSDurSubQueue', 'myDSQueue'));

DBMS_MGWADM.CREATE_MSGSYSTEM_LINK(
  linkname      => 'mqjmslink',
  agent_name    => 'myagent',
  properties    => v_prop,
  options       => v_options );
END;
```

SSLを使用するためのWebSphere MQリンクの作成

メッセージ・ゲートウェイでは、IBM WebSphere MQとWebSphere MQのJMS接続においてSSLをサポートしています。この項では、WebSphere MQ Base JavaリンクでSSLを使用できるようにメッセージ・ゲートウェイを構成する方法を説明します。同じ情報をWebSphere MQのJMSリンクにも適用できます。メッセージ・ゲートウェイの構成に関しては違いはありません。

WebSphere MQ接続にSSLを使用するには、次のものがが必要です。

- SSLを使用するように構成されたWebSphere MQチャネル。
- トラスト・ストア・ファイルおよびキーストア・ファイル(オプション)は、メッセージ・ゲートウェイ・エージェントのプロセスにアクセス可能な場所に存在します。Oracle RAC環境では、これらのファイルは、同じパス指定を使用してメッセージ・ゲートウェイ・エージェントのプロセスが実行される可能性のある、すべてのインスタンスに対してアクセス可能である必要があります。
- DBMS_MGWADM.CREATE_MSGSYSTEM_LINKを使用して、必要なSSL関連リンク・オプションでWebSphere MQリンクを作成します。少なくとも、チャネルで使用されるSSL暗号スイートを指定するように、MQ_SSLCIPHERSUITEプロパティを設定する必要があります。
- DBMS_MGWADM.SET_OPTIONを使用して、そのリンクに関連付けられているメッセージ・ゲートウェイ・エージェント用に特定のJSSE Javaプロパティを設定します。

JSEE関連プロパティ

- java.net.ssl.keyStore

このプロパティは、キーストアの場所を指定するために使用されます。キーストアは、認証およびデータ整合性など、様々な目的で使用されるキー・マテリアルのデータベースです。
- java.net.ssl.keyStorePassword

このプロパティは、キーストアのパスワードを指定するために使用されます。このパスワードは、キーストアへのアクセス前に、キーストアに含まれているデータの整合性を確認するために使用されます。
- java.net.ssl.trustStore

このプロパティは、トラスト・ストアの場所を指定するために使用されます。トラスト・ストアは、どのクライアントおよびサー

バーが信頼できるか、判断するために使用されるキーストアです。

- java.net.ssl.trustStorePassword

このプロパティは、トラスト・ストアのパスワードを指定するために使用されます。このパスワードは、トラスト・ストアへのアクセス前に、キーストアに含まれているデータの整合性を確認するために使用されます。

java.net.ssl.keyStoreおよびjava.net.ssl.keyStorePasswordプロパティは、WebSphere MQチャネルがSSLクライアント認証を使用するように構成されている場合のみに必要です。

[例19-9](#)では、SSL_RSA_WITH_RC4_128_MD5暗号スイートを使用して、WebSphere MQ Base JavaリンクmqssllinkでSSL接続を使用するよう構成します。リンクに関連付けられたメッセージ・ゲートウェイ・エージェントDEFAULT_AGENTにキーストアおよびトラストストアの両方のJavaプロパティを構成できるように、チャネルがSSLクライアント認証用に構成されていると仮定しています。

DBMS_MGWADM.SET_OPTIONによって設定されるJavaプロパティはエージェントの初回起動時のみ設定されるため、この構成は、メッセージ・ゲートウェイ・エージェントの停止時に実行される必要があります。構成時にエージェントが実行中の場合は、エージェントを停止し、SSL接続の使用前に再起動する必要があります。

注意:



[WebSphere MQ システム・プロパティ](#)

例19-9 SSLのためのWebSphere MQ Base Javaリンクの構成

```
DECLARE
v_options sys.mgw_properties;
v_prop sys.mgw_mqseries_properties;
v_agent varchar2(30) := 'default_agent';
BEGIN
    v_prop := sys.mgw_mqseries_properties.construct();
    v_prop.interface_type := DBMS_MGWADM.MQSERIES_BASE_JAVA_INTERFACE;
    v_prop.max_connections := 1;
    v_prop.username := 'mqm';
    v_prop.password := 'mqm';
    v_prop.hostname := 'myhost.mydomain';
    v_prop.port := 1414;
    v_prop.channel := 'mysslchannel';
    v_prop.queue_manager := 'my.queue.manager';
    v_prop.outbound_log_queue := 'mylogq';

    -- specify the SSL ciphersuite
    v_options := sys.mgw_properties(
        sys.mgw_property('MQ_SSLCIPHERSUITE', 'SSL_RSA_WITH_RC4_128_MD5') );

    -- create the MQSeries link
    DBMS_MGWADM.CREATE_MSGSYSTEM_LINK(linkname => 'mqssllink',
        agent_name => v_agent,
        properties => v_prop,
        options => v_options);

    -- set Java properties for the agent that specify the JSSE security
    -- properties for the keystore and truststore; the paths will be
    -- saved as cleartext and the passwords encrypted

    DBMS_MGWADM.SET_OPTION(target_type => DBMS_MGWADM.AGENT_JAVA_PROP,
        target_name => v_agent,
```

```
option_name => 'javax.net.ssl.keyStore',
option_value => '/tmp/mq_ssl/key.jks',
encrypted => false);
```

```
DBMS_MGWADM.SET_OPTION(target_type => DBMS_MGWADM.AGENT_JAVA_PROP,
target_name => v_agent,
option_name => 'javax.net.ssl.keyStorePassword',
option_value => 'welcome',
encrypted => true);
```

```
DBMS_MGWADM.SET_OPTION(target_type => DBMS_MGWADM.AGENT_JAVA_PROP,
target_name => v_agent,
option_name => 'javax.net.ssl.trustStore',
option_value => '/tmp/mq_ssl/trust.jks',
encrypted => false);
```

```
DBMS_MGWADM.SET_OPTION(target_type => DBMS_MGWADM.AGENT_JAVA_PROP,
target_name => v_agent,
option_name => 'javax.net.ssl.trustStorePassword',
option_value => 'welcome',
encrypted => true);
```

END;

TIB/Rendezvousリンクの作成

TIB/Rendezvousリンクは、DBMS_MGWADM.CREATE_MSGSYSTEM_LINKをコールして作成します。TibrvRvdTransport型の転送を作成するための3つのパラメータ(service、networkおよびdaemon)を指定します。

TIB/Rendezvousメッセージ・システム・リンクには、伝播ログ・キューは必要ありません。ログ情報はメモリーに格納されます。したがって、メッセージ・ゲートウェイは最大1回のメッセージ配信のみを保証できます。

[例19-10](#)では、ローカル・コンピュータ上でrvdデーモンに接続するrvlinkという名前のTIB/Rendezvousリンクを構成します。リンクに対してエージェント名は指定されず、メッセージ・ゲートウェイのデフォルトのエージェント(DEFAULT_AGENT)は、リンクおよびそのリンクを使用するすべての伝播ジョブを処理する必要があります。

関連項目:

[TIB/Rendezvousシステム・プロパティ](#)

例19-10 TIB/Rendezvousリンクの構成

```
DECLARE
v_options sys.mgw_properties;
v_prop    sys.mgw_tibrv_properties;
BEGIN
v_prop := sys.mgw_tibrv_properties.construct();

DBMS_MGWADM.CREATE_MSGSYSTEM_LINK(linkname => 'rvlink', properties => v_prop);
END;
```

メッセージ・システム・リンクの変更

一部のリンク情報は、リンクの作成後にDBMS_MGWADM.ALTER_MSGSYSTEM_LINKを使用して変更できます。メッセージ・ゲートウェイ・エージェントを実行中または停止中でもリンク情報を変更できます。[例19-11](#)では、リンクmqlinkを変更して、

max_connectionsプロパティを変更します。

関連項目:

メッセージ・ゲートウェイ・エージェントを実行中の変更に関する制限については、[「構成プロパティ」](#)を参照してください。

例19-11 WebSphere MQリンクの変更

```
DECLARE
  v_options sys.mgw_properties;
  v_prop sys.mgw_mqseries_properties;
BEGIN
  -- use alter_construct() for initialization
  v_prop := sys.mgw_mqseries_properties.alter_construct();
  v_prop.max_connections := 2;

  DBMS_MGWADM.ALTER_MSGSYSTEM_LINK(
    linkname => 'mqlink', properties => v_prop);
END;
```

メッセージ・システム・リンクの削除

メッセージ・ゲートウェイのリンクに関連付けられたすべての登録済キューがすでに削除されている場合のみ、Oracle以外のメッセージ・システムへのメッセージ・ゲートウェイのリンクをDBMS_MGWADM.REMOVE_MSGSYSTEM_LINKで削除できます。メッセージ・ゲートウェイ・エージェントを実行中または停止中でもリンクを削除できます。[例19-12](#)では、リンクmqlinkを削除します。

例19-12 メッセージ・ゲートウェイのリンクの削除

```
BEGIN
  dbms_mgwadm.remove_msgsystem_link(linkname => 'mqlink');
END;
```

メッセージ・システム・リンクのビュー

MGW_LINKSビューで作成されたリンクを確認できます。[例19-13](#)で示されているように、名前とリンク・タイプが表示されます。

[例19-14](#)に示したように、MGW_MQSERIES_LINKビューとMGW_TIBRV_LINKSビューを使用して、メッセージ・システムの型固有の構成情報をチェックできます。

例19-13 すべてのメッセージ・ゲートウェイのリンクのリスト

```
SQL> select link_name, link_type from MGW_LINKS;
```

LINK_NAME	LINK_TYPE
MQLINK	MQSERIES
RVLINK	TIBRV

例19-14 メッセージ・システム・リンクの構成情報のチェック

```
SQL> select link_name, queue_manager, channel, hostname from mgw_mqseries_links;
```

LINK_NAME	QUEUE_MANAGER	CHANNEL	HOSTNAME
MQLINK	my.queue.manager	mychannel	myhost.mydomain

```
SQL> select link_name, service, network, daemon from mgw_tibrv_links;
```

LINK_NAME	SERVICE	NETWORK	DAEMON
-----------	---------	---------	--------

RVLINK			
--------	--	--	--

Oracle以外のメッセージ・システムのキューの構成

ソース・キュー、宛先キューまたは[例外キュー](#)として伝播に関連しているすべてのOracle以外のメッセージ・システム・キューは、メッセージ・ゲートウェイ管理インタフェースを介して登録する必要があります。伝播に関連するOracle Database Advanced Queuingキューを登録する必要はありません。

この項の内容は次のとおりです。

- [Oracle以外のキューの登録](#)
- [Oracle以外のキューの登録解除](#)
- [Oracle以外の登録済キューのビュー](#)

Oracle以外のキューの登録

Oracle以外のキューは、DBMS_MGWADM. REGISTER_FOREIGN_QUEUEを使用して登録できます。Oracle以外のキューを登録すると、メッセージ・ゲートウェイ・エージェントがキューにアクセスするための情報が提供されます。ただし、Oracle以外のメッセージ・システムの物理的キューは作成されません。物理的キューは、メッセージ・ゲートウェイ・エージェントがキューにアクセスする前に、Oracle以外のメッセージ・システムの管理インタフェースを使用して作成する必要があります。

次の情報を使用して、Oracle以外のキューを登録します。

- キューにアクセスする際に使用するメッセージ・システム・リンク名
- キューの固有名 (Oracle以外のメッセージ・システム内の名前)
- キューのドメイン
 - Point-to-PointキューのDBMS_MGWADM. DOMAIN_QUEUE
 - パブリッシュ・サブスクライブ・キューのDBMS_MGWADM. DOMAIN_TOPIC
- Oracle以外のメッセージ・システム固有のオプション

これらのオプションは、{名前, 値}の組(いずれも文字列)の集合です。

関連項目:

[オプションの外部キューの構成プロパティ](#)

[例19-15](#)は、メッセージ・ゲートウェイ・キューdestqとしてWebSphere MQ Base Javaキューmy_mq_queueを登録する方法を示したものです。

例19-15 WebSphere MQ Base Javaキューの登録

```
BEGIN
DBMS_MGWADM. REGISTER_FOREIGN_QUEUE (
  name          => 'destq',
  linkname      => 'mqlink',
```

```
provider_queue => 'my_mq_queue',
domain         => dbms_mgwadm.DOMAIN_QUEUE);
END;
```

WebSphere MQ Base Javaキューの登録

WebSphere MQではPoint-to-Pointキューのみがサポートされているため、ドメインはDBMS_MGWADM.DOMAIN_QUEUEまたはNULLである必要があります。

WebSphere MQ JMSキューまたはトピックの登録

WebSphere MQ JMSキューを登録する際は、ドメインはDBMS_MGWADM.DOMAIN_QUEUEであり、linknameパラメータはWebSphere MQ JMS統一リンクまたはキュー・リンクを参照する必要があります。

WebSphere MQ JMSトピックを登録する際は、ドメインはDBMS_MGWADM.DOMAIN_TOPICであり、linknameパラメータはWebSphere MQ JMS統一リンクまたはトピック・リンクを参照する必要があります。伝播ソースとして使用されるWebSphere MQ JMSトピックのprovider_queueには、ワイルドカードを含めることができます。[ワイルドカード](#)構文については、WebSphere MQドキュメントを参照してください。

TIB/Rendezvousサブジェクトの登録

TIB/Rendezvousサブジェクトをメッセージ・ゲートウェイに登録する際には、provider_queueパラメータでTIB/Rendezvousサブジェクト名を指定します。登録済TIB/Rendezvousキューのドメインは、DBMS_MGWADM.DOMAIN_TOPICまたはNULLである必要があります。

登録済のTIB/Rendezvousキューでprovider_queueがワイルドカード・サブジェクト名に設定されている場合、インバウンド伝播の伝播ソース・キューとして使用できます。ワイルドカード・サブジェクト名を持つキューを伝播宛先キューまたは例外キューに使用することは避けてください。TIB/Rendezvousで記載されているように、ワイルドカード・サブジェクトにメッセージを送信すると、予期しない動作を引き起こす可能性があります。ただし、メッセージ・ゲートウェイまたはTIB/Rendezvousではこれを阻止しません。

Oracle以外のキューの登録解除

Oracle以外のキューは、それを参照する伝播ジョブが存在しない場合のみ、DBMS_MGWADM.UNREGISTER_FOREIGN_QUEUEで登録を解除できます。

[例19-16](#)では、リンクmqlinkのキューdestqの登録を解除します。

例19-16 Oracle以外のキューの登録解除

```
BEGIN
DBMS_MGWADM.UNREGISTER_FOREIGN_QUEUE (name =>'destq', linkname=>'mqlink');
END;
```

Oracle以外の登録済キューのビュー

[例19-17](#)に示すとおりMGW_FOREIGN_QUEUESビューを使用して、どのOracle以外のキューが登録され、それぞれがどのリンクを使用しているかをチェックできます。

例19-17 登録済キューの確認

```
SELECT name, link_name, provider_queue FROM MGW_FOREIGN_QUEUES;
```

```
NAME    LINK_NAME    PROVIDER_QUEUE
-----
```

Oracle Messaging Gatewayの伝播ジョブの構成

Oracle Database AQキューとOracle以外のメッセージ・システム・キュー間でメッセージを伝播するには伝播ジョブが必要です。各伝播ジョブには、一意の伝播型、ソースおよび宛先の3つが含まれます。

伝播ジョブを作成し、JMS宛先間でメッセージを伝播できます。伝播ジョブを作成し、JMS以外のキュー間でメッセージを伝播できます。メッセージ・ゲートウェイでは、JMS宛先とJMS以外のキュー間におけるメッセージの伝播はサポートされていません。

この項の内容は次のとおりです。

- [伝播ジョブの概要](#)
- [Oracle Messaging Gatewayの伝播ジョブの作成](#)
- [伝播ジョブの有効化および無効化](#)
- [伝播ジョブのリセット](#)
- [伝播ジョブの変更](#)
- [伝播ジョブの削除](#)

伝播ジョブの概要

伝播ジョブは、伝播されるメッセージおよびその方法を指定します。

メッセージ・ゲートウェイでは、双方向のメッセージ伝播が可能です。発信伝播では、メッセージがOracle Database Advanced QueuingからOracle以外のメッセージ・システムに移動されます。着信伝播では、メッセージがOracle以外のメッセージ・システムからOracle Database Advanced Queuingに移動されます。

伝播ソースがPoint-to-Pointキューの場合は、メッセージ・ゲートウェイ・エージェントによってキュー内のすべてのメッセージがソース・キューから宛先に移動されます。伝播ソースがトピック(パブリッシュ・サブスクライブ)の場合は、メッセージ・ゲートウェイ・エージェントによって伝播ソース・キューのサブスクライバがメッセージ・システム上に作成されます。エージェントでは、サブスクライバの作成後、パブリッシュされたすべてのメッセージがソース・キューに移動されるだけです。

メッセージを伝播する際に、メッセージ・ゲートウェイ・エージェントによって、メッセージがソース・メッセージ・システムのフォーマットから宛先メッセージ・システムのフォーマットに変換されます。ユーザーは、メッセージ変換を指定することによってメッセージ変換をカスタマイズできます。メッセージ変換に失敗すると、例外キューが指定されている場合は、エージェントがサブスクライバに伝播できるようにメッセージは例外キューに移動されます。

メッセージ・ゲートウェイの例外キューとOracle Database Advanced Queuingの例外キューは異なります。メッセージ変換に失敗すると、メッセージはメッセージ・ゲートウェイによってメッセージ・ゲートウェイの例外キューに移動されます。メッセージに対してMAX_RETRIESデキューを試行後、メッセージはOracle Database Advanced QueuingによってOracle Database Advanced Queuingの例外キューに移動されます。

Oracle Database Advanced Queuingの例外キューに移動されたメッセージは、関連するメッセージ・ゲートウェイの伝播ジョブ上で回復不能な障害を発生します。この問題を避けるために、メッセージ・ゲートウェイの伝播ジョブの伝播ソースとして使用されるすべてのOracle Database Advanced QueuingのキューのMAX_RETRIESパラメータは、16を超える値に設定する必要があります。

伝播ソース・キューのメッセージ・システムがメッセージ選択をサポートしている場合は、伝播サブスクライバにメッセージ選択ルールを指定できます。メッセージ・セクタの条件を満たすメッセージのみが伝播されます。

[JMSメッセージ](#)の配信モードおよびTIB/Rendezvousのキュー・ポリシーに対するオプションなど、伝播ジョブ・オプションを指定することにより、ユーザーはメッセージの伝播方法を制御できます。

MGW_JOBSビューを使用して、メッセージ・ゲートウェイの伝播ジョブの構成およびステータスを確認できます。

関連項目:

[Oracle Messaging Gatewayの監視](#)

Oracle Messaging Gatewayの伝播ジョブの作成

メッセージ・ゲートウェイの伝播ジョブは、DBMS_MGWADM.CREATE_JOBを使用して作成されます。

JMS伝播以外の伝播ソースが、Oracle Database AQキューである場合、キューはシングル・コンシューマ・キューか、マルチ・コンシューマ・キューのいずれかです。マルチ・コンシューマ・キューの場合、メッセージ・ゲートウェイは、DBMS_MGWADM.CREATE_JOBがコールされると、伝播ジョブのjob_nameに対応するOracle Database AQサブスクリバのMGW_job_nameを作成します。

伝播ソースが、[Oracle Java Message Service](#) (Oracle JMS)トピックまたはWebSphere MQ JMSトピックなどのJMSトピックである場合は、メッセージ・ゲートウェイ・エージェントによってソース・メッセージ・システムのトピックに対してJMSサブスクリバMGW_job_nameが作成されます。エージェントを実行中でない場合は、エージェントを再起動するまでサブスクリバは作成されません。

伝播ソースがキューの場合、そのキューを伝播ソースとして使用して伝播ジョブを1つのみ作成できます。伝播ソースがトピックの場合は、そのトピックを各伝播ジョブがメッセージ・システム内のトピックに対応する固有のサブスクリバを有する伝播ソースとして使用して、複数の伝播ジョブを設定できます。

[例19-18](#)では、メッセージ・ゲートウェイの伝播ジョブjob_aq2mqを作成します。

注意:



WebSphere MQ JMS トピックが伝播ジョブに関連していて、リンクのインタフェース型が DBMS_MGWADM.JMS_TOPIC_CONNECTION の場合、永続サブスクリバ MGL_subscriber_id はログ・トピックに対して作成されます。メッセージ・ゲートウェイの伝播ジョブが正常に削除されると、永続サブスクリバは削除されません。

例19-18 メッセージ・ゲートウェイ伝播ジョブの作成

```
BEGIN
  DBMS_MGWADM.CREATE_JOB (
    job_name          => 'job_aq2mq',
    propagation_type => DBMS_MGWADM.OUTBOUND_PROPAGATION,
    source            => 'mquser.srcq',
    destination      => 'deqq@mqlink');
END;
```

伝播ジョブの有効化および無効化

伝播ジョブは、DBMS_MGWADM.CREATE_JOBによる作成時に最初に有効化または無効化することができます。デフォルトでは、作成時にジョブは有効化されます。伝播ジョブを有効化するにはDBMS_MGWADM.ENABLE_JOB、無効化するにはDBMS_MGWADM.DISABLE_JOBを使用します。ジョブが無効化されている場合、いずれの伝播処理も発生しません。

[例19-19](#)では、伝播ジョブ job_aq2mqの伝播を有効化します。

[例19-20](#)では、伝播ジョブ job_aq2mqの伝播を無効化します。

例19-19 メッセージ・ゲートウェイ伝播ジョブの有効化

```
BEGIN
  DBMS_MGWADM. ENABLE_JOB (job_name => ' job_aq2mq' );
END;
```

例19-20 メッセージ・ゲートウェイ伝播ジョブの無効化

```
BEGIN
  DBMS_MGWADM. DISABLE_JOB (job_name => ' job_aq2mq' );
END;
```

伝播ジョブのリセット

伝播ジョブに問題が発生した場合、メッセージ・ゲートウェイ・エージェントでは伝播ジョブを停止する前に、指数バックオフ・スキームで失敗した操作が最大16回再試行されます。DBMS_MGWADM.RESET_JOBを使用すると、失敗回数をゼロにリセットし、エージェントで失敗した操作をただちに再試行できます。

[例19-21](#)では、伝播ジョブ job_aq2mqの失敗回数をリセットします。

例19-21 伝播ジョブのリセット

```
BEGIN
  DBMS_MGWADM. RESET_JOB (job_name => ' job_aq2mq' );
END;
```

伝播ジョブの変更

伝播ジョブが作成されたら、DBMS_MGWADM.ALTER_JOBを使用して、そのジョブの選択ルール、変換、例外キュー、ジョブ・オプションおよびポーリング間隔を変更できます。メッセージ・ゲートウェイ・エージェントを実行中または停止中でもジョブを変更できます。

[例19-22](#)では、伝播ジョブの例外キューを追加します。

[例19-23](#)では、伝播ジョブのポーリング間隔を変更します。ポーリング間隔によって、エージェントが伝播ソース・キュー内で使用可能なメッセージを検出する速度が決まります。デフォルトのポーリング間隔は5秒、またはメッセージ・ゲートウェイの初期化ファイル内のoracle.mgw.polling_intervalで設定されている値です。

例19-22 例外キューの追加による伝播ジョブの変更

```
BEGIN
  DBMS_MGWADM. ALTER_JOB (
    job_name          => ' job_aq2mq',          exception_queue => 'mgwuser.my_ex_queue');
END;
```

例19-23 ポーリング間隔変更による伝播ジョブの変更

```
BEGIN
  DBMS_MGWADM. ALTER_JOB (
    job_name          => ' job_aq2mq',          poll_interval => 2);
END;
```

伝播ジョブの削除

メッセージ・ゲートウェイの伝播ジョブは、DBMS_MGWADM.REMOVE_JOBをコールすることによって削除できます。

メッセージ・ゲートウェイ構成から伝播ジョブが削除される前に、メッセージ・ゲートウェイは、次のクリーンアップを実行します。

- メッセージ・ゲートウェイによって作成された関連サブスクリバの、メッセージ・システムからの削除
- 削除対象のジョブに関する伝播ログ記録の、ログ・キューからの削除

次の場合には、メッセージ・ゲートウェイはクリーン・アップに失敗することがあります。

- メッセージ・ゲートウェイ・エージェントが実行されていない
- Oracle以外のメッセージ・システムが実行されていない
- メッセージ・ゲートウェイ・エージェントが、ソースまたは宛先メッセージ・システムと対話できない

いずれかの理由によりメッセージ・ゲートウェイがクリーン・アップに失敗した場合、削除対象の伝播ジョブはDELETE_PENDING状態に置かれます。次の場合、メッセージ・ゲートウェイ・エージェントは、DELETE_PENDING状態にある伝播ジョブのクリーンアップを試行します。

- DBMS_MGWADM.REMOVE_JOBがコールされると、メッセージ・ゲートウェイ・エージェントが実行されます。
- メッセージ・ゲートウェイ・エージェントが起動し、DELETE_PENDING状態の伝播ジョブを検出します。

DBMS_MGWADM.REMOVE_JOBには、DELETE_PENDING状態にしなくても、メッセージ・ゲートウェイ構成からの伝播ジョブの強制的な削除を可能にする強制パラメータが含まれています。これは、クリーンアップが失敗した場合、またはメッセージ・ゲートウェイ・エージェントが実行中でないときに伝播ジョブを削除する場合に便利です。

伝播ジョブを強制的に削除すると、使用されなくなったログ記録がログ・キュー内に、サブスクリプションがメッセージ・システム内に残り、不要なメッセージが蓄積されることとなります。DBMS_MGWADM.REMOVE_JOBには、可能なかぎり強制オプションを使用しないことをお勧めします。

[例19-24](#)では、強制しない方法で伝播ジョブを削除します。

例19-24 伝播ジョブの削除

```
BEGIN
  DBMS_MGWADM.REMOVE_JOB (job_name => 'job_aq2mq');
END;
```

伝播ジョブ、サブスクリバおよびスケジュール

サブプログラムは、伝播ジョブの作成および管理を簡素化するDBMS_MGWADMパッケージの一部として提供されます。これらのサブプログラムを使用して、前のリリースのような非結合サブスクリバおよびスケジュールではなく、伝播ジョブを構成できます。伝播ジョブ・プロシージャを使用することをお勧めしますが、下位互換性を保つために、サブスクリバ・プロシージャおよびスケジュール・プロシージャもサポートされています。

[表19-1](#)は、メッセージ・ゲートウェイの伝播ジョブ・プロシージャ、および置換するサブスクリバ・プロシージャまたはスケジュール・プロシージャ(あるいはその両方)を示したものです。すべてのプロシージャは、DBMS_MGWADMパッケージからのものです。

表19-1 メッセージ・ゲートウェイ伝播ジョブのサブプログラム

ジョブ・プロシージャ	サブスクリバ、スケジュール・プロシージャを置換します。
CREATE_JOB	ADD_SUBSCRIBER、SCHEDULE_PROPAGATION
ALTER_JOB	ALTER_SUBSCRIBER、ALTER_PROPAGATION_SCHEDULE

ジョブ・プロシージャ	サブスクライバ、スケジュール・プロシージャを置換します。
REMOVE_JOB	REMOVE_SUBSCRIBER、UNSCHEDULE_PROPAGATION
ENABLE_JOB	ENABLE_PROPAGATION_SCHEDULE
DISABLE_JOB	DISABLE_PROPAGATION_SCHEDULE
RESET_JOB	RESET_SUBSCRIBER

この項では、次の項目について説明します。

- [伝播ジョブ、サブスクライバおよびスケジュールのインタフェースの互換性](#)
- [伝播ジョブ、サブスクライバ、スケジュールのビュー](#)
- [伝播ソースとしてのシングル・コンシューマ・キュー](#)

伝播ジョブ、サブスクライバおよびスケジュールのインタフェースの互換性

新しい形式のジョブと古い形式のジョブという、2つのタイプの伝播ジョブを作成できます。新しい形式のジョブは、DBMS_MGWADM.CREATE_JOBを使用して作成されます。古い形式のジョブは、同じ{propagation_type、source、destination}の3つを使用してDBMS_MGWADM.ADD_SUBSCRIBERおよびDBMS_MGWADM.SCHEDULE_PROPAGATIONをコールすることによって作成されます。スケジュールが一致しないサブスクライバ、またはサブスクライバが一致しないスケジュールは、伝播ジョブとしてみなされません。

新しい形式のジョブの場合、ジョブ名は、サブスクライバIDとスケジュールIDの両方として機能します。古い形式のジョブの場合は、ジョブ名としてサブスクライバIDが使用されます。

古い形式の伝播ジョブには、伝播ジョブ・サブプログラムと、サブスクライバ・サブプログラムまたはスケジュール・サブプログラムの両方を使用できます。伝播ジョブの作成および管理には、ジョブ・サブプログラムを使用することをお勧めします。伝播ジョブは、一致しないサブスクライバまたはスケジュールでは構成されないため、これらに対してジョブ・サブプログラムは使用できません。

新しい形式の伝播ジョブには、新しいジョブ・サブプログラムのみを使用できます。新しい形式のジョブで、サブスクライバまたはスケジュール・プロシージャをコールしようとすると、エラーが発生します。

DBMS_MGWADM.REMOVE_JOB以外、古い形式のジョブ用のジョブ・サブプログラムのコールは簡単で、対応するサブスクライバまたはスケジュール・サブプログラムのコールと同様、効率的な結果が得られます。今後、なんらかの制限がある可能性もありますが、現在はいずれの制限もありません。

DBMS_MGWADM.REMOVE_JOBプロシージャは、新旧の両方の形式のジョブを削除するために使用できます。強制的および非強制的な削除がサポートされています。非強制的な削除の実行時に、メッセージ・ゲートウェイ・エージェントが実行中でない場合、ジョブには削除保留中のフラグが付き、基になるサブスクライバまたはスケジュールのいずれも削除されません。エージェントが再起動し、クリーンアップ作業または強制的なDBMS_MGWADM.REMOVE_JOBが実行されると、ジョブ(サブスクライバとスケジュール・ペア)が削除されます。サブスクライバとスケジュールのペアが同時に削除されるようにするため、DBMS_MGWADM.REMOVE_JOBを最初にコールした後で、古い形式のジョブ用のDBMS_MGWADM.REMOVE_SUBSCRIBERまたはDBMS_MGWADM.UNSCHEDULE_PROPAGATIONのコールを試行すると、エラーが発生します。

DBMS_MGWADM.REMOVE_JOBがジョブにコールされ、削除保留としてフラグ付けされると、DBMS_MGWADM.REMOVE_JOB以外のすべてのジョブ・プロシージャが、新しい形式および古い形式のジョブの両方に関して失敗します。さらに、伝播ジョブが古い形式のジョブであった場合、すべてのサブスクライバおよびスケジュール・サブプログラムも失敗します。

DBMS_MGWADM.REMOVE_SUBSCRIBERおよびDBMS_MGWADM.UNSCHEDULE_PROPAGATIONは、DBMS_MGWADM.REMOVE_JOBがその

ジョブに対してコールされていないかぎり、古い形式のジョブに使用できます。DBMS_MGWADM.UNSCHEDULE_PROPAGATIONが古い形式のジョブに対してコールされた場合、スケジュールが即時削除され、伝播ジョブでなくなり、サブスクライバの削除にはDBMS_MGWADM.REMOVE_SUSCRIBERの使用が必要になります。DBMS_MGWADM.REMOVE_SUBSCRIBERが古い形式のジョブに対してコールされた場合、サブスクライバが存在するかぎり、ユーザーは続けてDBMS_MGWADM.REMOVE_JOBをコールできます。

伝播ジョブ、サブスクライバ、スケジューラのビュー

MGW_JOBSビューには、現在の伝播ジョブ、新旧両方の形式のジョブに関する情報、およびMGW_SUBSCRIBERSビューとMGW_SCHEDULESビューで表示されるすべての関連情報が表示されます。一致しないサブスクライバまたはスケジュールは伝播ジョブを構成せず、MGW_JOBSビューには表示されないため、これらの検索には引き続きMGW_SUBSCRIBERSビューおよびMGW_SCHEDULESビューが役立ちます。

伝播ソースとしてのシングル・コンシューマ・キュー

メッセージ・ゲートウェイを使用して、Oracle Database AQのマルチ・コンシューマ・キューまたはシングル・コンシューマ・キューを、DBMS_MGWADM.CREATE_JOBにより作成される新しい形式の発信ジョブの伝播ソースにすることができます。古い形式の発信ジョブの伝播ソースには、マルチ・コンシューマ・キューを使用する必要があります。管理者がDBMS_MGWADM.ADD_SUBSCRIBERのコールを試行し、ソースがシングル・コンシューマ・キューである場合、エラーが発生します。

伝播ソースがシングル・コンシューマ・キューの場合、システム固有(JMS以外)の発信伝播について、Oracle Database AQのデキュー条件はサポートされていません。

構成プロパティ

この項では、メッセージ・ゲートウェイのリンク、外部キューおよび伝播ジョブに関連する基本的なプロパティとオプションのプロパティについて説明します。

この項の内容は次のとおりです。

- [WebSphere MQシステム・プロパティ](#)
- [TIB/Rendezvousシステム・プロパティ](#)
- [オプションのリンクの構成プロパティ](#)
- [オプションの外部キューの構成プロパティ](#)
- [オプションのジョブの構成プロパティ](#)

WebSphere MQシステム・プロパティ

[表19-2](#)は、WebSphere MQメッセージ・リンクの基本的な構成プロパティをまとめたものです。この表は、SYS.MGW_MQSERIES_PROPERTIESのオプションの(NULLが可能な)プロパティ、変更可能なプロパティ、値が動的に変更可能なプロパティを示しています。

関連項目:

『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のSYS.MGW_MQSERIES_PROPERTIES型に関する項

属性	NULL	値の変更	動的変更
queue_manager	不可	不可	不可
hostname	可(1)	不可	不可
port	可(1)	不可	不可
channel	可(1)、(6)	可	不可
interface_type	可(2)	不可	不可
max_connections	可(3)	可	可
username	可	可	可
password	可	可	可
inbound_log_queue	可(4)	可(4)	可
outbound_log_queue	可(5)	可(5)	可

表19-2に関する注意

- hostnameがNULLの場合、portおよびchannelはNULLである必要があります。hostnameがNULLでない場合、portはNULL以外である必要があります。hostnameがNULLの場合は、WebSphere MQバインド接続が使用され、それ以外の場合はクライアント接続が使用されます。
- interface_typeがNULLの場合は、デフォルト値DBMS_MGWADM.MQSERIES_BASE_JAVA_INTERFACEが使用されます。
- max_connectionsがNULLの場合は、デフォルト値1が使用されます。
- リンクが着信伝播に使用されない場合は、属性inbound_log_queueにNULLを設定できます。ログ・キューを変更できるのは、このリンクが着信伝播ジョブによって参照されない場合のみです。
- リンクが発信伝播に使用されない場合は、属性outbound_log_queueにNULLを設定できます。ログ・キューを変更できるのは、このリンクが発信伝播ジョブによって参照されない場合のみです。
- クライアント・チャンネル定義表(CCDT)が使用される場合、channel属性は、NULLである必要があります。CCDTは、MQ_ccdtURLリンク・オプションを使用して指定できます。

表19-3は、WebSphere MQメッセージ・システムへのアクセスにWebSphere MQ Base Javaインタフェースが使用される場合にサポートされているオプションの構成プロパティをまとめたものです。表19-4は、WebSphere MQ JMSインタフェースが使用される場合にサポートされているオプションの構成プロパティをまとめたものです。それぞれの表は、プロパティ名、そのプロパティの適用場所、プロパティを変更できるかどうか、値を動的に変更可能かどうかを示しています。サポートされているのはこの表に示すプロパティのみで、その他のプロパティは無視されます。

表19-3 WebSphere MQ Base Javaのオプションの構成プロパティ

プロパティ名	使用対象	値の変更	動的変更
"MQ_ccdtUrl"	リンク	可	不可
"MQ_ccsid"	リンク	可	不可
"MQ_ReceiveExit"	リンク	可	不可
"MQ_SecurityExit"	リンク	可	不可
"MQ_SendExit"	リンク	可	不可
"MQ_SSLEncipherSuite"	リンク	可	不可
"MQ_SSLEncipherRequired"	リンク	可	不可
"MQ_SSLPeerName"	リンク	可	不可
"MQ_SSLResetCount"	リンク	可	不可
"MQ_openOptions"	外部キュー	不可	不可
"MsgBatchSize"	ジョブ	可	可
"PreserveMessageID"	ジョブ	可	可

表19-4 WebSphere MQ JMSのオプションの構成プロパティ

プロパティ名	使用対象	値の変更	動的変更
"MQ_BrokerControlQueue"	リンク	可	不可
"MQ_BrokerPubQueue"	リンク	可	不可
"MQ_BrokerQueueManager"	リンク	可	不可
"MQ_BrokerVersion"	リンク	可	不可
"MQ_ccdtUrl"	リンク	可	不可
"MQ_ccsid"	リンク	可	不可

プロパティ名	使用対象	値の変更	動的変更
"MQ_JmsDurSubQueue"	リンク	不可	不可
"MQ_PubAckInterval"	リンク	可	不可
"MQ_ReceiveExit"	リンク	可	不可
"MQ_ReceiveExitInit"	リンク	可	不可
"MQ_SecurityExit"	リンク	可	不可
"MQ_SecurityExitInit"	リンク	可	不可
"MQ_SendExit"	リンク	可	不可
"MQ_SendExitInit"	リンク	可	不可
"MQ_SSLEncipherSuite"	リンク	可	不可
"MQ_SSLEncr"	リンク	可	不可
"MQ_SSLEncrRequired"	リンク	可	不可
"MQ_SSLEncrPeerName"	リンク	可	不可
"MQ_SSLEncrResetCount"	リンク	可	不可
"MQ_CharacterSet"	外部キュー	不可	不可
"MQ_JmsDurSubQueue"	外部キュー	不可	不可
"MQ_JmsTargetClient"	外部キュー	不可	不可
"JMS_DeliveryMode"	ジョブ	可	可
"JMS_NoLocal"	ジョブ	不可	不可
"MsgBatchSize"	ジョブ	可	可
"PreserveMessageID"	ジョブ	可	可

TIB/Rendezvousシステム・プロパティ

表19-5は、TIB/Rendezvousメッセージ・リンクの基本的な構成プロパティをまとめたものです。これは、SYS.MGW_TIBRV_PROPERTIESのオプションの(NULLが可能な)プロパティ、変更可能なプロパティ、値が動的に変更可能なプロパティを示しています。

関連項目:

『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のSYS.MGW_TIBRV_PROPERTIES型に関する項

表19-5 TIB/Rendezvousリンク・プロパティ

属性	NULL	値の変更	動的変更
service	可(1)	不可	不可
daemon	可(1)	不可	不可
network	可(1)	不可	不可
cm_name	可(2)	不可	不可
cm_ledger	可(2)	不可	不可

表19-5に関する注意:

- service、daemonまたはnetworkがNULLの場合は、システムのデフォルト値が使用されます。
- cm_nameおよびcm_ledger属性は、TIB/Rendezvousの証明済メッセージがサポートされる将来の使用に備えて確保されています。現在、TIB/Rendezvousリンクの構成時には、これらのパラメータにNULLを指定する必要があります。

表19-6は、TIB/Rendezvousメッセージ・システムが使用される場合にサポートされているオプションの構成プロパティをまとめたものです。この表は、プロパティ名、そのプロパティの適用場所、プロパティを変更できるかどうか、変更可能な場合は値が動的に変更可能かどうかを示しています。サポートされているのはこの表に示すプロパティのみで、その他のプロパティは無視されます。

表19-6 TIB/Rendezvousのオプション・プロパティ

プロパティ名	使用対象	値の変更	動的変更
"AQ_MsgProperties"	ジョブ	可	可
"MsgBatchSize"	ジョブ	可	可
"PreserveMessageID"	ジョブ	可	可

プロパティ名	使用対象	値の変更	動的変更
"RV_discardAmount"	ジョブ	可	不可
"RV_limitPolicy"	ジョブ	可	不可
"RV_maxEvents"	ジョブ	可	不可

オプションのリンクの構成プロパティ

この項では、DBMS_MGWADM.CREATE_MSGSYSTEM_LINKおよびDBMS_MGWADM.ALTER_MSGSYSTEM_LINKのoptionsパラメータを使用して指定できるオプションのリンク・プロパティについて説明します。それぞれの項目には、そのプロパティを使用できるメッセージ・システムも示されています。

MQ_BrokerControlQueue

このプロパティは WebSphere MQ JMS で使用されます。ブローカの制御キューの名前を指定し、WebSphere MQ JMS 管理ツールの BROKERCONQ プロパティに対応します。WebSphere MQ のデフォルトは SYSTEM.BROKER.CONTROL.QUEUE です。

MQ_BrokerPubQueue

このプロパティは WebSphere MQ JMS で使用されます。ブローカのパブリック・キューの名前を指定し、WebSphere MQ JMS 管理ツールの BROKERPUBQ プロパティに対応します。WebSphere MQ のデフォルトは SYSTEM.BROKER.DEFAULT.STREAM です。

MQ_BrokerQueueManager

このプロパティは WebSphere MQ JMS で使用されます。ブローカのキュー・マネージャの名前を指定し、WebSphere MQ 管理ツールの BROKERQMGR プロパティに対応します。設定されていない場合、デフォルトは使用されません。

MQ_BrokerVersion

このプロパティは WebSphere MQ JMS で使用されます。ブローカのバージョン番号を指定し、WebSphere MQ JMS 管理ツールの BROKERVER プロパティに対応します。WebSphere MQ のデフォルトは 0 です。

MQ_ccdtUrl

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。使用するクライアント・チャンネル定義表(CCDT)の URL 文字列を指定します。設定されていない場

合、CCDT は使用されません。CCDT を使用する場合、SYS.MGW_MQSERIES_PROPERTIES.channel リンク・プロパティは NULL である必要があります。

MQ_ccsid

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。WebSphere MQ メッセージ・ヘッダーの情報を変換するために使用するキャラクタ・セット識別子を指定します。これには、記述的文字列ではなく、キャラクタ・セットの整数値(819 など)を使用します。設定されていない場合、WebSphere MQ のデフォルトのキャラクタ・セット 819 が使用されます。

MQ_JmsDurSubQueue

このプロパティは WebSphere MQ JMS で使用されます。WebSphere MQ JMS のトピック・リンクにのみ適用されます。SYS.MGW_MQSERIES_PROPERTIES の属性 inbound_log_queue および outbound_log_queue は、伝播のロギングに使用される WebSphere MQ JMS トピックの名前を指定します。このプロパティは、ログ・トピックのサブスクライバが永続サブスクリプション・メッセージを取り出す WebSphere MQ キューの名前を指定します。WebSphere MQ のデフォルト・キューは SYSTEM.JMS.D.SUBSCRIBER.QUEUE です。

MQ_PubAckInterval

このプロパティは WebSphere MQ JMS で使用されます。ブローカからの確認を必要とするパブリッシュ・リクエスト間の間隔をメッセージ数で指定し、WebSphere MQ JMS 管理ツールの PUBACKINT プロパティに対応します。WebSphere MQ のデフォルトは 25 です。

MQ_ReceiveExit

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。これは MQReceiveExit インタフェースを実装するクラスの完全修飾 Java クラス名を指定します。このクラスは、メッセージ・ゲートウェイ・エージェントの CLASSPATH に含まれている必要があります。デフォルトはありません。

MQ_ReceiveExitInit

この初期化文字列は WebSphere MQ JMS で使用されます。WebSphere MQ JMS から MQ_ReceiveExit によって指定されるクラスのコンストラクタに渡され、WebSphere MQ JMS 管理ツールの RECEXITINIT プロパティに対応します。デフォルトはありません。

MQ_SecurityExit

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。これは MQSecurityExit インタフェースを実装するクラスの完全修飾 Java クラス名を指定します。このクラスは、メッセージ・ゲートウェイ・エージェントの CLASSPATH に含まれている必要があります。デフォルトはありません。

MQ_SecurityExitInit

この初期化文字列は WebSphere MQ JMS で使用されます。WebSphere MQ JMS から MQ_SecurityExit によって指定されるクラスのコンストラクタに渡され、WebSphere MQ JMS 管理ツールの SEEXITINIT プロパティに対応します。デフォルトはありません。

MQ_SendExit

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。これは MQSendExit インタフェースを実装するクラスの完全修飾 Java クラス名を指定します。このクラスは、メッセージ・ゲートウェイ・エージェントの CLASSPATH に含まれている必要があります。デフォルトはありません。

MQ_SendExitInit

この初期化文字列は WebSphere MQ JMS で使用されます。WebSphere MQ JMS から MQ_SendExit によって指定されるクラスのコンストラクタに渡されます。WebSphere MQ JMS 管理ツールの SENDEXITINIT プロパティに対応します。デフォルトはありません。

MQ_SSLCipherSuite

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。SSL_RSA_WITH_RC4_128_MD5 のように、使用する暗号スイートを指定します。これは WebSphere MQ の SSLCIPHERSUITE 管理プロパティに対応します。

MQ_SSLCrl

このプロパティは WebSphere MQ JMS で使用されます。証明書失効リスト(CRL)の確認に使用できる LDAP サーバーのスペース区切りリストを指定します。設定されていない場合、CRL は確認されません。これは WebSphere MQ の SSLCRL 管理プロパティに対応します。このオプションは WebSphere MQ Base Java ではサポートされていません。CRL の確認が必要な場合、かわりにクライアント・チャネル定義表(CCDT)を使用する必要があります。

MQ_SSLFipsRequired

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。使用されている暗号スイートが IBM Java JSSE FIPS プロバイダ (IBMSJSSEFIPS) によってサポートされているかどうかを示します。この値は TRUE または FALSE のいずれかです。デフォルト値は FALSE です。これは WebSphere MQ の SSLFIPSREQUIRED 管理プロパティに対応します。

MQ_SSLPeerName

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。接続を確立するためにキュー・マネージャの証明書が一致する必要がある識別名 (DN) のパターンを指定します。設定されていない場合、DN は確認されません。これは WebSphere MQ の SSLPEERNAME 管理プロパティに対応します。

MQ_SSLResetCount

このプロパティは WebSphere MQ Base Java および WebSphere MQ JMS で使用されます。秘密キーが再ネゴシエートされるまでに送受信される合計バイト数を指定します。設定されていない場合、キーは再ネゴシエートされません。これは WebSphere MQ の SSLRESETCOUNT 管理プロパティに対応します。

オプションの外部キューの構成プロパティ

この項では、DBMS_MGWADM.REGISTER_FOREIGN_QUEUEのoptionsパラメータを使用して指定できるオプションの外部キュー・プロパティについて説明します。それぞれの項目には、そのプロパティを使用できるメッセージ・システムも示されています。

MQ_CharacterSet

このプロパティは WebSphere MQ JMS で使用されます。これは JMS キューまたはトピックへのアウトバウンド伝播にのみ使用されます。宛先に送信されるテキスト文字列のエンコードに使用されるキャラクタ・セットを指定します。これには、記述的文字列ではなく、キャラクタ・セットの整数値 (1208 など) を使用します。メッセージ・ゲートウェイで使用されるデフォルト値は 1208 (UTF8) です。

MQ_JmsDurSubQueue

このプロパティは WebSphere MQ JMS で使用されます。これは、このトピックのサブスクライバが永続サブスクリプション・メッセージを取り出す WebSphere MQ キューの名前を表す文字列です。WebSphere MQ JMS のトピックにのみ適用されます。WebSphere MQ のデフォルト・キューは SYSTEM. JMS. D. SUBSCRIBER. QUEUE です。

MQ_JmsTargetClient

このプロパティは WebSphere MQ JMS で使用されます。これは JMS キューまたはトピックへのアウト

トバウンド伝播にのみ使用されます。サポートされている値は TRUE および FALSE です。TRUE は WebSphere MQ がメッセージを JMS メッセージとして保存することを示します。FALSE は、JMS 以外のアプリケーションがアクセスできるよう、WebSphere MQ がメッセージを非 JMS フォーマットで保存することを示します。デフォルトは TRUE です。

MQ_openOptions

このプロパティは WebSphere MQ Base Java で使用されます。WebSphere MQ Base Java MQQueueManager.accessQueue メソッドの openOptions 引数に使用される値を指定します。値は必要がありません。しかし、入力された場合は、メッセージ・ゲートウェイ・エージェントによって MQ00_OUTPUT がエンキュー(put)操作の指定された値に追加されます。デキュー(get)操作には MQ00_INPUT_SHARED が追加されます。デフォルトは、エンキュー/put 操作では MQ00_OUTPUT、デキュー/get 操作では MQ00_INPUT_SHARED です。

オプションのジョブの構成プロパティ

この項では、オプションのパラメータの DBMS_MGWADM.CREATE_JOB および DBMS_MGWADM.ALTER_JOB を使用して指定可能なオプションの伝播ジョブのプロパティについて説明します。

AQ_MsgProperties

このプロパティは TIB/Rendezvous で使用されます。メッセージが伝播される際に Oracle Database AQ メッセージ・プロパティがどのように使用されるかを指定します。サポートされている値は TRUE および FALSE です。デフォルト値は FALSE です。

出力伝播ジョブの場合、値が TRUE(大/小文字区別なし)であれば、メッセージ・ゲートウェイ・エージェントは TIB/Rendezvous サブジェクトに伝播されるメッセージにほとんどの Oracle Database AQ メッセージ・プロパティのフィールドを追加します。

着信伝播ジョブの場合、値が TRUE(大/小文字区別なし)であれば、メッセージ・ゲートウェイ・エージェントは予約済の名前を持つフィールドをソース・メッセージ内で検索し、存在する場合は、その値を使用して対応する Oracle Database AQ メッセージ・プロパティを設定します。フィールドが存在しない場合、または予期したデータ型でない場合は、デフォルト値が使用されます。

JMS_DeliveryMode

このプロパティは WebSphere MQ JMS および Oracle JMS で使用されます。このプロパティは、伝播先が JMS メッセージ・システムの場合に使用できます。JMS MessageProducer によって伝播先キューにエンキューされるメッセージの配信モードを設定します。デフォルトは PRESERVE_MSG です。サポートされる値とその関連付けられた配信モードは次のとおりです。

- PERSISTENT (DeliveryMode.PERSISTENT)

- NON_PERSISTENT(DeliveryMode. NON_PERSISTENT)
- PRESERVE_MSG(ソース JMS メッセージの配信モードを使用)

JMS_NoLocal

このプロパティは WebSphere MQ JMS および Oracle JMS で使用されます。これは、伝播ソースが JMS メッセージ・システムの場合に使用できます。JMS TopicSubscriber の noLocal パラメータを設定します。TRUE は、同じメッセージ・ゲートウェイ・リンクを介してこのトピックにパブリッシュされたメッセージは伝播されないことを示します。デフォルト値は FALSE で、このようなメッセージがトピックから伝播されることを示します。

MsgBatchSize

このプロパティは、サポートされているすべてのメッセージ・システムで使用できます。これは、1 回のトランザクションで伝播可能なメッセージの最大数(使用可能な場合)を指定します。デフォルトは 30 です。

PreserveMessageID

このプロパティは、WebSphere MQ Base Java、WebSphere MQ JMS、TIB/Rendezvous および Oracle JMS で使用されます。メッセージが宛先メッセージ・システムに伝播される際、メッセージ・ゲートウェイが元のメッセージ識別子を保持するかどうかを指定します。詳細は、使用されるメッセージ・システムの機能によって異なります。サポートされている値は TRUE および FALSE です。デフォルト値は FALSE です。

RV_discardAmount

このプロパティは TIB/Rendezvous で使用されます。キューの廃棄量を指定します。インバウンド伝播ジョブでのみ意味があります。デフォルトは 0 (ゼロ)です。

RV_limitPolicy

このプロパティは TIB/Rendezvous で使用されます。キュー制限のオーバーフローを解決するための制限ポリシーを指定します。インバウンド伝播ジョブでのみ意味があります。デフォルトは DISCARD_NONE です。サポートされている値と関連付けられている制限ポリシーは、DISCARD_NONE、DISCARD_FIRST、DISCARD_LAST および DISCARD_NEW です。

- DISCARD_NONE(TibrvQueue. DISCARD_NONE)
- DISCARD_FIRST(TibrvQueue. DISCARD_FIRST)

- DISCARD_LAST(TibrvQueue. DISCARD_LAST)
- DISCARD_NEW(TibrvQueue. DISCARD_NEW)

RV_maxEvents

このプロパティは TIB/Rendezvous で使用されます。キューの最大イベント制限を指定します。インバウンド伝播ジョブでのみ意味があります。デフォルトは 0 (ゼロ)です。

20 Oracle Messaging Gatewayのメッセージの変換

次のトピックでは、Oracle Messaging Gateway (MGW)がメッセージ・フォーマットを1つのメッセージ・システムから別のメッセージ・システムに変換する方法について説明します。通常、変換が必要になるのはメッセージをOracle Database Advanced Queuingと別のシステムの間で移動する場合で、これは異なるメッセージ・システムでは異なるメッセージ・フォーマットが使用されているためです。[Java Message Service \(JMS\)](#)メッセージは特殊です。[JMSメッセージ](#)はJMS宛先へのみ伝播可能なので、変換は単純な処理になります。

- [Oracle Messaging GatewayのJMS以外のメッセージの変換](#)
- [WebSphere MQに対するメッセージ変換](#)
- [TIB/Rendezvousに対するメッセージ変換](#)
- [JMSメッセージ](#)

Oracle Messaging GatewayのJMS以外のメッセージの変換

MGWでは、[伝播](#)中に、ソース側メッセージ・システム固有のメッセージ・フォーマットが宛先側メッセージ・システム固有のメッセージ・フォーマットに変換されます。MGWでは、[標準](#)型およびOracle Database Advanced Queuingベースのモデルを使用して変換が行われます。

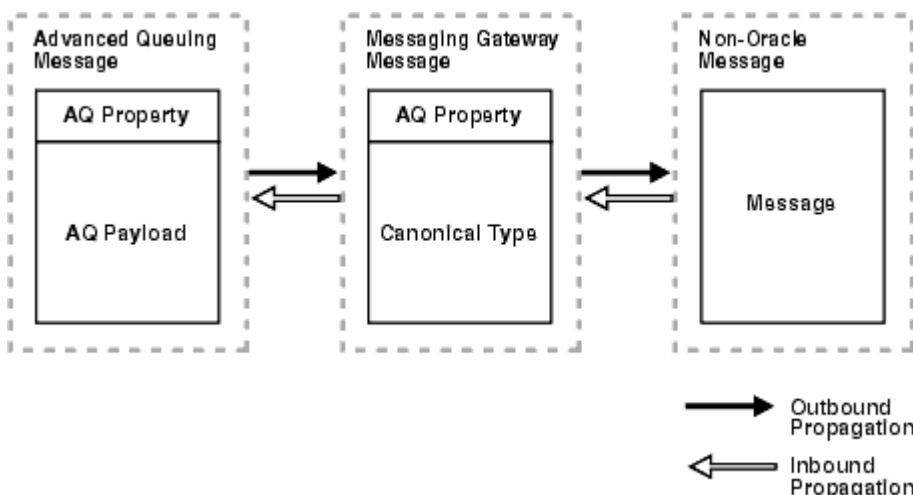
JMS以外のメッセージ変換プロセスの概要

MGWによってメッセージが伝播される場合、メッセージはソース・[キュー](#)・システム固有のフォーマットから宛先キュー・システム固有のフォーマットに変換されます。

ネイティブなメッセージには、通常、メッセージ・ヘッダーとメッセージ本体が含まれます。ヘッダーには、メッセージ・システム内のすべてのメッセージに存在する固定ヘッダー・フィールドが含まれます。たとえば、Oracle Database Advanced Queuingのメッセージ・プロパティやWebSphere MQの固定ヘッダーなどです。本体には、Oracle Database Advanced Queuingのペイロード、WebSphere MQのメッセージ本体、またはTIB/Rendezvousのメッセージ全体などのメッセージ・コンテンツが含まれます。MGWでは、メッセージ・ヘッダーおよびメッセージ本体コンポーネントの両方が変換されます。

[図20-1](#)は、2つのステージでJMS以外のメッセージを変換する方法を示したものです。メッセージは、最初にソース・キューのシステム固有のフォーマットからMGWの内部メッセージ・フォーマットに変換され、次に、内部メッセージ・フォーマットから宛先キューのシステム固有のフォーマットに変換されます。

図20-1 JMS以外のメッセージの変換



MGWエージェントでは、Oracle Database Advanced Queuingのメッセージ・プロパティに類似したヘッダーと、MGWの標準型で表される本体で構成された、内部メッセージ・フォーマットが使用されます。

Oracle Messaging Gateway標準型

MGWでは標準型が定義され、Oracle Database Advanced QueuingとOracle以外のメッセージ・システム間でのメッセージ変換がサポートされます。標準型は、メッセージ型をOracle DatabaseでのPL/SQL Oracleタイプの形式で表したものです。標準型は、RAW、SYS.MGW_BASIC_MSG_TおよびSYS.MGW_TIBRV_MSG_Tです。

WebSphere MQ伝播では、標準型SYS.MGW_BASIC_MSG_TおよびRAWがサポートされます。TIB/Rendezvous伝播では、標準型SYS.MGW_TIBRV_MSG_TおよびRAWがサポートされます。

関連項目:

SYS.MGW_BASIC_MSG_TおよびSYS.MGW_TIBRV_MSG_Tの構文および属性情報は、『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のDBMS_MGWMSGに関する項を参照してください。

メッセージ・ヘッダーの変換

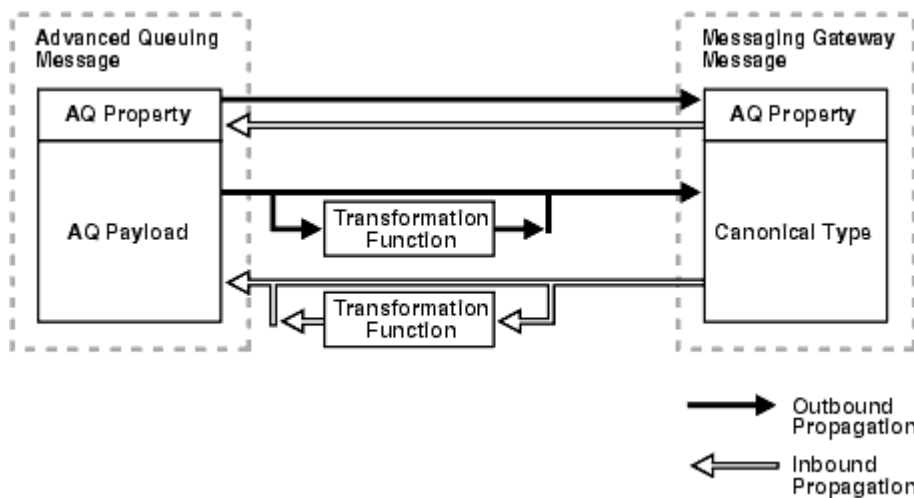
MGWでは、Oracle Database Advanced Queuingメッセージ・プロパティと、同じセマンティクスのOracle Database Advanced Queuingメッセージ・プロパティに対応する値を含むOracle以外のメッセージ・ヘッダー・フィールドとの間で、デフォルトのマッピングが行われます。MGWによってマッピングが行われない場合、メッセージ・ヘッダー・フィールドはデフォルト値(通常、メッセージ・システムによって定義されたデフォルト値)に設定されます。

メッセージ変換を使用した任意のペイロード型の処理

Oracle Database Advanced Queuingメッセージ間で変換する場合、MGWエージェントでは通常、標準型のみが使用されます。ただし、任意のペイロード型は、Oracle Database Advanced Queuingのキュー・ペイロードとMGW標準型間で変換するために、ユーザー定義のOracle Database Advanced Queuingのメッセージ変換機能によってサポートされません。

任意のADTペイロード(発信伝播)によってOracle Database Advanced Queuingのキューからメッセージを伝播するMGWには、MGWの標準ユーザー定義型に対するマッピングが必要です。[変換](#)は、MGWエージェントによってOracle Database Advanced Queuingのキューからメッセージがデキューされると起動されます。同様に、任意のADTペイロード(着信伝播)によってOracle Database Advanced Queuingのキューにメッセージを伝播するMGWには、MGWの標準ADTからのマッピングが必要です。変換は、MGWエージェントによってOracle Database Advanced Queuingのキューにメッセージがエンキューされると起動されます。

図20-2 Oracle Database Advanced Queuingのメッセージ変換



変換は常にMGWエージェントとの関連で実行されますが、これはMGWエージェント・ユーザー(DBMS_MGWADM.CREATE_AGENT またはDBMS_MGWADM.ALTER_AGENTを使用して指定したユーザー)には、変換ファンクションおよびOracle Database Advanced Queuingペイロード型に対するEXECUTE権限が必要であることを意味しています。これは、PUBLICにEXECUTE権限を付与するか、またはMGWエージェントのユーザーに直接EXECUTE権限を付与することによって実現できます。

変換によってMGW伝播ジョブを構成する手順は、次のとおりです。

1. 変換ファンクションを作成します。
2. MGWエージェントのユーザーまたはPUBLICに、ファンクションおよびそのファンクションが参照するオブジェクト型に対するEXECUTE権限を付与します。
3. DBMS_TRANSFORM.CREATE_TRANSFORMATIONをコールして、変換を登録します。
4. 変換によってMGW伝播ジョブを作成する場合はDBMS_MGWADM.CREATE_JOBを、既存のジョブを変更する場合はDBMS_MGWADM.ALTER_JOBをコールします。

例20-1 変換ファンクションの署名

```
FUNCTION trans_sampleadt_to_mgw_basic(in_msg IN mgwuser.sampleADT)
RETURN SYS.MGW_BASIC_MSG_T;
```

[例20-2](#)に示すように、DBMS_TRANSFORM.CREATE_TRANSFORMATIONを使用して変換を作成できます。

例20-2 変換の作成

```
BEGIN
  DBMS_TRANSFORM.CREATE_TRANSFORMATION(
    schema      => 'mgwuser',
    name        => 'sample_adt_to_mgw_basic',
    from_schema => 'mgwuser',
    from_type   => 'sampleadt',
    to_schema   => 'sys',
    to_type     => 'MGW_BASIC_MSG_T',
    transformation => 'mgwuser.trans_sampleadt_to_mgw_basic(user_data)');
END;
```

例20-3 変換の登録

```
BEGIN
  DBMS_MGWADM.CREATE_JOB(
    job_name      => 'job_aq2mq',
    propagation_type => DBMS_MGWADM.OUTBOUND_PROPAGATION,
    source        => 'mgwuser.srcq',
    destination   => 'destq.mqlink',
    transformation => 'mgwuser.sample_adt_to_mgw_basic',
    exception_queue => 'mgwuser.excq');
```

これらのAPIの変換パラメータに渡される値は、ファンクション名ではなく登録された変換名である必要があります。たとえば、trans_sampleadt_to_mgw_basicは、[例20-1](#)で示した署名によって変換ファンクションを表すストアド・プロシージャです。

注意:



この例のすべてのコマンドは、変換を作成するコマンドを除き、MGW_ADMINISTRATOR_ROLE が付与されたユーザーとして実行する必要があります。

いったん作成した後は、伝播ジョブの作成時にこの変換をMGWに登録できます。[例20-3](#)ではジョブ job_aq2mqを作成します。このメッセージはOracle Database Advanced Queuingのキューmgwuser.srcqからOracle以外のメッセージ・システム・キューdestq@mqlinkに変換mgwuser.sample_adt_to_mgw_basicを使用して伝播されます。

通常、ユーザー定義の変換を実行中に発生するエラーはメッセージ変換例外とみなされ、メッセージは[例外キュー](#)に移動されず(存在する場合)。

関連項目:

『[Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のDBMS_MGWADM、DBMS_MGWMSGおよびDBMS_TRANSFORMに関する項

論理変更レコードの処理

MGWでは、論理変更レコード(LCR)を伝播する機能を提供しています。変換の作成を支援し、ペイロード型ANYDATAでキューに格納されているLCR行およびDDL LCRの両方の伝播を処理するルーチンが提供されています。LCRは、適切なメッセージ型で格納されているXML文字列として伝播されます。

注意:



LCR 伝播については、XDB パッケージをロードする必要があります。

Oracle Streamsでは、ANYDATAキューを使用してLCRを格納するため、ANYDATAキューが発信伝播のソースとなります。変換では最初にMGWルーチンDBMS_MGWMSG.LCR_TO_XMLを使用して、LCRを含むANYDATAオブジェクトをXML型オブジェクトに変換する必要があります。ANYDATAオブジェクトにLCRが含まれない場合は、このルーチンでエラーが発生します。次に、LCRのXML文書の文字列がXML型から抽出され、適切なMGW標準型(SYS.MGW_BASIC_MSG_TまたはSYS.MGW_TIBRV_MSG_T)に配置されます。

[例20-4](#)は、LCR発信伝播用に使用される変換を簡素化して示したものです。変換によって、LCRを含むANYDATAペイロードがSYS.MGW_TIBRV_MSG_Tオブジェクトに変換されます。LCRをXML文書として表す文字列が、ORACLE_LCRというフィールドに配置されます。

LCR着信伝播の場合、MGW標準型(SYS.MGW_BASIC_MSG_TまたはSYS.MGW_TIBRV_MSG_T)が変換ソースの型となります。

LCRをXML文書のフォーマットで表す文字列は、標準型に含まれる必要があります。変換関数はメッセージから文字列を抽出し、そこからXML型オブジェクトを作成し、MGWルーチンDBMS_MGWMSG.XML_TO_LCRを使用してLCRを含むANYDATAオブジェクトに変換する必要があります。元のXML文書がLCRを表していない場合は、このルーチンでエラーが発生します。

[例20-5](#)は、LCR着信伝播用に使用される変換を簡素化して示したものです。変換によって、LCRを表すXML文字列を含むフィールドがあるSYS.MGW_TIBRV_MSG_Tオブジェクトが、ANYDATAオブジェクトに変換されます。LCRをXML文書として表す文字列が、ORACLE_LCRというフィールドから取得されます。

関連項目:

- [『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_MGWMSGに関する項
- LCR変換の詳細例は、[ORACLE_HOME/mgw/samples/lcr](#)を参照してください。

例20-4 発信LCRの変換

```
create or replace function any2tibrv(adata in anydata)
return SYS.MGW_TIBRV_MSG_T is
  v_xml XMLType;
  v_text varchar2(2000);
  v_tibrv sys.mgw_tibrv_msg_t;
BEGIN
  v_xml := dbms_mgwmsg.lcr_to_xml(adata);
  -- assume the lcr is smaller than 2000 characters long.
  v_text := v_xml.getStringVal();
  v_tibrv := SYS.MGW_TIBRV_MSG_T.CONSTRUCT;
  v_tibrv.add_string('ORACLE_LCR', 0, v_text);
  return v_tibrv;
END any2tibrv;
```

例20-5 着信LCRの変換

```
create or replace function tibrv2any(tdata in sys.mgw_tibrv_msg_t)
return anydata is
  v_field sys.mgw_tibrv_field_t;
  v_xml XMLType;
  v_text varchar2(2000);
  v_any anydata;
BEGIN
  v_field := tdata.get_field_by_name('ORACLE_LCR');
  -- type checking
  v_text := v_field.text_value;
  -- assume it is not null
  v_xml := XMLType.createXML(v_text);
  v_any := dbms_mgwmsg.xml_to_lcr(v_xml);
  return v_any;
END tibrv2any;
```

WebSphere MQに対するメッセージ変換

MGWによって、MGW標準型とWebSphere MQ固有のメッセージ・フォーマット間の変換が行われます。WebSphere MQ固有のメッセージは、固定されたメッセージ・ヘッダーとメッセージ本体で構成されます。メッセージ本体は、TEXT値またはRAW(バイト)値のいずれかとして処理されます。WebSphere MQ伝播用にサポートされている標準型は、SYS.MGW_BASIC_MSG_TとRAWです。

図20-3 SYS.MGW_BASIC_MSG_T使用によるWebSphere MQのメッセージ変換

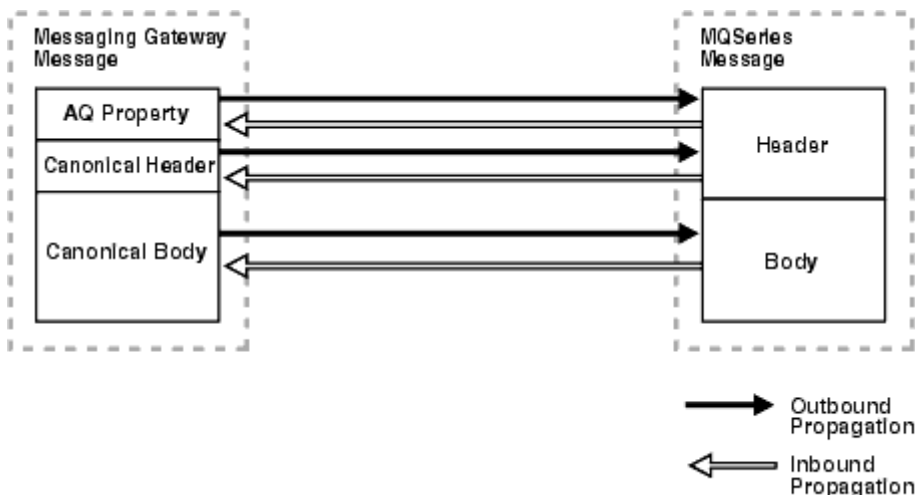


図20-3は、MGW WebSphere MQドライバで標準型SYS.MGW_BASIC_MSG_Tを使用して実行されるメッセージ変換を示したものです。発信伝播の場合、ドライバによって、Oracle Database Advanced Queuingメッセージ・プロパティおよび標準型メッセージが、固定ヘッダーおよびメッセージ本体を含むWebSphere MQシステム固有のメッセージにマップされます。着信伝播の場合、ドライバによって、システム固有のメッセージが一連のOracle Database Advanced Queuingメッセージ・プロパティおよび標準型メッセージにマップされます。標準型がRAWの場合、標準型ヘッダーが存在しないこと以外はマッピング方法は同じです。

WebSphere MQメッセージ・ヘッダーのマッピング

MGW発信伝播ジョブで使用される標準型がRAWの場合、WebSphere MQヘッダー情報はRAWメッセージ本体からは設定されません。同様に着信伝播の場合、WebSphere MQヘッダー情報はRAWメッセージ本体には保持されません。ただし、MGW標準型SYS.MGW_BASIC_MSG_Tには、発信伝播のWebSphere MQヘッダー・フィールドを指定する際に使用でき、さらに着信伝播のWebSphere MQヘッダー・フィールドを保持できるヘッダーがあります。

この項では、SYS.MGW_BASIC_MSG_Tを標準型として使用してWebSphere MQメッセージ・システムに対してサポートされているメッセージ・プロパティについて説明します。表20-1に、WebSphere MQヘッダー・プロパティの記述に使用されるMGWの{名前, 値}の組の定義を示します。最初の列は、SYS.MGW_BASIC_MSG_TヘッダーのSYS.MGW_NAME_VALUE_T.NAMEフィールドの有効文字列値です。2番目の列は、名前に対応するSYS.MGW_NAME_VALUE_T.TYPE値です。(括弧内の数値の詳細は、各行はメッセージ・ゲートウェイの名前に対応します。第1列は名前、第2列はメッセージ・ゲートウェイの型、第3列は対応するWebSphere MQプロパティ名、第4列はその用途を示します。を参照してください。)

着信伝播の場合、WebSphere MQドライバによって、ソース・メッセージ・ヘッダーに基づいて{名前, 値}の組が生成され、SYS.MGW_BASIC_MSG_T型の標準型メッセージのヘッダー部分に格納されます。発信伝播の場合、WebSphere MQドライバによって、SYS.MGW_BASIC_MSG_T標準型メッセージのヘッダー部分に格納されたこれらのプロパティに対する{名前, 値}の組から、メッセージ・ヘッダーおよびエンキュー・オプションが設定されます。

表20-1 WebSphere MQのヘッダー値に対するMGWの名前

MGWの名前	MGWの型	WebSphere MQプロパティ名	使用対象
MGW_MQ_accountingToken	RAW_VALUE (size 32)	accountingToken	発信(1)、着信
MGW_MQ_applicationIdData	TEXT_VALUE (size 32)	applicationIdData	発信(1)、着信

MGWの名前	MGWの型	WebSphere MQプロパティ名	使用対象
MGW_MQ_applicationOriginData	TEXT_VALUE (size 4)	applicationOriginData	発信(1)、着信
MGW_MQ_backoutCount	INTEGER_VALUE	backoutCount	着信
MGW_MQ_characterSet	INTEGER_VALUE	characterSet	発信、着信
MGW_MQ_correlationId	RAW_VALUE (size 24)	correlationId	発信(1)、着信
MGW_MQ_encoding	INTEGER_VALUE	encoding	発信、着信
MGW_MQ_expiry	INTEGER_VALUE	expiry	発信、着信
MGW_MQ_feedback	INTEGER_VALUE	feedback	発信、着信
MGW_MQ_format	TEXT_VALUE (size 8)	format	発信(1)、着信
MGW_MQ_groupId	RAW_VALUE (size 24)	groupId	発信(1)、着信
MGW_MQ_messageFlags	INTEGER_VALUE	messageFlags	発信、着信
MGW_MQ_messageId	RAW_VALUE (size 24)	messageId	発信、着信
MGW_MQ_messageSequenceNumber	INTEGER_VALUE	messageSequenceNumber	発信、着信
MGW_MQ_messageType	INTEGER_VALUE	messageType	発信、着信
MGW_MQ_offset	INTEGER_VALUE	offset	発信、着信
MGW_MQ_originalLength	INTEGER_VALUE	originalLength	発信、着信
MGW_MQ_persistence	INTEGER_VALUE	persistence	着信
MGW_MQ_priority	INTEGER_VALUE	priority	発信、着信
MGW_MQ_putApplicationName	TEXT_VALUE (size 28)	putApplicationName	発信(1)、着信
MGW_MQ_putApplicationType	INTEGER_VALUE	putApplicationType	発信(1)、着信
MGW_MQ_putDateTime	DATE_VALUE	putDateTime	着信

MGWの名前	MGWの型	WebSphere MQプロパティ名	使用対象
MGW_MQ_putMessageOptions	INTEGER_VALUE	putMessageOptions	発信(1) (2)
MGW_MQ_replyToQueueManagerName	TEXT_VALUE (size 48)	replyToQueueManagerName	発信、着信
MGW_MQ_replyToQueueName	TEXT_VALUE (size 48)	replyToQueueName	発信、着信
MGW_MQ_report	INTEGER_VALUE	report	発信(1)、着信
MGW_MQ_userId	TEXT_VALUE (size 12)	userId	発信、着信

関連項目:

[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_MGWMSGに関する項

表20-1に関する注意

- この表での使用方法には、WebSphere MQの制約が適用されます。たとえば、送信メッセージに対してMGW_MQ_accountingTokenが設定されている場合、WebSphere MQの値は、MGW_MQ_putMessageOptionsがWebSphere MQ定数のMQPMD_SET_ALL_CONTEXTに設定されていないかぎり、オーバーライドされます。
- MGW_MQ_putMessageOptionsは、WebSphere MQ Base Java Queue.put()メソッドのputMessageOptions引数として使用されています。WebSphere MQのヘッダー情報の一部ではないため、実際のメッセージのプロパティではありません。

WebSphere MQ Base JavaのMQQueueManager.accessQueueメソッドに対するopenOptions引数の値は、DBMS_MGWADM.REGISTER_FOREIGN_QUEUEをコールしてWebSphere MQキューが登録されるときに指定されます。これらの値の間には依存関係が存在する場合があります。たとえば、MGW_MQ_putMessageOptionsにMQPMD_SET_ALL_CONTEXTを含めるには、MQ_openMessageOptionsキュー・オプションにMQOO_SET_CONTEXTを含める必要があります。

MGWエージェントによって、指定可能なすべての値にMQPMO_SYNCPOINTという値が追加されます。

MGWによって、次の2つのWebSphere MQのメッセージ・ヘッダー・フィールドのデフォルト値が設定されます。messageTypeのデフォルト値はMQMT_DATAGRAMに、putMessageOptionsのデフォルト値はMQPMO_SYNCPOINTに設定されます。

MGWには、Oracle Database Advanced Queuingのメッセージ・プロパティとWebSphere MQのヘッダー・フィールド間の2つのデフォルト・マッピングがあります。

1つは、キュー内でメッセージが使用可能になったときのメッセージのTime-to-Liveを表すOracle Database Advanced Queuingのメッセージ・プロパティのexpirationを、メッセージのTime-to-Liveを表すWebSphere MQのヘッダー・フィールドのexpiryにマップします。発信伝播の場合、expiryに使用する値はキュー内でメッセージが使用可能になった時刻をexpirationから引き算し、10分の1秒単位の値に変換します。Oracle Database Advanced Queuing値NEVERは、MQEI_UNLIMITEDにマップされます。着信伝播の場合、expirationは単に秒単位に変換されたexpiryです。WebSphere MQ値のMQEI_UNLIMITEDは、NEVERにマップされます。

その他のデフォルトは、Oracle Database Advanced Queuingのメッセージ・プロパティのpriorityをWebSphere MQ

のヘッダー・フィールドのpriorityとともにマップします。[表20-2](#)で説明します。

表20-2 伝播のデフォルト優先順位のマッピング

伝播型	メッセージ・システム	優先順位の値										
発信	Oracle Database Advanced Queuing	0	1	2	3	4	5	6	7	8	9	
発信	WebSphere MQ	9	8	7	6	5	4	3	2	1	0	
着信	Oracle Database Advanced Queuing	9	8	7	6	5	4	3	2	1	0	
着信	WebSphere MQ	0	1	2	3	4	5	6	7	8	9	

注意:



発信伝播の場合、0未満のOracle Database Advanced Queuing優先順位は、WebSphere MQ優先順位9にマップされ、9より大きいOracle Database Advanced Queuing優先順位はWebSphere MQ優先順位0にマップされます。

WebSphere MQ発信伝播

発信伝播用のメッセージ変換が指定されていない場合は、Oracle Database Advanced Queuingのソース・キューのペイロード型は、SYS.MGW_BASIC_MSG_TまたはRAWである必要があります。メッセージ変換が指定されている場合、変換のターゲットのADTはSYS.MGW_BASIC_MSG_Tである必要がありますが、ソースのADTはOracle Database Advanced Queuingによってサポートされている任意のADTが有効です。

Oracle Database Advanced QueuingキューのペイロードがRAWの場合、WebSphere MQから取得されるメッセージには、RAWバイトの値に設定されたメッセージ本体と、デフォルトでは「MGW_Byte」値に設定されたformatフィールドがあります。

Oracle Database Advanced Queuingキューのペイロードまたは変換ターゲットのADTがSYS.MGW_BASIC_MSG_Tの場合、メッセージは次のようにWebSphere MQの固有メッセージにマップされます。

- WebSphere MQの固定ヘッダー・フィールドは、[「WebSphere MQメッセージ・ヘッダーのマッピング」](#)で説明されているように、標準型メッセージの内部Oracle Database Advanced Queuingメッセージ・プロパティおよびSYS.MGW_BASIC_MSG_T.header属性に基づいています。
- 標準型メッセージにTEXT本体がある場合は、WebSphere MQのフォーマット・ヘッダー・フィールドは、ヘッダー・プロパティMGW_MQ_formatによってオーバーライドされないかぎりMQFMT_STRINGに設定されます。メッセージ本体はテキストとみなされます。
- 標準型メッセージにRAW本体がある場合は、WebSphere MQのフォーマット・ヘッダー・フィールドは、ヘッダー・プロパティMGW_MQ_formatによってオーバーライドされないかぎりMGW_Byteに設定されます。メッセージ本体はRAWバイトとみなされます。

- 標準型メッセージに、TEXTおよびRAWの両方の本体が含まれる場合、メッセージ変換は失敗します。
- 標準型メッセージに、TEXT本体またはRAW本体のいずれも含まれていない場合、メッセージ本体は設定されず、WebSphere MQフォーマット・ヘッダー・フィールドはMQFMT_NONEとなります。
- 標準型メッセージに小さい値セットと大きい値セット(空ではないSYS.MGW_BASIC_MSG_T.TEXT_BODY.small_valueおよびSYS.MGW_BASIC_MSG_T.TEXT_BODY.large_value)の両方があるTEXT本体が含まれる場合は、メッセージ変換は失敗します。
- 標準型メッセージに小さい値セットと大きい値セット(空ではないSYS.MGW_BASIC_MSG_T.RAW_BODY.small_valueおよびSYS.MGW_BASIC_MSG_T.RAW_BODY.large_value)の両方があるRAW本体が含まれる場合は、メッセージ変換は失敗します。

ジョブ・オプションPreserveMessageIDが値TRUEに指定されている場合、WebSphereメッセージ・ヘッダーのcorrelationIdフィールドはAQのソース・メッセージ識別子に設定されます。correlationIdの値は“AQMSGID:”+AQ_msgidという書式の24バイト値で、AQ_msgidは16バイトのDatabase AQメッセージ識別子を表します。

WebSphere MQ着信伝播

着信伝播用のメッセージ変換が指定されていない場合は、Oracle Database Advanced Queuingの宛先キューのペイロード型は、SYS.MGW_BASIC_MSG_TまたはRAWである必要があります。メッセージ変換が指定されている場合、変換のソースのADTはSYS.MGW_BASIC_MSG_Tである必要がありますが、宛先のADTはOracle Database Advanced Queuingによってサポートされている任意のユーザー定義型が有効です。

Oracle Database Advanced QueuingキューのペイロードがRAWで、着信WebSphere MQメッセージにMQFMT_STRINGのformatが含まれる場合、メッセージ変換は失敗します。そうでない場合は、メッセージ本体はRAWバイトとみなされ、宛先キューに直接エンキューされます。バイト数が32KBを超える場合は、メッセージ変換は失敗します。実際の限度は、32767バイトではなく32512バイトです。

Oracle Database Advanced Queuingキューのペイロードまたは変換ソースのADTがSYS.MGW_BASIC_MSG_Tの場合、WebSphere MQメッセージは次のようにSYS.MGW_BASIC_MSG_Tメッセージにマップされます。

- 特定のWebSphere MQのヘッダー・フィールドは、前に説明したようにOracle Database Advanced Queuingのメッセージ・プロパティにマップされます。
- 標準型メッセージのSYS.MGW_BASIC_MSG_T.header属性は、[表20-1](#)に示すとおり、WebSphere MQヘッダー・フィールドに基づいて{名前, 値}の組に設定されます。これらの値は、WebSphere MQのメッセージ・ヘッダーの当初の内容を保持します。
- WebSphere MQのformatヘッダー・フィールドがMQFMT_STRINGの場合、WebSphere MQのメッセージ本体はテキストとして処理され、その値はSYS.MGW_BASIC_MSG_T.text_bodyにマップされます。他のformat値の場合、メッセージ本体はRAWバイトとして処理され、その値はSYS.MGW_BASIC_MSG_T.raw_bodyにマップされます。

関連項目:

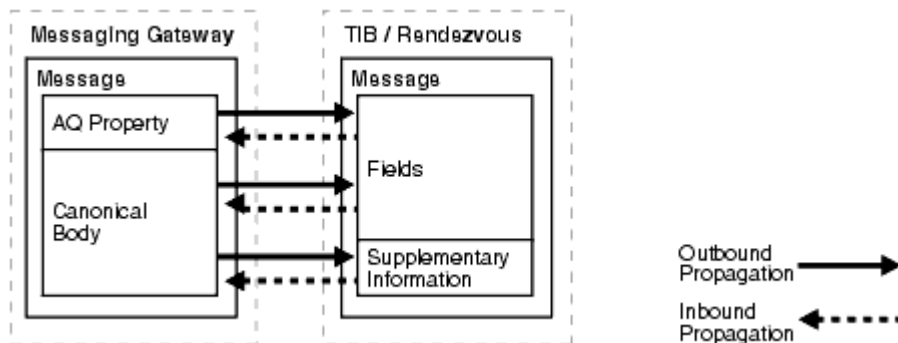
[WebSphere MQメッセージ・ヘッダーのマッピング](#)

TIB/Rendezvousに対するメッセージ変換

MGWでは、TIB/Rendezvousメッセージはフィールドのセットおよび追加情報とみなされます。[図20-4](#)は、MGWと

TIB/Rendezvousの間におけるメッセージの変換方法を示したものです。

図20-4 TIB/Rendezvousに対するメッセージ変換



メッセージ変換障害が発生すると、メッセージはMGWによって、ソース・キュー内の残りのメッセージの伝播を続行できるように例外キュー(例外キューがある場合)に移されます。TIB/Rendezvousからの着信伝播では、例外キューは登録済サブジェクトです。

TIB/RendezvousのフィールドのTIB/Rendezvousのワイヤ・フォーマットのデータ型は、符号のない整数およびネストされたメッセージ型以外はすべてサポートされています。今回のリリースでは、ユーザー定義のカスタム・データ型はサポートされていません。メッセージにサポートされていないデータ型のデータが含まれる場合は、処理時にメッセージ変換障害が発生します。メッセージ変換障害が発生すると、例外キューがある場合は、失敗したメッセージはソース・キューから例外キューに移されます。

表20-3は、MGWがTIB/Rendezvousの固有メッセージと標準のユーザー定義型とで変換を行う場合に使用されるデータ型マッピングを示したものです。サポートされているTIB/Rendezvousの各ワイヤ・フォーマット型に対して、データの格納に使用されるOracleタイプおよびその型を表すDBMS_MGWMSG定数が示されています。

表20-3 TIB/Rendezvousデータ型のマッピング

TIB/Rendezvousワイヤ・フォーマット	Oracle型	ADTフィールド型
Bool	NUMBER	TIBRVMSG_BOOL
F32	NUMBER	TIBRVMSG_F32
F64	NUMBER	TIBRVMSG_F64
I8	NUMBER	TIBRVMSG_I8
I16	NUMBER	TIBRVMSG_I16
I32	NUMBER	TIBRVMSG_I32
I64	NUMBER	TIBRVMSG_I64
U8	サポートされません	サポートされません
U16	サポートされません	サポートされません

TIB/Rendezvousワイヤフォーマット	Oracle型	ADTフィールド型
U32	サポートされません	サポートされません
U64	サポートされません	サポートされません
IPADDR32	VARCHAR2	TIBRVMSG_IPADDR32
IPPORT16	NUMBER	TIBRVMSG_IPPORT16
DATETIME	DATE	TIBRVMSG_DATETIME
F32ARRAY	SYS. MGW_NUMBER_ARRAY_T	TIBRVMSG_F32ARRAY
F64ARRAY	SYS. MGW_NUMBER_ARRAY_T	TIBRVMSG_F64ARRAY
I8ARRAY	SYS. MGW_NUMBER_ARRAY_T	TIBRVMSG_I8ARRAY
I16ARRAY	SYS. MGW_NUMBER_ARRAY_T	TIBRVMSG_I16ARRAY
I32ARRAY	SYS. MGW_NUMBER_ARRAY_T	TIBRVMSG_I32ARRAY
I64ARRAY	SYS. MGW_NUMBER_ARRAY_T	TIBRVMSG_I64ARRAY
U8ARRAY	サポートされません	サポートされません
U16ARRAY	サポートされません	サポートされません
U32ARRAY	サポートされません	サポートされません
U64ARRAY	サポートされません	サポートされません
MSG	サポートされません	サポートされません
OPAQUE	RAW または BLOB	TIBRVMSG_OPAQUE
STRING	VARCHAR2 または CLOB	TIBRVMSG_STRING
XML	RAW または BLOB	TIBRVMSG_XML

Oracle Database Advanced QueuingおよびTIB/Rendezvous間の伝播の場合、MGWは、Oracle Database Advanced Queuingのペイロード型RAWおよびSYS. MGW_TIBRV_MSG_Tを直接サポートします。その他のOracle Database Advanced Queuingのペイロード型をサポートするには、変換を指定する必要があります。

TIB/Rendezvousに対するAQメッセージ・プロパティ・マッピング

この項では、Oracle Database AQのメッセージ・プロパティとTIB/Rendezvousのフィールドの間のマッピングについて説明します。このマッピングを使用して、発信伝播中にはDatabase AQメッセージ・プロパティが保持され、着信伝播中にはDatabase AQメッセージ・プロパティが設定されます。

表20-4は、TIB/Rendezvousのフィールドを使用してサポートされているDatabase AQメッセージ・プロパティを示したものです。第1列は、Database AQメッセージ・プロパティのDBMS_AQ.MESSAGE_PROPERTIES_Tフィールドを示します。第2列と第3列は、TIB/Rendezvousのフィールドに使用される名前とデータ型を示します。最後の列は、メッセージ・プロパティが着信伝播と発信伝播に対してサポートされているかどうかを示します。

表20-4 Oracle Database Advanced Queuingメッセージ・プロパティに対するTIB/RendezvousおよびMGWの名前

Oracle Database Advanced Queuingメッセージ・ プロパティ	MGWの名前	TIB/Rendezvousワイ ヤ・フォーマット・データ型	使用対象
priority	MGW_AQ_priority	TibrvMsg. I32	発信、着信
expiration	MGW_AQ_expiration	TibrvMsg. I32	発信、着信
delay	MGW_AQ_delay	TibrvMsg. I32	発信、着信
correlation	MGW_AQ_correlation	TibrvMsg. STRING	発信、着信
exception_queue	MGW_AQ_exception_queue	TibrvMsg. STRING	発信、着信
enqueue_time	MGW_AQ_enqueue_time	TibrvMsg. DATETIME	発信
original_msgid	MGW_AQ_original_msgid	TibrvMsg. OPAQUE	発信
msgid (1)	MGW_AQ_messageID	TibrvMsg. OPAQUE	発信

表20-4に関する注意:

1. Database AQプロパティmsgidは、DBMS_AQ.MESSAGE_PROPERTIES_Tレコードの特定のフィールドではなくDatabase AQメッセージ識別子を表します。

TIB/Rendezvous発信伝播

発信伝播に対する伝播変換が指定されていない場合は、Oracle Database Advanced Queuingのソース・キューのペイロード型は、SYS.MGW_TIBRV_MSG_TまたはRAWである必要があります。伝播変換が指定されている場合、変換のターゲットのADTはSYS.MGW_TIBRV_MSG_Tである必要がありますが、ソースのADTはOracle Database Advanced Queuingによってサポートされている任意のADTが有効です。

Oracle Database Advanced Queuingキューのペイロードまたは変換ターゲットのADTがSYS.MGW_TIBRV_MSG_Tの場合は、次のようになります。

- ソース・メッセージのすべてのフィールドは、TIB/Rendezvousから取得されるメッセージのTIB/Rendezvousメッセージ・フィールドに変換されます。
- reply_subject属性がNULLではない場合は、応答サブジェクトの追加情報が設定されます。
- send_subjectフィールドは無視されます。

Oracle Database Advanced QueuingキューのペイロードがRAWの場合は、次のようになります。

- 結果メッセージには、値がTibrvMsg. OPAQUEでMGW_RAW_MSGという名前のフィールドが含まれます。フィールドIDは0に設定されます。

ジョブ・オプションAQ_MsgPropertiesが値TRUEに指定されている場合は、MGWエージェントによって、TIB/Rendezvousメッセージの各Database AQメッセージ・プロパティを保持するフィールドが表20-4に従って生成されます。

ジョブ・オプションPreserveMessageIDが値TRUEに指定されている場合、Database AQメッセージID(msgid)は表20-4に従ってTIB/Rendezvousメッセージに保持されます。

TIB/Rendezvous着信伝播

着信伝播に対する伝播変換が指定されていない場合、Oracle Database Advanced Queuingの宛先キューのペイロード型はRAWまたはSYS. MGW_TIBRV_MSG_Tである必要があります。伝播変換が指定されている場合、変換ターゲットのADTにはOracle Database Advanced Queuingでサポートされている任意のADTが有効ですが、変換ソースのADTはSYS. MGW_TIBRV_MSG_Tである必要があります。

Oracle Database Advanced Queuingキューのペイロードまたは変換ソースのADTがSYS. MGW_TIBRV_MSG_Tの場合は、次のようになります。

- TIB/Rendezvousのソース・メッセージの個々のフィールドは、結果としてSYS. MGW_TIBRV_MSG_T型のメッセージのフィールドに変換されます。
- MGWエージェントによって、TIB/Rendezvousのソース・メッセージから送信サブジェクト名を抽出し、SYS. MGW_TIBRV_MSG_Tにsend_subject属性を設定します。通常、送信サブジェクト名は登録済の伝播ソース・キューのサブジェクト名と同じですが、ワイルドカードを使用する場合は異なる場合があります。
- MGWエージェントによって、TIB/Rendezvousのソース・メッセージから応答サブジェクト名を抽出し、SYS. MGW_TIBRV_MSG_Tにreply_subject属性を設定します。
- TIB/Rendezvousのソース・メッセージに3つを超える大きなテキスト・フィールド(4000バイトを超えるテキスト)または3つを超える大きなバイト・フィールド(2000バイトを超える)が含まれる場合は、メッセージ変換は失敗します。

Oracle Database Advanced QueuingキューのペイロードがRAWの場合は、次のようになります。

- TIB/Rendezvousのソース・メッセージにTibrvMsg. OPAQUE型またはTibrvMsg. XML型のMGW_RAW_MSGという名前のフィールドが含まれる場合、Oracle Database Advanced Queuingのメッセージ・ペイロードはフィールド・データです。フィールド名とIDは無視されます。該当するフィールドがない、またはフィールドの型が予想したものと異なる場合は、メッセージ変換障害が発生します。
- RAWデータ・サイズが32KBを超える場合は、メッセージ変換障害が発生します。これは、バインド変数に許可されたデータ・サイズ制限によるものです。また、実際の限度は32767ではなく32512です。

ジョブ・オプションAQ_MsgPropertiesが値TRUEに指定されている場合は、MGWエージェントによって、TIB/Rendezvousの当初のメッセージの中から予約したフィールド名のフィールドが検索されます。表20-4は、フィールド名文字列とTIB/Rendezvousのメッセージで使用される対応する値を示したものです。

該当するフィールドがある場合は、MGWエージェントでは、デフォルト値ではなくフィールド値を使用して対応するOracle

Database Advanced Queuingのメッセージ・プロパティが設定されます。同じ名前のフィールドが複数ある場合は、最初のフィールドのみが使用されます。Oracle Database Advanced Queuingのキュー・ペイロードがRAWの場合のみ、フィールドが結果のペイロードから削除されます。予約名のあるフィールドのデータ型が予想したものと異なる場合は、メッセージ変換障害が発生します。

関連項目:

データ型の値は、『[PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス](#)』のDBMS_MGWMSGに関する項を参照してください。

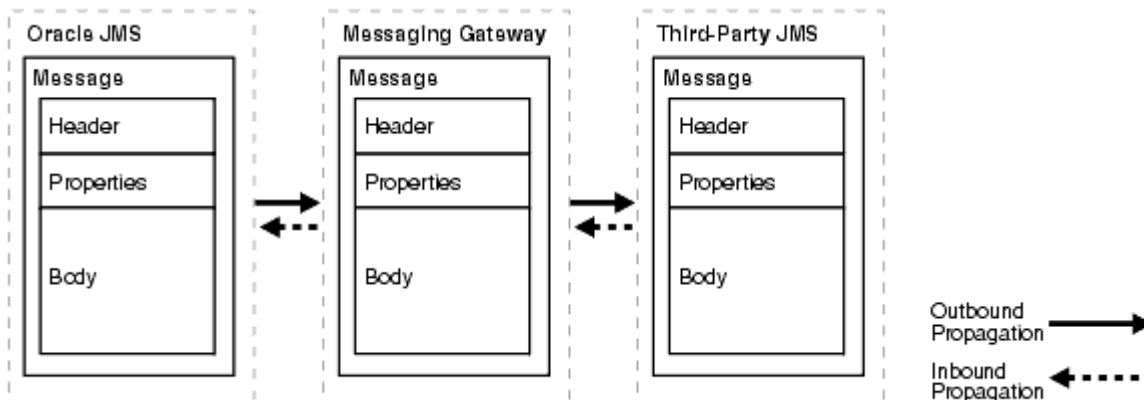
JMSメッセージ

MGWではメッセージの内容を変更することなく、Oracle JMSシステムとOracle JMS以外のシステム間でJMSメッセージのみを伝播します。[図20-5](#)は、JMSメッセージ伝播を示したものです。

MGWでは、標準のJMSメッセージ・タイプのみサポートしています。次のものはサポートしていません。

- JMSプロバイダの拡張機能。これらの拡張機能は宛先JMSシステムで認識されないためです。このようなJMS以外のメッセージの伝播を試みると、エラー・メッセージが表示されます。
- JMS伝播用のユーザー変換。
- 論理変更レコード(LCR)の伝播。

図20-5 JMSメッセージの伝播



この説明のために、JMSメッセージは5つのJMSメッセージ・インタフェースの1つを実装するクラスのJavaオブジェクトとします。[表20-5](#)は、JMSメッセージ・インタフェースと対応するOracle JMSのユーザー定義型を示したものです。また表には、5つの特定の型のいずれかであるインタフェースの`javax.jms.Message`、および一般的なOracle JMS型の`SYS.AQ$_JMS_MESSAGE`も示します。

表20-5 Oracle JMSメッセージの変換

JMSメッセージ	ADT
<code>javax.jms.TextMessage</code>	<code>SYS.AQ\$_JMS_TEXT_MESSAGE</code>
<code>javax.jms.BytesMessage</code>	<code>SYS.AQ\$_JMS_BYTES_MESSAGE</code>

JMSメッセージ	ADT
javax.jms.MapMessage	SYS.AQ\$_JMS_MAP_MESSAGE
javax.jms.StreamMessage	SYS.AQ\$_JMS_STREAM_MESSAGE
javax.jms.ObjectMessage	SYS.AQ\$_JMS_OBJECT_MESSAGE
javax.jms.Message	SYS.AQ\$_JMS_MESSAGE

伝播ジョブを有効にするとMGWエージェントによって、伝播ソースまたは宛先に対するOracle Database Advanced Queuingのペイロード型がチェックされます。型が、[表20-5](#)に示した型またはANYDATAである場合は、メッセージ伝播が実行されます。そうでない場合は、例外が記録され伝播は実行されません。

MGWエージェントが、OracleMGW_OriginalMessageIDというJMS Stringプロパティを宛先キューに送信されたJMSメッセージに追加する場合があります。これは、ソース・メッセージの元のメッセージ識別子を保持するためです。このプロパティは、PreserveMessageIDジョブ・オプションがTRUEという値で指定された場合に追加されます。また、メッセージの変換が失敗した際に例外キューに移動されるすべてのメッセージにも追加されます。

JMS発信伝播

Oracle Database Advanced Queuingキューからメッセージをデキューすると、Oracle JMSによって[表20-5](#)に示すADTのインスタンスがJMSメッセージに変換されます。さらに、ANYDATAのインスタンスを内容に応じてJMSメッセージに変換できます。

ペイロード型ANYDATAのキューは、JMSメッセージにマップされないメッセージを保持できます。MGWではメッセージのデキューはできません。エラーが記録され、そのキューからのメッセージの伝播はメッセージが削除されるまで続行されません。

JMS着信伝播

WebSphere MQ JMSを使用して正常にデキューされるメッセージは、すべてJMSメッセージです。Oracle JMSを使用してエンキューする前にメッセージを変換する必要はありません。ただし、伝播宛先のペイロードのユーザー定義型が着信メッセージの型を受け入れない場合は例外が記録され、メッセージが例外キュー内に配置されます。このような型の不一致の例には、JMS TextMessageとキュー・ペイロード型SYS.AQ\$_JMS_BYTES_MESSAGEなどがあります。

21 Oracle Messaging Gatewayの監視

次のトピックでは、Oracle Messaging Gateway (MGW)のエージェントの監視方法、ユーザーが経験する異常状態、メッセージ・ゲートウェイのエラーと例外に関する複数ソースの情報および推奨される改善措置を説明します。

- [Oracle Messaging Gatewayのログ・ファイル](#)
- [Oracle Messaging Gatewayのエージェント・ステータスの監視](#)
- [Oracle Messaging Gatewayの伝播の監視](#)
- [Oracle Messaging Gatewayのエージェントのエラー・メッセージ](#)

Oracle Messaging Gatewayのログ・ファイル

メッセージ・ゲートウェイ・エージェントの状態、履歴およびエラーは、メッセージ・ゲートウェイのログ・ファイルに記録されます。メッセージ・ゲートウェイ・エージェントが起動されるたびに、異なるログ・ファイルが作成されます。あらゆるエラー、起動時に読み込まれた設定情報または動的設定情報はログ・ファイルに書き込まれているため、ログ・ファイルを監視する必要があります。

デフォルトのエージェントのログ・ファイル名のフォーマットは次のとおりです。

```
oramgw-hostname-timestamp-processid. log
```

指定エージェントのログ・ファイル名のフォーマットは次のとおりです。

```
oramgw-AGENTNAME-hostname-timestamp-processid. log
```

デフォルトで、メッセージ・ゲートウェイのログ・ファイルは、*ORACLE_HOME*/mgw/logにあります。この場所は、エージェントで使用されるメッセージ・ゲートウェイの初期化ファイル(通常、mgw.ora)内のlog_directoryパラメータで上書きできます。

この項の内容は次のとおりです。

- [Oracle Messaging Gatewayのログ・ファイル・サンプル](#)
- [Oracle Messaging Gatewayのログ・ファイルにおける例外メッセージの解釈](#)

Oracle Messaging Gatewayのログ・ファイル・サンプル

次のサンプル・ログ・ファイルは、メッセージ・ゲートウェイ・エージェントの起動を示しています。サンプル・ログ・ファイルには、リンク、登録された外部キュー、サブスクライバおよびスケジュールが追加されたことが表示されます。また、伝播ジョブがアクティブ化されていることも示されています。最後の行は、メッセージ・ゲートウェイが起動され、実行中であり、メッセージを伝播する準備が完了していることを示しています。

例21-1 メッセージ・ゲートウェイのサンプル・ログ・ファイル

```
>>2007-01-16 15:04:49 MGW C-Bootstrap 0 LOG process-id=11080
Bootstrap program starting
>>2007-01-16 15:04:50 MGW C-Bootstrap 0 LOG process-id=11080
JVM created -- heapsize = 64
>>2007-01-16 15:04:53 MGW Engine 0 200 main
MGW Agent version: 11.1.0.0
>>2007-01-16 15:04:53 MGW AdminMgr 0 LOG main
Connecting to database using connect string = jdbc:oracle:oci:@INST1
>>2007-01-16 15:05:00 MGW Engine 0 200 main
MGW Component version: 11.1.0.3.0
>>2007-01-16 15:05:01 MGW Engine 0 200 main
```



```

MGW agent name: DEFAULT_AGENT, MGW job instance id: 273006EC6ED255F1E040578C6D021A8C, MGW database
instance: 1
>>2007-01-16 15:05:09 MGW Engine 0 1 main
Agent is initializing.
>>2007-01-16 15:05:09 MGW Engine 0 23 main
The number of worker threads is set to 1.
>>2007-01-16 15:05:09 MGW Engine 0 22 main
The default polling interval is set to 5000ms.
>>2007-01-16 15:05:09 MGW MQD 0 LOG main
Creating MQSeries messaging link:
link : MQLINK
link type : Base Java interface
queue manager : my.queue.manager
channel : channel1
host : my.machine
port : 1414
user :
ccdt url :
ssl cipherSuite :
connections : 1
inbound logQ : logq1
outbound logQ : logq2
>>2007-01-16 15:05:09 MGW Engine 0 4 main
Link MQLINK has been added.
>>2007-01-16 15:05:09 MGW Engine 0 7 main
Queue DESTQ@MQLINK has been registered; provider queue: MGWUSER.MYQUEUE.
>>2007-01-16 15:05:09 MGW Engine 0 9 main
Propagation Schedule JOB_AQ2MQ (MGWUSER.MGW_BASIC_SRC --> DESTQ@MQLINK) has been
added.
>>2007-01-16 15:05:09 MGW AQN 0 LOG main
Creating AQ messaging link:
link : oracleMgwAq
link type : native
database : INST1
user : MGWAGENT
connection type : JDBC OCI
connections : 1
inbound logQ : SYS.MGW_RECV_LOG
outbound logQ : SYS.MGW_SEND_LOG
>>2007-01-16 15:05:10 MGW Engine 0 19 main
MGW propagation job JOB_AQ2MQ has been activated.
>>2007-01-16 15:05:10 MGW Engine 0 14 main
MGW propagation job JOB_AQ2MQ (MGWUSER.MGW_BASIC_SRC --> DESTQ@MQLINK) has been added.
>>2007-01-16 15:05:11 MGW Engine 0 2 main
Agent is up and running.

```

Oracle Messaging Gatewayのログ・ファイルにおける例外メッセージの解釈

メッセージ・ゲートウェイのログ・ファイルに記録された例外メッセージには、ログ・ファイル内で [Linked-exception] と特定された 1つ以上のリンクされた例外が含まれる場合があります。これが問題の原因を特定するための最も便利な方法となります。たとえば、リンクされた例外には、Oracleエラー・メッセージ、PL/SQLスタック・トレースまたはその両方を含む java.sql.SQLException があります。

次の例は、無効な値 (bad_service_name) が DBMS_MGWADM.CREATE_AGENT または DBMS_MGWADM.ALTER_AGENT の database パラメータに指定された場合の、メッセージ・ゲートウェイのログ・ファイルからのエントリを示したものです。この場合、メッセージ・ゲートウェイ・エージェントではデータベース接続を確立できません。

例21-2 例外メッセージ・サンプル

```
>>2003-07-22 15:27:26 MGW AdminMgr 0 LOG main
Connecting to database using connect string = jdbc:oracle:oci8:@BAD_SERVICE_NAME
>>2003-07-22 15:27:29 MGW Engine 0 EXCEPTION main
oracle.mgw.admin.MgwAdminException: [241] Failed to connect to database. SQL
error: 12154, connect string: jdbc:oracle:oci8:@BAD_SERVICE_NAME
[ ...Java stack trace here... ]
[Linked-exception]
java.sql.SQLException: ORA-12154: TNS:could not resolve the connect identifier
specified
[ ...Java stack trace here... ]
>>2003-07-22 15:27:29 MGW Engine 0 25 main
Agent is shutting down.
```

Oracle Messaging Gatewayのエージェント・ステータスの監視

この項の内容は次のとおりです。

- [MGW_GATEWAYビュー](#)
- [Oracle Messaging Gatewayのリカバリ不能なエラー・メッセージ](#)
- [その他のOracle Messaging Gatewayのエラー状態](#)

MGW_GATEWAYビュー

MGW_GATEWAYビューでは、メッセージ・ゲートウェイ・エージェントの進捗を監視します。エージェントの監視に使用できるフィールドは、次のとおりです。

- AGENT_NAME
- AGENT_INSTANCE
- AGENT_PING
- AGENT_STATUS
- LAST_ERROR_MSG
- SERVICE

AGENT_STATUSフィールドには、エージェントのステータスが表示されます。この列に有効な値は、次のとおりです。

NOT_STARTED

エージェントが実行されず、実行されるスケジュールもないことを示します。

START_SCHEDULED

エージェント・ジョブがジョブ・スケジューラによって実行されるのを待機中であることを示します。

STARTING

エージェントが起動プロセス中であることを示します。

INITIALIZING

エージェントが起動済で設定データを読み込み中であることを示します。

RUNNING

エージェントで、使用可能メッセージの伝播または動的な設定変更のプロセスの準備が完了していることを示します。

SHUTTING_DOWN

エージェントが停止プロセス中であることを示します。

BROKEN

エージェントのプロセスの起動時に、メッセージ・ゲートウェイが他のエージェントが実行中であることを検出したことを示します。通常の使用状態でこのような状況が発生することはありません。

AGENT_PINGフィールドを問い合わせると、メッセージ・ゲートウェイ・エージェントがpingされます。この値は、REACHABLEまたはUNREACHABLEのいずれかになります。ステータスがRUNNINGのエージェントは、常にREACHABLEである必要があります。

LAST_ERROR_MSG、LAST_ERROR_DATEおよびLAST_ERROR_TIMEの列には、メッセージ・ゲートウェイ・エージェントの起動時または実行時にエラーが発生した場合に有効な情報が表示されます。AGENT_INSTANCEは、メッセージ・ゲートウェイのインスタンスが起動されたOracle Databaseのインスタンスを示します。

関連項目:

MGW_GATEWAYビューの詳細は、[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』](#)のDBMS_MGWADMに関する項を参照してください。

Oracle Messaging Gatewayのリカバリ不能なエラー・メッセージ

MGW_GATEWAYビューのAGENT_STATUSフィールドにあるNOT_STARTEDのステータスは、メッセージ・ゲートウェイ・エージェントが実行中でないことを示します。AGENT_STATUSがNOT_STARTEDで、LAST_ERROR_MSGフィールドがNULLでない場合は、メッセージ・ゲートウェイ・エージェントが起動中または実行中にリカバリ不能なエラーが発生したことを示します。メッセージ・ゲートウェイのログ・ファイルが生成済であり、なんらかのエラーが表示されていないかどうかを確認します。ログ・ファイルが存在しない場合、メッセージ・ゲートウェイ・エージェントのプロセスは起動されなかった可能性があります。

この項では、MGW_GATEWAYビューのLAST_ERROR_MSGフィールドに表示されるエラー・メッセージの原因と解決方法を説明します。特に他で記述がないかぎり、メッセージ・ゲートウェイ・エージェントでは、これらのエラーが発生した場合に、自動的に再起動することはありません。

ORA-01089: 即時シャットダウン処理中 - 操作はできません。

エージェントを実行中のインスタンスで、実行中のOracleデータベース・インスタンスを停止するSHUTDOWN IMMEDIATEコマンドが使用されたため、メッセージ・ゲートウェイ・エージェントは停止されました。エージェントは、実行が設定されている次の使用可能データベース・インスタンスで自動的に再起動されます。

ORA-06520: PL/SQL: 外部ライブラリのロード中にエラーが発生しました。

共有ライブラリがロードされていないため、メッセージ・ゲートウェイ・エージェントのプロセスを起動できませんでした。Java共有ライブラリがライブラリ・パスにない可能性があります。listener.oraのライブラリ・パスが正しく設定されていることを確認してください。

ORA-28575: 外部プロシージャ・エージェントへのRPC接続をオープンできません。

メッセージ・ゲートウェイ・エージェントが起動されませんでした。自動的に再起動されます。

考えられる原因は次のとおりです。

- リスナーが実行されていません。listener.oraを変更した場合は、リスナーを停止してから再起動して変更を有効にする必要があります。
- tnsnames.ora、listener.oraまたはその両方の値が正しくありません。

特に、tnsnames.oraには、MGW_AGENTというネット・サービス名エントリが必要です。このエントリは、Windowsのメッセージ・ゲートウェイには必要ありません。tnsnames.oraのMGW_AGENTネット・サービス名のCONNECT_DATAに指定されているSID値は、listener.oraのSID_DESCエントリのSID_NAME値と一致する必要があります。MGW_AGENTネット・サービス名がプロセス間通信(IPC)接続用に設定されている場合、tnsnames.oraとlistener.oraのADDRESSのKEY値が一致する必要があります。sqlnet.oraのnames.default_domainパラメータを使用してデフォルト・ドメインを設定した場合は、そのドメインをtnsnames.oraのネット・サービス名MGW_AGENTに追加する必要があります。

ORA-28576: 外部プロシージャ・エージェントへのRPC接続が失われました。

メッセージ・ゲートウェイ・エージェントのプロセスが強制終了しました。外部エンティティによってプロセスが終了したか、または内部エラーによって障害が発生した可能性があります。エージェントは自動的に再起動されます。メッセージ・ゲートウェイのログ・ファイルを確認し、詳細情報がないかを調べてください。問題が解決しない場合は、Oracleサポート・サービスに連絡してください。

ORA-32830: 結果コード-2がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイ・エージェントが初期化ファイル(通常、mgw.ora)の読み込みを試行した場合、エラーが発生します。ファイルが読み込み可能であることを確認します。

ORA-32830: 結果コード-3がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイのログ・ファイル作成時にエラーが発生しました。ログ・ディレクトリが書き込み可能であることを確認してください。デフォルトの位置は、ORACLE_HOME/mgw/logです。

ORA-32830: 結果コード-8がMessaging Gatewayエージェントによって戻されました。

Java Virtual Machine(JVM)起動時に、エラーが発生しました。次のことを確認してください。

- 正しいJavaバージョンを使用していること
- オペレーティング・システムのバージョンおよびパッチ・レベルがJDKのバージョンに適合していること
- JVMヒープ・サイズに適切な値が設定されていること
ヒープ・サイズがDBMS_MGWADM.ALTER_AGENTのmax_memoryパラメータで指定されていること
- Windowsプラットフォームの場合は、mgw.oraに設定されているMGW_PRE_PATHに適切なJVMライブラリ(jvm.dll)へのパスが含まれていること

ORA-32830: 結果コード-12がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイのログ・ファイル作成時にエラーが発生しました。ファイルのI/O問題を発生させる可能性のある、ディスクの空き領域またはその他の問題がないかどうか確認します。

ORA-32830: 結果コード-17がMessaging Gatewayエージェントによって戻されました。

JVMは正常に作成されましたが、メッセージ・ゲートウェイJavaエージェント・プログラムのコール試行中にエラーが発生しました。mgw. oraで設定されているCLASSPATHが適切であることを確認します。

ORA-32830: 結果コード-19がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイ・エージェントは特定の初期化ファイルを使用するように構成されていますが、そのファイルは存在しません。管理者によって指定されているファイルのフルパス名は、MGW_GATEWAYビューのINITFILEフィールドに表示されます。その初期化ファイルを作成するか、DBMS_MGWADM.ALTER_AGENTを使用して、INITFILEを他のファイルに設定するか、またはデフォルトの初期化ファイルを使用するようにNULLを設定します。

ORA-32830: 結果コード-100がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイ・エージェントJVMで、ログ・ファイルに書き込む前に、起動時にランタイム例外またはエラーが発生しました。

ORA-32830: 結果コード-101がMessaging Gatewayエージェントによって戻されました。

回復不能なエラーにより、メッセージ・ゲートウェイ・エージェントが停止されました。メッセージ・ゲートウェイのログ・ファイルで詳細を確認してください。mgw. oraで指定されている値が適切であることを確認します。値が不適切な場合、メッセージ・ゲートウェイ・エージェントは異常なエラー状態のため終了することがあります。

ORA-32830: 結果コード-102がMessaging Gatewayエージェントによって戻されました。

ファイルORACLE_HOME/mgw/jlib/mgw.jarのバージョンとメッセージ・ゲートウェイのPL/SQLパッケージのバージョンが一致しないため、メッセージ・ゲートウェイ・エージェントが停止されました。すべてのメッセージ・ゲートウェイのコンポーネントのリリース番号が同じであることを確認してください。

ORA-32830: 結果コード-103がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイ・エージェントを実行中のデータベース・インスタンスが停止されたため、エージェントが停止されました。エージェントは、別のインスタンス上(そのように設定した場合)、または停止したインスタンスを再起動したとき、自動的に再起動されます。

ORA-32830: 結果コード-104がMessaging Gatewayエージェントによって戻されました。

前のエラーを参照してください。

ORA-32830: 結果コード-105がMessaging Gatewayエージェントによって戻されました。

メッセージ・ゲートウェイ・エージェントが、実行不可のときに実行中であることが検出されました。これは起りえないことです。このようなことが発生すると、AGENT_STATUSはBROKENとなりエージェントは自動的に停止します。このエラーが発生する場合は、次のようにします。

- 実行中のメッセージ・ゲートウェイ・エージェントのプロセスをすべて終了します。プロセス名は、通常 extprocmgwextprocです。
- DBMS_MGWADM.CLEANUP_GATEWAY (DBMS_MGWADM.CLEAN_STARTUP_STATE) を実行します。
- DBMS_MGWADM.STARTUPを使用してメッセージ・ゲートウェイ・エージェントを起動します。

ORA-32830: 結果コード-106がMessaging Gatewayエージェントによって戻されました。

前のエラーを参照してください。

関連項目:

[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』のDBMS_MGWADMに関する項](#)

その他のOracle Messaging Gatewayのエラー状態

この項では、長時間にわたりMGW_GATEWAYビューにSTART_SCHEDULEDで残留しているAGENT_STATUSの原因を説明します。

データベース・サービスが起動していない

メッセージ・ゲートウェイでは、Oracle Schedulerジョブを使用して、メッセージ・ゲートウェイ・エージェントを起動します。Oracle Schedulerを使用して、どのジョブでデータベース・サービスを実行するか(サービス・アフィニティ)を指定できます。管理者はメッセージ・ゲートウェイで、メッセージ・ゲートウェイ・エージェントを、そのエージェントに関連付けられているSchedulerジョブ・クラスを構成するときに使用されるデータベース・サービスで構成できます。

データベースを停止すると、そのデータベースへのサービスがすべて停止されます。データベースの起動時、それらのサービスを手動で再起動する必要がある場合があります。Schedulerジョブがサービスに関連付けられている場合、そのサービスが開始されるまでジョブは実行されません。メッセージ・ゲートウェイ・エージェントのAGENT_STATUSが、長時間にわたりSTART_SCHEDULEDのままである場合、データベース・サービスが無効化されているか、またはそのサービスに関連付けられているデータベース・インスタンスがいずれも実行中でない可能性があります。MGW_GATEWAYビュー、Oracle Schedulerビューおよびサービス・ビューを使用して、エージェントの構成方法、Schedulerジョブとデータベース・サービスの現在の状態を判断できます。

注意:



Messaging Gateway で使用される Oracle Scheduler オブジェクトの詳細は、[「Oracle Messaging Gateway のエージェントのスケジューラのジョブ」](#)を参照してください。

不足しているジョブ・キュー・プロセス

メッセージ・ゲートウェイでは、Oracle Schedulerを使用して、メッセージ・ゲートウェイの外部プロセスを起動します。AGENT_STATUSがSTART_SCHEDULEDの場合、メッセージ・ゲートウェイ・エージェントのSchedulerジョブはスケジューラによって実行されるのを待機します。メッセージ・ゲートウェイ・ジョブは、使用可能なジョブ・プロセスが現れるまで実行されません。メッセージ・ゲートウェイでは、Schedulerジョブ・プロセスはメッセージ・ゲートウェイ・エージェント・セッションの存続期間中、保持されます。複数のメッセージ・ゲートウェイ・エージェントが起動された場合、エージェントはそれぞれ独自のSchedulerジョブを使用し、独自のジョブ・プロセスを必要とします。

JOB_QUEUE_PROCESSESデータベース初期化パラメータの値がゼロの場合、Oracle Schedulerジョブは実行されません。値がゼロ以外の場合、事実上、この値は同時に実行できるSchedulerジョブおよびジョブ・キューのジョブの最大数です。

メッセージ・ゲートウェイのステータスが長時間START_SCHEDULEDのままである場合、JOB_QUEUE_PROCESSESがゼロまたは低すぎる値ですべてのジョブ・スレーブがビジーの状態データベースが起動された可能性があります。値にゼロ以外が設定され、十分な数のジョブ・キュー・プロセスがあり、各メッセージ・ゲートウェイ・エージェントでプロセスが1つずつ使用可能な状態でデータベース・インスタンスが起動されていることを確認してください。

破損または無効化されたSchedulerジョブ

メッセージ・ゲートウェイ・エージェントに関連付けられたOracle Schedulerジョブがなんらかの理由により無効化または破損した場合、メッセージ・ゲートウェイ・エージェントのステータスはSTART_SCHEDULEDのままになります。そのような場合であるかを判定するには、DBA_SCHEDULER_JOBSビューを使用して、エージェントのSchedulerジョブのSTATEフィールドを確認します。通常、Schedulerジョブのステータスは、メッセージ・ゲートウェイ・エージェントのSchedulerジョブが実行されるのを待機している場合、SCHEDULEDまたはメッセージ・ゲートウェイ・エージェントが実行中の場合、RUNNINGです。メッセージ・ゲートウェイ・エージェントのステータスがNOT_STARTEDの場合、エージェントのSchedulerジョブは存在しません。

DBA_SCHEDULER_JOB_RUN_DETAILSなどのその他のSchedulerビューで、メッセージ・ゲートウェイのSchedulerジョブの追加

情報を確認してください。また、問題を示すエラー・メッセージがないかどうか、MGW_GATEWAYビューおよびメッセージ・ゲートウェイのログ・ファイルも確認してください。

注意:



Messaging Gateway で使用される Oracle Scheduler オブジェクトの詳細は、[「Oracle Messaging Gatewayのエージェントのスケジューラのジョブ」](#)を参照してください。

Oracle Real Application Clusters(Oracle RAC)環境

メッセージ・ゲートウェイをOracle RAC環境で使用し、データベース・サービスでエージェントが構成されているが、サービスが使用可能なデータベース・インスタンスが実行されていない場合、実行中のデータベース・インスタンス上でサービスが起動されるまで、メッセージ・ゲートウェイAGENT_STATUSはSTART_SCHEDULEDのままです。

Oracle Messaging Gatewayの伝播の監視

メッセージ・ゲートウェイ伝播は、MGW_JOBSビューおよびメッセージ・ゲートウェイのログ・ファイルを使用して監視できます。ビューでは、伝播済メッセージに関する情報および伝播試行中に発生したエラーに関する情報が提供されます。ログ・ファイルを使用して、エラーの原因を判定できます。

MGW_JOBSビューには、構成情報の表示の他に、[メッセージ](#)伝播の監視に使用できる動的情報も表示されます。適用可能フィールドは、STATUS、ENABLED、PROPAGATED_MSGS、EXCEPTIONQ_MSGS、FAILURES、LAST_ERROR_MSG、LAST_ERROR_DATEおよびLAST_ERROR_TIMEです。

STATUSフィールドには、ジョブの現在のステータスが表示されます。READYは、伝播のジョブが準備完了であることを表します(ただし、ENABLEDフィールドがTRUEの場合のみ)。RETRYは、伝播障害が発生したが、伝播が再試行されることを表します。FAILEDは、リカバリ不可能なエラーが発生または伝播障害に達したため、エージェントがジョブの伝播を停止したことを表します。DELETE_PENDINGは、DBMS_MGWADM.REMOVE_JOBがコールされたが、このジョブに関連する特定のクリーン・アップ・タスクが未完了で、ジョブの削除が保留中であることを表します。SUBSCRIBER_DELETE_PENDINGは、DBMS_MGWADM.REMOVE_SUBSCRIBERが古い形式のジョブでコールされたが、このジョブに関連する特定のクリーン・アップ・タスクが未完了で、ジョブの削除が保留中であることを表します。

ENABLEDフィールドには、伝播ジョブが現在有効化されているかどうかが表示されます。TRUEは、ジョブが有効であることを、FALSEはジョブが無効であることを表します。ジョブが有効化されていないと、いずれの伝播も発生しません。

MGW_JOBSビューのPROPAGATED_MSGSフィールドには、正常に伝播されたメッセージの数が表示されます。メッセージ・ゲートウェイ・エージェントが起動すると、このフィールドはゼロにリセットされます。

メッセージ・ゲートウェイのジョブに[例外キュー](#)が構成されている場合、メッセージ変換障害のためメッセージ・ゲートウェイ・エージェントで最初の伝播障害が発生した時点で、メッセージ・ゲートウェイ・エージェントによってメッセージがその例外キューに移されます。メッセージ変換障害は、メッセージ・ゲートウェイのログ・ファイルでoracle.mgw.common.MessageExceptionによって示されます。EXCEPTIONQ_MSGSフィールドには、例外キューに移されたメッセージの数が表示されます。メッセージ・ゲートウェイ・エージェントが起動すると、このフィールドはゼロにリセットされます。

伝播ジョブに対してメッセージ伝播中にエラーが発生すると、FAILURESフィールドの値が増分されます。このフィールドは、最後にメッセージが正常に伝播されてからの障害回数を示します。障害が発生するたびに、LAST_ERROR_MSG、LAST_ERROR_DATEおよびLAST_ERROR_TIMEによって、エラー・メッセージとその発生時刻が表示されます。障害の発生回数が16件になると、メッセージ・ゲートウェイはこの伝播ジョブに対する伝播試行を停止します。伝播試行を再開するには、伝播ジョブに対して

DBMS_MGWADM.RESET_JOBをコールする必要があります。

エラーが発生した場合、詳細はメッセージ・ゲートウェイのログ・ファイルを調べてください。

関連項目:

[『Oracle Database PL/SQLパッケージ・プロシージャおよびタイプ・リファレンス』のDBMS_MGWADMに関する項](#)

Oracle Messaging Gatewayのエージェントのエラー・メッセージ

この項では、MGW_JOBSビューのLAST_ERROR_MSG列に表示され、メッセージ・ゲートウェイ・エージェントのログ・ファイルに記録される、特に頻繁に発生する一部のエラーを示します。また、特別なアクションを必要とするエラーも示されています。エラーの発生に気付いたら、ログ・ファイルでリンクされている例外を参照して、問題の原因を判断します。

次の2つの主なエラー・タイプが、メッセージ・ゲートウェイ・エージェントのログ・ファイルに記録されています。

- oracle.mgw.common.MessageException

このエラー・タイプは、メッセージ変換障害が発生すると記録されます。メッセージ・ゲートウェイ・エージェントでは、障害の原因となったメッセージを伝播できない可能性があり、この伝播ジョブは最終的に停止されます。

- oracle.mgw.common.GatewayException

このエラー・タイプは、メッセージ変換以外の障害が発生すると記録されます。原因によっては、問題が自動的に修復されたり、またはユーザー・アクションが必要になる場合があります。

[221] <messaging_system>キューへのアクセスに失敗しました: <queue>

Oracle Database Advanced QueuingキューまたはOracle以外のキューにアクセスしたとき、エラーが発生しました。ログ・ファイルにあるリンクされた例外エラー・コードとメッセージを調べてください。

[241] データベースへの接続に失敗しましたSQLエラー: <error>、接続文字列: <connect_string>

これは、DBMS_MGWADM.CREATE_AGENTまたはDBMS_MGWADM.ALTER_AGENTで指定されたMGW接続情報が不適切であった可能性があります。メッセージ・ゲートウェイ・エージェントのユーザーまたはパスワードが不適切か、またはデータベース指定子(データベース・パラメータ)のいずれかが不適切です。エージェント、JDBC OCIまたはJDBC Thinによって使用される接続タイプに関して、接続情報が適切であることを確認します。

databaseパラメータがNULLの場合は、メッセージ・ゲートウェイのログ・ファイルで次のOracleリンク・エラーを調べてください。

```
ORA-01034: ORACLE not available
ORA-27101: shared memory realm does not exist
```

これらの2つのエラーはどちらも、メッセージ・ゲートウェイ・エージェントがローカルIPC接続を使用してデータベースへの接続を試みましたが、ORACLE_SID値が不適切であることを示しています。

データベース・パラメータがNULLに設定されている場合、ローカル接続が使用されます。ローカルで接続する必要がある場合、メッセージ・ゲートウェイ・エージェントのプロセスに正しいORACLE_SID値を設定する必要があります。この値を設定するには、MGW初期化ファイル(通常、mgw.ora)に次の行を追加します。

```
set ORACLE_SID = sid_value
```

データベースがNULLでない場合、MGW初期化ファイルでORACLE_SIDを設定する必要はありません。

MGW初期化ファイルでORACLE_SIDを設定しても機能しない場合は、データベース・パラメータをNULL以外の値に設定する必

必要があります。

JDBC Thin接続が使用されている場合は、データベース・パラメータをNULL以外の値に設定する必要があります。JDBC Thin接続が使用されていて、データベース・パラメータがTNSNamesの別名である場合、`oracle.net.tns_names` JavaプロパティがMGW初期化ファイルで設定されていることを確認します。このプロパティを設定するには、MGW初期化ファイルに次の行を追加します。

```
setJavaProp oracle.net.tns_admin=<directory containing tnsnames.ora>
```

注意:



詳細は、[\[oracle.net.tns_admin\]](#)を参照してください。

[415] ジョブ<job_name>のソース・キューからのメッセージの欠落

考えられる原因は次のとおりです。

- エージェントがメッセージ・ゲートウェイ・エージェント以外によってデキューされた永続メッセージの一部を処理しました。
- 伝播ソース・キューがページまたは再作成されました。
- メッセージがOracle Database Advanced Queuingの例外キューに移されました。

このエラーが発生した場合は、次のように、DBMS_MGWADMパッケージのプロシージャCLEANUP_GATEWAYをコールしてください。

```
DBMS_MGWADM.CLEANUP_GATEWAY (  
    action => DBMS_MGWADM.RESET_SUB_MISSING_MESSAGE,  
    sarg => <job_name>);
```

このコールは、伝播ジョブでメッセージが欠落する問題が発生した場合で、エージェントが実行中の場合のみ有効になります。エージェントは、欠落メッセージを**非永続**メッセージとみなし、伝播ジョブ処理を続行します。

関連項目:

メッセージ・ゲートウェイ例外キューの詳細は、[「伝播ジョブの概要」](#)を参照してください。

[416] ジョブ<job_name>の受信ログ・キューのログ・レコードの欠落

考えられる原因は次のとおりです。

- ログ記録が、メッセージ・ゲートウェイ・エージェント以外によってログ・キューからデキューされました。
- ログ・キューがページまたは再作成されました。

このエラーが発生した場合は、次のように、DBMS_MGWADMパッケージのプロシージャCLEANUP_GATEWAYをコールしてください。

```
DBMS_MGWADM.CLEANUP_GATEWAY (  
    action => DBMS_MGWADM.RESET_SUB_MISSING_LOG_REC,  
    sarg => <job_name>);
```

このコールは、伝播ジョブでログ記録が欠落する問題が発生した場合で、エージェントが実行中の場合のみ有効になります。

注意:



欠落メッセージが宛先キューに伝播済の場合、プロセス DBMS_MGWADM.CLEANUP_GATEWAY をコールするとメッセージが重複する可能性があります。ソースおよび宛先キューの両方にあるメッセージをチェックする必要があります。このようなメッセージがある場合は、ソースまたは宛先キューのいずれかからメッセージを削除してから、このプロセスをコールする必要があります。

[417] ジョブ<job_name>の送信ログ・キューのログ・レコードの欠落

前のエラーを参照してください。

[421] 警告: ジョブ<job_name>のリカバリのための接続を取得できません

これは、他の伝播ジョブによって接続がすべて使用中であるため、メッセージ・ゲートウェイ・エージェントが伝播ジョブを回復するための接続を取得できなかったことを示す警告メッセージです。エージェントは接続を取得できるまで試行します。

WebSphere MQのリンクに対してこのメッセージが何回も繰り返される場合は、伝播ジョブに関連付けられたメッセージ・ゲートウェイのリンクが使用する最大接続数を増やしてください。

関連項目:

[メッセージ・システム・リンクの変更](#)

[434] キュー<queue>、プロバイダ・キュー<queue>へのアクセスに失敗しました

このメッセージは、メッセージ・システムの固有キューにアクセスできないことを示します。キューが DBMS_MGWADM.REGISTER_FOREIGN_QUEUEによって登録されている、またはOracle Database Advanced Queuing キューである可能性があります。詳細は、リンクされた例外を参照してください。

考えられる原因は次のとおりです。

- 外部キーの登録が正しくない、またはメッセージ・ゲートウェイのリンク設定が正しくありません。
構成情報を確認してください。可能な場合は、同じ構成情報を使用してOracle以外のメッセージ・システムのサンプル・アプリケーションを実行してください。
- Oracle以外のメッセージ・システムにアクセスできません。
Oracle以外のメッセージ・システムが実行中で、メッセージ・ゲートウェイのリンクで提供されている情報を使用してアクセス可能であることを確認してください。
- Oracle Database Advanced Queuingキューが存在しません。メッセージ・ゲートウェイの伝播ジョブの作成後、キューが削除された可能性があります。

Oracle Database Advanced Queuingキューが存在することを確認してください。

[436] LOW MEMORY WARNING: 合計メモリ = < >、free_mem = < >

メッセージ・ゲートウェイ・エージェントJVMの実行用メモリが不足しています。Javaガベージ・コレクションが起動されますが、JVMヒープ・サイズが小さすぎると表示される可能性があります。DBMS_MGWADM.ALTER_AGENTのmax_memoryパラメータを使用して、JVMヒープ・サイズを増やしてください。メッセージ・ゲートウェイ・エージェントを実行中の場合は、再起動して変更を有効にする必要があります。

[703] 変換<transformation_id>の情報の取得に失敗しました

メッセージ・ゲートウェイ・エージェントが、[変換](#)に関する必要な情報をすべて取得できませんでした。DBMS_MGWADM.CREATE_JOBの変換パラメータでは、変換ファンクションではなく登録済の変換の名前を指定する必要があります。

考えられる原因は次のとおりです。

- 変換が存在しません。変換が作成済であることを確認します。これは、ユーザーSYSとして実行される次の問合せで確認できます。

```
SELECT TRANSFORMATION_ID, OWNER FROM DBA_TRANSFORMATIONS;
```

- メッセージ・ゲートウェイに登録された変換が間違っています。登録済の変換が間違っていないことを確認します。
- メッセージ・ゲートウェイ・エージェント・ユーザーに、例外で示された変換のfrom_typeまたはto_typeに使用される[オブジェクト型](#)に対するEXECUTE権限がありません。

MGW_AGENT_ROLEにEXECUTE権限を付与し、その後エージェントのユーザーにMGW_AGENT_ROLEを付与するだけでは不十分です。オブジェクト型に対するEXECUTE権限をエージェントのユーザーに直接付与するか、またはPUBLICに付与する必要があります。

次の例は、from_typeでのそのような場合を示しています。また、エラーの厳密な原因を判断するために使用されているリンクされた例外も示されています。

オブジェクト型に対するEXECUTE権限がない場合

```
Errors occurred during processing of job JOB_AQ2MQ_2
oracle.mgw.common.GatewayException: [703] Failed to retrieve information for
transformation mgwuser.SAMPLEADT_TO_MGW_BASIC_MSG
[...Java stack trace here...]
[Linked-exception]
java.sql.SQLException: "from_type" is null
[...Java stack trace here...]
```

[720] AQペイロード型<type>はサポートされていません。キュー: <queue>

メッセージ・ゲートウェイの伝播ジョブによって使用されるOracle Database Advanced Queuingキューのペイロード型がメッセージ・ゲートウェイによって直接サポートされていません。JMS以外の伝播の場合、メッセージ・ゲートウェイではペイロード型のRAW、SYS.MGW_BASIC_MSG_TおよびSYS.MGW_TIBRV_MSG_Tを直接サポートしています。

考えられるアクションは次のとおりです。

- メッセージ・ゲートウェイの伝播ジョブをキュー・ペイロード型からサポートされている型へ変換するように構成します。
- メッセージ・ゲートウェイの伝播ジョブを削除し、サポートされているペイロード型のOracle Database Advanced Queuingキューを使用する新規ジョブを作成します。

[Java Message Service](#) (JMS)伝播の場合は、メッセージ・ゲートウェイの伝播ジョブを削除し、[Oracle Java Message Service](#) (Oracle JMS)によってそのOracle Database Advanced Queuingペイロード型がサポートされている新規ジョブを作成する必要があります。JMS伝播に対する変換はサポートされていません。

[721] 変換タイプ<type>はサポートされていません。キュー: <queue_name>、変換: <transformation>

メッセージ・ゲートウェイの伝播ジョブが、[標準型](#)の1つではないオブジェクト型を使用する変換によって構成されています。

発信ジョブの場合、変換from_typeはOracle Database Advanced Queuingペイロード型、変換to_typeはメッセージ・ゲートウェイ標準型である必要があります。着信ジョブの場合、変換from_typeはメッセージ・ゲートウェイ標準型、変換to_typeはOracle Database Advanced Queuingペイロード型である必要があります。

[722] メッセージの変換に失敗しました。キュー: <queue_name>、変換: <transformation>

変換実行中にエラーが発生しました。通常、ORA-25229は、変換ファンクションによってPL/SQL例外が発生したとき、または変換を使用しようとしたときにその他のOracleエラーが発生した場合に、Oracle Database Advanced Queuingによって表示されます。

考えられる原因は次のとおりです。

- メッセージ・ゲートウェイ・エージェントのユーザーに変換ファンクションに対するEXECUTE権限がありません。これは、次の例に示されています。

MGW_AGENT_ROLEにEXECUTE権限を付与し、その後メッセージ・ゲートウェイ・エージェントのユーザーにMGW_AGENT_ROLEを付与するだけでは不十分です。変換ファンクションに対するEXECUTE権限を、メッセージ・ゲートウェイ・エージェントのユーザーに直接付与するか、またはPUBLICに付与する必要があります。

変換ファンクションに対するEXECUTE権限がない場合

```
Errors occurred during processing of job JOB_MQ2AQ_2
oracle.mgw.common.GatewayException: [722] Message transformation failed queue:
MGWUSER.DESTQ_SIMPLEADT, transform: MGWUSER.MGW_BASIC_MSG_TO_SIMPLEADT
[...Java stack trace here...]
[Linked-exception]
oracle.mgw.common.MessageException: [722] Message transformation failed;
queue: MGWUSER.DESTQ_SIMPLEADT, transform:
MGWUSER.MGW_BASIC_MSG_TO_SIMPLEADT
[...Java stack trace here...]
[Linked-exception]
java.sql.SQLException: ORA-25229: error on transformation of message msgid:
9749DB80C85B0BD4E03408002086745E
ORA-00604: error occurred at recursive SQL level 1
ORA-00904: invalid column name
[...Java stack trace here...]
```

- 登録済の変換が存在しても、変換ファンクションが存在しません。変換ファンクションが存在しない場合は、再作成する必要があります。
- メッセージ・ゲートウェイ・エージェントのユーザーには、例外で示されたキューのペイロード・オブジェクト型に対するEXECUTE権限がありません。
MGW_AGENT_ROLEにEXECUTE権限を付与し、その後メッセージ・ゲートウェイ・エージェントのユーザーにMGW_AGENT_ROLEを付与するだけでは不十分です。オブジェクト型に対するEXECUTE権限を、メッセージ・ゲートウェイ・エージェントのユーザーに直接付与するか、またはPUBLICに付与する必要があります。
- 変換ファンクションでエラーが発生しました。変換ファンクションが受信するすべてのメッセージを処理できること確認してください。

[724] メッセージの変換がサポートされていません。宛先AQペイロード型: <type>、送信元タイプ: <type>

メッセージ・ゲートウェイの伝播ジョブは、Oracle以外のメッセージ・システム・リンクによって生成される標準メッセージ型がOracle Database Advanced Queuingキュー・ペイロード型と互換性がない着信伝播用に構成されています。SYS.MGW_BASIC_MSG_Tペイロード型によるTIB/Rendezvousメッセージ・システムからOracle Database Advanced Queuingキューへの伝播、またはWebSphere MQからSYS.MGW_TIBRV_MSG_Tペイロード型を使用したOracle Database Advanced Queuingへの伝播です。

考えられるアクションは次のとおりです。

- Oracle以外のメッセージ・リンクによって生成される標準メッセージ型をOracle Database Advanced Queuing

ペイロード型にマップする変換によって、メッセージ・ゲートウェイの伝播ジョブを構成します。

- メッセージ・ゲートウェイの伝播ジョブを削除し、Oracle Database Advanced Queuingキュー・ペイロード型が、Oracle以外のリンクによって生成される標準メッセージ型と一致する新規ジョブを作成します。

[725] テキスト・メッセージがRAWペイロードをサポートしていません。

メッセージ・ゲートウェイの伝播ジョブが、RAWペイロード型のOracle Database Advanced Queuing宛先への着信伝播に対して構成されています。Oracle以外のソース・キューから受信したテキスト・メッセージのメッセージ変換に失敗しました。

テキスト・データに対するサポートが必要な場合は、メッセージ・ゲートウェイの伝播ジョブを削除し、ペイロード型がテキスト・データをサポートしているOracle Database Advanced Queuing宛先に対する新規ジョブを作成します。

[726] メッセージ・サイズ<size>はRAWペイロードには大きすぎます。最大サイズは<size>です

メッセージ・ゲートウェイの伝播ジョブが、RAWペイロード型のOracle Database Advanced Queuing宛先への着信伝播に対して構成されています。Oracle以外のソース・キューからの大きなRAW値を含むメッセージを受信時に、メッセージ変換障害が発生しました。

大容量データに対するサポートが必要な場合は、メッセージ・ゲートウェイの伝播ジョブを削除し、ペイロード型が大容量データ(通常、[BLOB](#)属性のオブジェクト型フォームのデータ)をサポートしているOracle Database Advanced Queuing宛先に対する新規ジョブを作成します。

[728] メッセージに含まれる(BLOB)フィールドが多すぎます。

BLOB型に格納される必要のあるフィールドが、ソース・メッセージに多すぎます。SYS.MGW_TIBRV_MSG_Tは、3つのBLOBフィールドに制限されています。小さなフィールドに分割するか、または少数の大きなフィールドに結合して、メッセージ内の大きなフィールドの数を減らします。

[729] メッセージに含まれる(CLOB)フィールドが多すぎます。

ソース・メッセージに、CLOBに格納される必要のある大きなテキスト値を含むフィールドが多すぎます。SYS.MGW_TIBRV_MSG_Tは、3つのCLOBフィールドに制限されています。小さなフィールドに分割するか、または少数の大きなフィールドに結合して、メッセージ内の大きなフィールドの数を減らします。

[805] キュー<queue>へのエンキュー中にMQSeries Messageエラーが発生しました

WebSphere MQキューにメッセージを格納したとき、WebSphere MQによってエラーが戻されました。ログ・ファイルにあるリンクされた例外エラー・コードとメッセージを調べてください。WebSphere MQのドキュメントを参照してください。

A 非永続キュー

Oracle Database Advanced Queuingでは、サブスクライバに非永続メッセージを非同期に配信できます。これらのメッセージはイベント・ドリブンの可能性があり、システム(またはインスタンス)に障害が発生すると保持されません。メッセージは、システムによって作成されたキュー表に格納されます。Oracle Database Advanced Queuingは、1つの共通APIで永続および非永続メッセージをサポートします。

非永続キューは、シングル・コンシューマまたはマルチ・コンシューマのいずれかで、現在接続されているすべてのユーザーへの通知のメカニズムを提供します。サブスクライバをマルチ・コンシューマ非永続キューに追加して、非永続キューを伝播の宛先にすることができます。

ユーザーは、エンキュー・インタフェースを使用して、通常の方法で非永続キューにメッセージをエンキューします。非永続キューに、RAW型およびOracleオブジェクト型のメッセージをエンキューできます。OCI通知は、現在通知登録されているユーザーに、そのようなメッセージを配信するために使用されます。

次のトピックでは、非永続的キューについて説明します(これは、Oracle Database Advanced Queuing 10gリリース2 (10.2)では推奨されていません)。かわりにバッファ済メッセージを使用することをお勧めします。

関連項目:

[バッファ済メッセージ](#)

トピック:

- [非永続キューの作成](#)
- [非永続キューの管理](#)
- [非永続キューの互換性](#)
- [非永続キュー通知](#)
- [非永続キューの制限](#)

非永続キューの作成

```
DBMS_AQADM.CREATE_NP_QUEUE (  
  queue_name          IN          VARCHAR2,  
  multiple_consumers IN          BOOLEAN DEFAULT FALSE,  
  comment             IN          VARCHAR2 DEFAULT NULL);
```

このプロシージャは、[非永続](#)キューを作成します。

非永続キューでは、ローカル受信者のみがサポートされています。このキューは、シングル・コンシューマまたはマルチ・コンシューマのいずれかです。すべてのキュー名はスキーマ内で重複しないようにしてください。このキューは、キュー名によって指定されたものと同じスキーマ内にある、システムが作成した8.1互換のキュー表(AQ\$_MEM_SCまたはAQ\$_MEM_MC)に作成されます。キュー名にスキーマ名が指定されていないときは、ログイン・ユーザーのスキーマにキューが作成されます。



注意:

非永続キューの名前は、24 文字より長くできません。長い名前の非永続キューを作成しようとすると、エラー ORA-24019 が発生します。

非永続キューの管理

キューは、CREATE_NP_QUEUEで作成した後、START_QUEUEをコールすると有効になります。デフォルトでは、キューはエンキューおよびデキューともに使用禁止で作成されます。

非永続キューに、RAW型およびOracleオブジェクト型のメッセージをエンキューできます。非永続キューからはデキューできません。非永続キューからメッセージを取り出す唯一の方法は、[Oracle Call Interface\(OCI\)](#)の通知メカニズムを使用する方法です。非永続キューにはlistenをコールできません。

非永続キューを削除できるのは、所有者のみです。

非永続キューの互換性

8.1互換以上のキューで、非永続キューを使用するには、init.oraのcompatibleパラメータおよび[キュー表](#)のcompatibleパラメータを8.1以上に設定する必要があります。

非永続キュー通知

非永続キューの場合、メッセージは通知の一部として配信されます。RAW表現が指定された場合に実行される[非永続キュー](#)のアクションを、通知メカニズム別で[表A-1](#)に示します。XML表現が指定されたときに実行されるキューのアクションを、[表A-2](#)に示します。

表A-1 RAW表現が指定された場合に実行される非永続キューのアクション

キュー・ペイロード型	OCICコールバック	電子メール	PL/SQLコールバック
RAW	OCI コールバックは、ペイロードの RAW データを受信します。	サポートされません	PL/SQL コールバックは、ペイロードの RAW データを受信しません。
Oracle オブジェクト型	サポートされません	サポートされません	サポートされません

表A-2 XML表現が指定された場合に実行される非永続キューのアクション

キュー・ペイロード型	OCICコールバック	電子メール	PL/SQLコールバック
RAW	OCI コールバックは、ペイロードの XML データを受信します。	XML データが SOAP メッセージとしてフォーマットされ、登録されている電子メール・アドレスに送信されます。	PL/SQL コールバックは、ペイロードの XML データを受信しません。
Oracle オブジェクト型	OCI コールバックは、ペイロードの XML データを受信します。	XML データが SOAP メッセージとしてフォーマットされ、登録されている電子メール・アドレスに送信されます。	PL/SQL コールバックは、ペイロードの XML データを受信しません。

非永続キューの制限

RAWおよびOracleオブジェクト型の非永続的キューを作成できます。メッセージの送信は、サブスクライバおよびローカルである明示的に指定された受信者に制限されます。非永続的キューからの伝播はサポートされていません。メッセージを取り出す際、デキュー・コールは使用できませんが、かわりにOCISubscriptionRegisterを使用して通知の登録を行い、非同期通知メカニズムを採用する必要があります。

非永続メッセージの場合、enqueue_optionsのvisibility属性をIMMEDIATEに設定する必要があります。

関連項目:

[「エンキュー・オプション」](#)

B Oracle JMSおよびOracle AQ XMLサブプレットのエラー・メッセージ

問題のトラブルシューティングに役立つエラー・メッセージのリストを提供します。

- [Oracle JMSのエラー・メッセージ](#)
- [Oracle AQ XMLサブプレットのエラー・メッセージ](#)

Oracle JMSのエラー・メッセージ

JMS-101 無効な送達モードです: (文字列)

原因: 送達モードがサポートされていません。

処置: 有効な送達モードは AQjmsConstants.PERSISTENT です。

JMS-102 機能: (文字列)はサポートされません。

原因: この機能はこのリリースでサポートされていません。

処置: 説明は省略します。

JMS-104 メッセージ・ペイロードの指定が必要です。

原因: メッセージ・ペイロードが NULL でした。

処置: メッセージに NULL 以外のペイロードを指定してください。

JMS-105 エージェントの指定が必要です

原因: AQjmsAgent オブジェクトが NULL でした。

処置: リモート・サブスクリバを表す有効な AQjmsAgent を指定してください。

JMS-106 1 つの JMSConnection で複数のオープン・セッションを保持することはできません。

原因: コネクションにすでに 1 つのオープンしている JMS セッションがあります。1 つのコネクションで複数のオープン・セッションを保持することはできません。

処置: オープン・セッションをクローズし、新しいセッションをオープンしてください。

JMS-107 (文字列)での操作は許可されていません。

原因: 指定された操作がこのオブジェクトで許可されていません。

処置: 説明は省略します。

JMS-108 型: (文字列) のメッセージは、型: (文字列)のペイロードを含む宛先に対しては使用できません。

原因: 使用されているメッセージ型と宛先に指定されたペイロード型が一致しませんでした。

処置: この宛先を含むキュー表に指定されたペイロードにマップするメッセージ型を使用してください。

JMS-109 クラスが見つかりません: (文字列)

原因: 指定されたクラスが見つかりませんでした。

処置: CLASSPATH にクラスが含まれていることを確認してください。

JMS-110 プロパティ: (文字列)は書込み不可です。

原因: 読取り専用メッセージのヘッダー・フィールドまたはプロパティを更新しようとした。

処置: 説明は省略します。

JMS-111 接続の指定が必要です。

原因: 接続オブジェクトが NULL でした。

処置: NULL 以外の JDBC 接続を指定してください。

JMS-112 接続が無効です。

原因: JDBC 接続が無効です。

処置: NULL 以外の Oracle JDBC 接続を指定してください。

JMS-113 接続が停止状態です。

原因: 停止状態の接続のメッセージを受信しようとした。

処置: 接続を開始してください。

JMS-114 接続はクローズされています。

原因: クローズしている接続を使用しようとした。

処置: 新しい接続を作成してください。

JMS-115 コンシューマはクローズされています。

原因: クローズしているコンシューマを使用しようとした。

処置: 新しいメッセージ・コンシューマを作成してください。

JMS-116 サブスクライバ名の指定が必要です。

原因: サブスクライバ名が NULL でした。

処置: NULL 以外のサブスクリプション名を指定してください。

JMS-117 変換に失敗しました - 無効なプロパティ型です

原因: 要求された型へのプロパティの変換中にエラーが発生しました。

処置: プロパティのデータ型に対応するメソッドで取得してください。

JMS-119 無効なプロパティ値です。

原因: 指定されたプロパティの値が無効です。

処置: 設定されているプロパティの適切な型の値を使用してください。

JMS-120 デキューに失敗しました。

原因: メッセージの受信中にエラーが発生しました。

処置: 詳細は、JMSEException 内のメッセージとリンクしている SQLException を参照してください。

JMS-121 DestinationProperty の指定が必要です。

原因: キュー/トピックの作成中に NULL の AQjmsDestinationProperty が指定されました。

処置: 宛先に NULL 以外の AQjmsDestinationProperty を指定してください。

JMS-122 内部エラー(文字列)

原因: 内部エラーが発生しました。

処置: サポートに連絡してください。

JMS-123 間隔は(整数)秒以上で設定してください。

原因: 無効な間隔が指定されました。

処置: 30 秒より大きい間隔を指定してください。

JMS-124 無効なデキュー・モードです

原因: 無効なデキュー・モードが指定されました。

処置: 無効なデキュー・モードは、AQConstants.DEQUEUE_BROWSE、AQConstants.DEQUEUE_REMOVE、AQConstants.DEQUEUE_LOCKED、AQConstants.DEQUEUE_REMOVE_NODATA です。

JMS-125 無効なキューが指定されました。

原因: 無効なキュー・オブジェクトが指定されました。

処置: 有効なキュー・ハンドルを指定してください。

JMS-126 無効な項目が指定されました。

原因: 無効なトピック・オブジェクトが指定されました。

処置: 有効なトピック・ハンドルを指定してください。

JMS-127 無効な宛先です

原因: 無効な宛先オブジェクトが指定されました。

処置: 有効な宛先(キュー/トピック)オブジェクトを指定してください。

JMS-128 無効なナビゲーション・モードです

原因: 無効なナビゲーション・モードが指定されました。

処置: 有効なナビゲーション・モードは、AQjmsConstants.NAVIGATION_FIRST_MESSAGE、

AQjmsConstants.NAVIGATION_NEXT_MESSAGE
AQjmsConstants.NAVIGATION_NEXT_TRANSACTION です。

JMS-129 無効なペイロード型です。

原因: 使用されているメッセージ型と宛先に指定されたペイロード型が一致しませんでした。

処置: この宛先を含むキュー表に指定されたペイロードにマップするメッセージ型を使用してください。
ADT メッセージには、適切な CustomDatum ファクトリを使用してメッセージ・コンシューマを作成してください。

JMS-130 JMS キューはマルチ・コンシューマには使用できません。

原因: JMS キューとして AQ マルチ・コンシューマ・キューを取得しようとした。

処置: JMS キューにマルチ・コンシューマは使用できません。

JMS-131 セッションはクローズされています。

原因: クローズしているセッションを使用しようとした。

処置: 新しいセッションをオープンしてください。

JMS-132 プロパティの最大数(整数)を超えました

原因: メッセージのユーザー定義プロパティの最大数を超えました

処置: 説明は省略します。

JMS-133 メッセージの指定が必要です。

原因: 指定されたメッセージが NULL でした。

処置: NULL 以外のメッセージを指定してください。

JMS-134 名前の指定が必要です。

原因: 指定されたキューまたはキュー表名が NULL でした。

処置: NULL 以外の名前を指定してください。

JMS-135 ドライバ(文字列)はサポートされません。

原因: 指定されたドライバがサポートされていません。

処置: 有効なドライバは、OCI8 および Thin です。kprb ドライバを使用して getDefaultConnection() による kprb 接続を取得するには、静的な createTopicConnection および createQueueConnection メソッドを使用してください。

JMS-136 ペイロード・ファクトリは、ADT ペイロードを持つ宛先にのみ指定できます。

原因: ADT ペイロードを含まない宛先のコンシューマに CustomDatumFactory が指定されました。

処置: SYS.AQ\$_JMS_TEXT_MESSAGE、SYS.AQ\$_JMS_BYTES_MESSAGE、SYS.AQ\$_JMS_MAP_MESSAGE、SYS.AQ\$_JMS_OBJECT_MESSAGE、SYS.AQ\$_JMS_STREAM_MESSAGE 型のペイロードを含む宛先に対して、このフィールドを NULL に設定する必要があります。

JMS-137 ペイロード・ファクトリは、ADT ペイロードを持つ宛先に対して指定する必要があります。

原因: ADT ペイロードを含む宛先に CustomDatumFactory が指定されませんでした。

処置: ADT メッセージを含む宛先には、宛先の SQL ADT 型にマップされる Java クラスの CustomDatumFactory を指定する必要があります。

JMS-138 プロシージャはクローズされています。

原因: クローズしているプロデューサを使用しようとしてしました。

処置: 新しいメッセージ・プロデューサを作成してください。

JMS-139 プロパティ名の指定が必要です。

原因: プロパティ名が NULL でした。

処置: NULL 以外のプロパティ名を指定してください。

JMS-140 無効な System プロパティです。

原因: 無効なシステム・プロパティ名が指定されました。

処置: 有効な JMS システム・プロパティのいずれかを指定してください。

JMS-142 JMS 項目はマルチ・コンシューマ対応のキュー表で作成する必要があります。

原因: シングル・コンシューマ・キュー表の JMS トピックを作成しようとした。

処置: マルチ・コンシューマを使用できるキュー表にのみ、JMS トピックを作成できます。

JMS-143 キューの指定が必要です。

原因: NULL のキューが指定されました。

処置: NULL 以外のキューを指定してください。

JMS-144 JMS キューはマルチ・コンシューマ対応のキュー表では作成できません。

原因: マルチ・コンシューマ・キュー表の JMS キューを作成しようとした。

処置: マルチ・コンシューマを使用できないキュー表にのみ、JMS キューを作成できます。

JMS-145 無効な受信者リストです。

原因: 指定された受信者リストが空でした。

処置: 少なくとも 1 つの受信者を含む受信者リストを指定してください。

JMS-146 登録に失敗しました。

原因: TypeMap の型の登録中にエラーが発生しました。

処置: 説明は省略します。

JMS-147 ReplyTo 宛先タイプが無効です

原因: ReplyTo 宛先オブジェクト型が無効です。

処置: ReplyTo 宛先に AQjmsAgent 型を使用する必要があります。

JMS-148 プロパティ名のサイズが制限を超えています

原因: プロパティ名が最大サイズを超えています。

処置: 100 文字未満のプロパティ名を指定してください。

JMS-149 サブスクライバの指定が必要です。

原因: 指定されたサブスクライバが NULL でした。

処置: NULL 以外のサブスクライバを指定してください。

JMS-150 プロパティはサポートされません。

原因: サポートされていないプロパティを使用しようとした。

処置: 説明は省略します。

JMS-151 項目の型は EXCEPTION にできません。

原因: トピックの型は AQjmsConstants.EXCEPTION にできません。

処置: AQjmsConstants.NORMAL 型のトピックを指定してください。

JMS-153 System プロパティの型が無効です。

原因: 指定された値の型が設定されているシステム・プロパティに定義された型と一致しません。

処置: システム・プロパティの設定に正しい型を使用してください。

JMS-154 SEQUENCE DEVIATION の値が無効です。

原因: 順序逸脱が無効です。

処置: 有効な値は、AQEnqueueOption.DEVIATION_BEFORE、AQEnqueueOption.DEVIATION_TOP です。

JMS-155 AQ 例外(文字列)

原因: AQ Java レイヤーでエラーが発生しました。

処置: 詳細は、JMSException 内のメッセージとリンクしている例外を参照してください。

JMS-156 無効なクラス: (文字列)

原因: 指定されたクラスが無効です。

処置: CLASSPATH に指定されたクラスが含まれていることを確認してください。

JMS-157 I/O 例外: (文字列)

原因: I/O 例外です。

処置: 詳細は、JMSEException のメッセージを参照してください。

JMS-158 SQL 例外: (文字列)

原因: SQL 例外です。

処置: 詳細は、リンクしている SQLException 内のメッセージを参照してください。

JMS-159 無効なセレクトク: (文字列)

原因: 指定されたセレクトクが無効か長すぎます。

処置: セレクトクの構文を確認してください。

JMS-160 EOF 例外: (文字列)

原因: バイト・ストリームの読み込み中に EOF 例外が発生しました。

処置: 説明は省略します。

JMS-161 メッセージ形式例外: (文字列)

原因: 指定された型へのストリーム・データの変換中にエラーが発生しました。

処置: ストリームで予期されるデータの型を確認し、適切な読み込みメソッドを使用してください。

JMS-162 メッセージは読取り不可です。

原因: メッセージが書き込み専用モードです。

処置: リセット・メソッドをコールして、メッセージを読取り可能にしてください。

JMS-163 メッセージは書き込み不可です。

原因: メッセージが読取り専用モードです。

処置: clearBody メソッドを使用して、メッセージを書き込み可能にしてください。

JMS-164 そのような要素はありません

原因: 指定された名前の要素がマップ・メッセージで見つかりませんでした。

処置: 説明は省略します。

JMS-165 プロパティ値が最大サイズを超えています

原因: プロパティの値が使用できる最大長を超えました

処置: JMS 定義プロパティの値は、最大 100 を設定できます。ユーザー定義プロパティの値は、最大 2000 を設定できます。

JMS-166 項目の指定が必要です。

原因: 指定されたトピックが NULL でした。

処置: NULL 以外のトピックを指定してください。

JMS-167 ペイロード・ファクトリまたは Sql_data_class の指定が必要です。

原因: ペイロード・ファクトリまたは Sql_data_class がオブジェクト・ペイロードを含むキューに指定されていません。

処置: キューに定義された ADT 型にマップされる Java オブジェクトの CustomDatumFactory または SQLData クラスを指定してください。

JMS-168 ペイロード・ファクトリと sql_data_class の両方は指定できません。

原因: CustomDatumFactory クラスおよび SQLData クラスの両方がデキュー中に指定されました。

処置: キューに定義された ADT 型にマップされる Java オブジェクトの CustomDatumFactory または SQLData クラスを指定してください。

JMS-169 Sql_data_class は NULL にできません。

原因: 指定された SQLData クラスが NULL です。

処置: キューに定義された ADT 型にマップされる SQLData クラスを指定してください。

JMS-171 (文字列)を含んだメッセージが定義されていません。

原因: メッセージのペイロード型が無効です。

処置: キューが定義されて RAW または OBJECT ペイロードが含まれているかどうかを確認し、メッセージの適切なペイロード型を使用してください。

JMS-172 複数のキュー表が問合せに一致しています: (文字列)

原因: 複数のキュー表が問合せと一致します。

処置: 所有者およびキュー表名を指定してください。

JMS-173 キュー表: (文字列) が見つかりません。

原因: 指定されたキュー表が見つかりませんでした。

処置: 有効なキュー表を指定してください。

JMS-174 オブジェクト・ペイロードとともにキューのクラスを指定する必要があります。

¥ndenqueue(deq_option, payload_fact) または dequeue(deq_option, sql_data_cl) を使用してください。

原因: このデキュー・メソッドを使用して、OBJECT ペイロードとともにキューからデキューできません。

処置: dequeue(deq_option, payload_fact) または dequeue(deq_option, sql_data_cl)のいずれかを使用してください。

JMS-175 デキュー・オプションの指定が必要です。

原因: 指定された DequeueOption が NULL です。

処置: NULL 以外のデキュー・オプションを指定してください。

JMS-176 インキュー・オプションの指定が必要です。

原因: 指定された EnqueueOption が NULL です。

処置: NULL 以外のインキュー・オプションを指定してください。

JMS-177 無効なペイロード・タイプ: ロー・ペイロード・キューには dequeue(deq_option)を使用してください。

原因: このメソッドを使用して、RAW ペイロードとともにキューからデキューできません。

処置: dequeue(deq_option)メソッドを使用してください。

JMS-178 無効なキュー名です - (文字列)

原因: 指定されたキュー名が NULL または無効です。

処置: NULL ではないキュー名を指定してください。キュー名をスキーマ名で修飾しないでください。スキーマ名は、所有者パラメータの値として指定する必要があります。

JMS-179 無効なキュー表名です - (文字列)

原因: 指定されたキュー表名が NULL または無効です。

処置: NULL ではないキュー表名を指定してください。キュー表名をスキーマ名で修飾しないでください。スキーマ名は、所有者パラメータの値として指定する必要があります。

JMS-180 無効なキュー・タイプです。

原因: キュー・タイプが無効です。

処置: 無効なタイプは、AQConstants.NORMAL または AQConstants.EXCEPTION です。

JMS-181 wait_time の値が無効です

原因: wait 型の値が無効です。

処置: 待機時間には、AQDequeueOption.WAIT_FOREVER、AQDequeueOption.WAIT_NONE、または 0 より大きい値を使用できます。

JMS-182 問合せで複数のキューが一致しています。

原因: 複数のキューが問合せと一致します。

処置: キューの所有者および名前を両方を指定してください。

JMS-183 AQ ドライバが登録されていません。

原因: AQDriver が登録されていません。

処置: AQ Java ドライバが登録されていることを確認してください。Class.forName("oracle.AQ.AQOracleDriver")を使用してください。

JMS-184 キュー・オブジェクトが無効です。

原因: キュー・オブジェクトが無効です。

処置: 基になる JDBC 接続がクローズしている可能性があります。キュー・ハンドルを再取得してください。

JMS-185 QueueProperty の指定が必要です。

原因: 指定された AQQueueProperty が NULL です。

処置: NULL 以外の AQQueueProperty を指定してください。

JMS-186 QueueTableProperty の指定が必要です。

原因: 指定された QueueTableProperty が NULL です。

処置: NULL 以外の AQQueueTableProperty を指定してください。

JMS-187 キュー表の指定が必要です。

原因: 指定されたキュー表が NULL です。

処置: NULL 以外のキュー表を指定してください。

JMS-188 QueueTable オブジェクトが無効です。

原因: キュー表オブジェクトが無効です。

処置: 基になる JDBC 接続がクローズしている可能性があります。キュー表ハンドルを再取得してください。

JMS-189 バイト配列が小さすぎます。

原因: 指定されたバイト配列が小さすぎて要求されたデータを格納できません。

処置: 要求されたデータの格納に十分なバイト配列を指定してください。または、要求された長さを短くしてください。

JMS-190 キュー: (文字列) が見つかりません。

原因: 指定されたキューが見つかりませんでした。

処置: 有効なキューを指定してください。

JMS-191 sql_data_cl は SQLData インタフェースを実装しているクラスであることが必要です。

原因: 指定されたクラスは、java.sql.SQLData インタフェースをサポートしていません。

処置: 説明は省略します。

JMS-192 無効な可視性値です

原因: 指定された可視性値が無効です。

処置: 有効な値は、AQConstants.VISIBILITY_ONCOMMIT、AQConstants.VISIBILITY_IMMEDIATE です。

JMS-193 JMS キューに RAW 型のペイロードを含めることはできません。

原因: RAW ペイロードとともに JMS キューを作成しようとした。

処置: JMS キュー/トピックに RAW ペイロードを含めることはできません。

JMS-194 セッション・オブジェクトが無効です。

原因: セッション・オブジェクトが無効です。

処置: 基になる JDBC 接続がクローズしている可能性があります。新しいセッションを作成してください。

JMS-195 無効なオブジェクト型: オブジェクトは CustomDatum または SQLData インタフェースを実装する必要があります

原因: 指定されたオブジェクト型が無効です。

処置: CustomDatum または SQLData インタフェースをオブジェクトに実装する必要があります。

JMS-196 1 つの JMS セッションで同じ宛先のオープン QueueBrowser を複数保持することはできません。

原因: このセッションのこのキューにオープンしている QueueBrowser がすでに存在します。

処置: 特定のセッションの同じキューに複数のキュー・ブラウザを使用できません。既存の

QueueBrowser をクローズして、新しい QueueBrowser をオープンしてください。

JMS-197 リモート・サブスクライバに対するエージェント・アドレスの指定が必要です。

原因: リモート・サブスクライバのアドレス・フィールドが NULL です。

処置: アドレス・フィールドにリモート・トピックの完全修飾名を含める必要があります。

JMS-198 無効な操作: セッションに特権メッセージ・リスナーを設定しました。

原因: セッションのメッセージ・リスナーが設定されていた場合に、クライアントがメッセージの受信にメッセージ・コンシューマを使用しようとしてしました。

処置: セッションのメッセージ・リスナーを使用して、メッセージを処理してください。メッセージを受信するためにコンシューマのメソッドは使用しないでください。

JMS-199 登録の通知に失敗しました。

原因: リスナー登録が失敗しました。

処置: 詳細は、リンクしている例外のエラー・メッセージを参照してください。

JMS-200 宛先の指定が必要です。

原因: 宛先が NULL です。

処置: NULL 以外の宛先を指定してください。

JMS-201 すべての受信者を recipient_list に指定する必要があります。

原因: 受信者リストの複数の要素が NULL です。

処置: 受信者リストのすべての AQjmsAgent を指定する必要があります。

JMS-202 メッセージの非同期受信の登録解除に失敗しました。

原因: 非同期受信のデータベースのコンシューマの登録解除中にエラーが発生しました。

処置: 詳細は、リンクしている例外のエラー・メッセージを確認してください。

JMS-203 ペイロード・ファクトリの指定が必要です。

原因: NULL のペイロード・ファクトリが指定されました。

処置: NULL 以外のペイロード・ファクトリを指定してください。

JMS-204 AQ JNI レイヤーでエラーが発生しました

原因: JNI エラーです。

処置: 詳細は、リンクしている例外のエラー・メッセージを確認してください。

JMS-205 ネーミング例外

原因: ネーミング例外です。

処置: 詳細は、リンクしている例外のエラー・メッセージを確認してください。

JMS-207 JMS 例外(文字列)

原因: JMS レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている JMSEException 内のメッセージを参照してください。

JMS-208 XML SQL 例外

原因: XML SQL レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている AQxmlException 内のメッセージを参照してください。

JMS-209 XML SAX 例外

原因: XML SAX レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている AQxmlException 内のメッセージを参照してください。

JMS-210 XML 解析例外

原因: XML パーサー・レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている AQxmlException 内のメッセージを参照してください。

JMS-220 接続できなくなりました

原因: データベースに接続できなくなりました。

処置: コメント: データベース/ネットワーク/マシンにアクセスできないと、このエラーが発生する場合があります。一時的な障害の可能性があります。

JMS-221 接続プールに使用可能な物理データベース接続がありません

原因: 指定された操作を実行するために、OCI 接続プールに使用可能な物理データベース接続がありませんでした。

処置: 後で操作を実行してください。

JMS-222: 無効なペイロード・ファクトリ型です

原因: ペイロード・ファクトリは CustomDatumFactory 型または ORADDataFactory 型である必要があります。

処置: CustomDatumFactory/ORADDataFactory 型のいずれか 1 つを使用してください。

JMS-223: ペイロード・ファクトリは Sys.AnyData ペイロードを持つ宛先に対して NULL である必要があります - かわりにタイプマップを使用します

原因: SYS.ANYDATA ペイロードを含む宛先のコンシューマに対して CustomDatumFactory/ORADDataFactory が指定されました。

処置: このフィールドは、SYS.AnyData 型のペイロードを含む宛先に対して NULL に設定する必要があります。セッションのタイプマップに ORADDataFactory(s)を登録する必要があります。

JMS-224: タイプマップが無効です - Sys.AnyData の宛先からメッセージを受信するには SQLType/OraDataFactory マッピングを移入する必要があります

原因: セッションのタイプマップが NULL または空になっています。 Sys.AnyData 型の宛先の場合、タイプマップはキュー/トピックに格納されている可能性のあるすべての型に対して OraDataFactory オブジェクトを含める必要があります。

処置: AQjmsSession.getTypeMap()メソッドを使用してタイプマップを取得してください。さらに、put()メソッドを使用して SQLType 名とそれに対応する OraDataFactory オブジェクトを登録してください。

JMS-225: JDBC ドライバが無効です - この操作には OCI ドライバを使用する必要があります

原因: Sys.AnyData のキュー/トピックに対してこの JDBC ドライバを使用する操作はサポートされていません。

処置: JDBC OCI ドライバを使用して、Sys.AnyData のキュー/トピックからのメッセージを受信してください。

JMS-226: ヘッダーのみのメッセージには本文がありません

原因: ヘッダーのみのメッセージには本文がないため、本文の消去、変更または問合せを行うことはできません。

処置: 読取り専用メッセージの本文にアクセスしたり、変更しないでください。

JMS-227: トランザクションが未処理の JMS セッションへのコミットは無効です

原因: トランザクションが未処理の JMS セッションでコミット・メソッドを起動しようとしたが、その起動は無効です。

処置: 現在の JMS セッションでのコミット・メソッドの起動を削除してください。

JMS-228: トランザクションが未処理の JMS セッションへのロールバックは無効です

原因: トランザクションが未処理の JMS セッションでロールバック・メソッドを起動しようとしたが、その起動は無効です。

処置: 現在の JMS セッションでのロールバック・メソッドの起動を削除してください。

JMS-229: (文字列)を指定する必要があります。

原因: 指定したパラメータが NULL でした。

処置: パラメータに NULL 以外の値を指定してください。

JMS-230: アクティブな TopicSubscriber の永続サブスクリプションでの不正な操作

原因: 永続サブスクリプションにアクティブな TopicSubscriber がある間に、その永続サブスクリプションをサブスクライブ解除、変更または作成しようとした。

処置: 有効な TopicSubscriber をクローズして、再試行してください。

JMS-231: 一時的な宛先にあるコンシューマは、一時的な宛先を作成した同じ接続/セッションに属す必

要があります。

原因: 指定したパラメータが NULL でした。

処置: パラメータに NULL 以外の値を指定してください。

JMS-232: 無効なユーザー/パスワードが JMS 接続で指定されました。

原因: 無効なユーザー/パスワードが接続に指定されました。

処置: 接続に有効なユーザー/パスワードを指定してください。

JMS-233: 必要なサブスクライバ情報が使用可能ではありません

原因: getDurableSubscriber メソッドで取得され、インメモリー・マップに対応する情報を持たないサブスクライバからサブスクライバ情報を取得しようとした。

処置: このようなサブスクライバからサブスクライバ情報を取得しないでください。

JMS-234: この操作は現行のメッセージング・ドメインでは許可されていません

原因: 現行のメッセージング・ドメインで許可されない操作を実行します。

処置: この操作を現行のメッセージング・ドメインで実行しないでください。

JMS-235: 永続サブスクライバ名を非サブスクライブ・メソッドのトピックにリンクすることはできません。

原因: OJMS では、指定された永続サブスクライバ名をトピックにリンクできません。このため、非サブスクライブ・メソッドが失敗しました。

処置: ユーザーがトピック名を指定できる別の非サブスクライブ・メソッドを使用してください。

JMS-236: OJMS で無効な OCI ハンドルが見つかりました。

原因: OJMS は、JDBC OCI ドライバを使用しているときに無効な OCI ハンドルを検出しました。

処置: OJMS がまだ操作している間に、基礎となる JDBC 接続がクローズされないようにしてください。

JMS-237: メッセージ・リスナーのスレッドを起動できません。

原因: OJMS はメッセージ・リスナーの新しいスレッドを起動できません。

処置: スレッドがユーザー指定のスレッド・プロキシにより起動されているかどうか、スレッド・プロキシ・コードを確認してください。

JMS-238: トランザクションが処理済の JMS セッションへのリカバリは無効です

原因: トランザクションが処理済の JMS セッションでリカバリ・メソッドを起動しようとしたが、その起動は無効です。

処置: トランザクションが処理済の JMS セッションでは、リカバリ・メソッドのかわりにロールバック・メソッドを使用してください。

JMS-239: XA セッションへの(文字列)メソッドのコールは無効です。

原因: XA セッションでコミット・メソッドまたはロールバック・メソッドを起動しようとしたが、その起動は無効です。

処置: JTA を使用して分散トランザクションをコミットまたはロールバックしてください。

JMS-240: 他の操作の後の setClientID のコールは無効です。

原因: この接続で他の操作が行われた後で Connection.setClientID メソッドを不正にコールしようとした。

処置: クライアント ID を設定したあとで、接続上でその他の操作を実行してください。

JMS-241: コンシューマが使用している場合、一時的な宛先の削除は無効です。

原因: 既存のコンシューマがまだ使用している間に、一時的な宛先を不正に削除しようとした。

処置: コンシューマをクローズしてから、一時的な宛先を削除してください。

JMS-242: 即時可視化および 3 フェーズ・エンキュー・プロセスの両方を使用したメッセージのエンキューは無効です。

原因: 即時可視化と 3 フェーズ・エンキュー・プロセスの両方を使用してメッセージを不正にエンキューしようとした。

処置: システム・プロパティ oracle.jms.useTemplobsForEnqueue を終了してください。

JMS-243: トピック(文字列)が見つかりません

原因: 指定したトピックが見つかりません。

処置: 有効なトピックを指定してください。

Oracle AQ XMLサブレットのエラー・メッセージ

JMS-400 宛先名を指定する必要があります

原因: 宛先名に NULL が指定されました。

処置: NULL 以外の宛先名を指定してください。

JMS-402 クラスが見つかりません: {0}

原因: 指定されたクラスが見つかりませんでした。

処置: CLASSPATH にエラー・メッセージで指定されたクラスが含まれていることを確認してください。

JMS-403 I/O 例外{0}

原因: I/O 例外です。

処置: 詳細は、リンクしている AQxmlException 内のメッセージを参照してください。

JMS-404 XML 解析例外

原因: XML パーサー・レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている AQxmlException 内のメッセージを参照してください。

JMS-405 XML SAX 例外

原因: XML SAX レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている AQxmlException 内のメッセージを参照してください。

JMS-406 JMS 例外{0}

原因: JMS レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている JMSEException 内のメッセージを参照してください。

JMS-407 {0}では操作できません

原因: 指定された操作がこのオブジェクトで許可されていません。

処置: 操作を実行するユーザーに必要な権限があることを確認してください。

JMS-408 変換に失敗しました - 無効なプロパティ型です

原因: 要求された型へのプロパティの変換中にエラーが発生しました。

処置: プロパティのデータ型に対応するメソッドで取得してください。

JMS-409 そのような要素はありません

原因: 指定された名前の要素がマップ・メッセージで見つかりませんでした。

処置: 有効な要素名を指定してください。

JMS-410 XML SQL 例外

原因: JDBC SQL レイヤーでエラーが発生しました。

処置: 詳細は、リンクしている SQLException 内のメッセージを参照してください。

JMS-411 ペイロードの主要部は NULL にできません

原因: 無効な主要部の文字列またはドキュメントが指定されました。

処置: ペイロードに NULL 以外の主要部の文字列またはドキュメントを指定してください。

JMS-412 バイト変換に失敗しました

原因: 無効なユーザー名/パスワードが指定されました。

処置: NULL 以外のユーザー名およびパスワードを指定してください。

JMS-413 操作には自動コミットできません

原因: この操作に自動コミット・フラグを設定できません。

処置: 自動コミット・フラグを設定しないでください。

JMS-414 宛先の所有者を指定する必要があります

原因: 宛先の所有者に NULL が指定されました。

処置: NULL 以外の宛先名を指定してください。

JMS-415 無効な可視性値です

原因: 指定された可視性値が無効です。

処置: 有効な値は、AQxmlConstants.VISIBILITY_ONCOMMIT、AQxmlConstants.VISIBILITY_IMMEDIATE です。

JMS-416 無効なデキュー・モードです

原因: 無効なデキュー・モードが指定されました。

処置: 有効なデキュー・モードは、AQxmlConstants.DEQUEUE_BROWSE、AQxmlConstants.DEQUEUE_REMOVE、AQxmlConstants.DEQUEUE_LOCKED、AQxmlConstants.DEQUEUE_REMOVE_NODATA です。

JMS-417 無効なナビゲーション・モードです

原因: 無効なナビゲーション・モードが指定されました。

処置: 有効なナビゲーション・モードは、次のとおりです。

- AQxmlConstants.NAVIGATION_FIRST_MESSAGE
- AQxmlConstants.NAVIGATION_NEXT_MESSAGE
- AQxmlConstants.NAVIGATION_NEXT_TRANSACTION

JMS-418 wait_time の値が無効です

原因: wait 型の値が無効です。

処置: 待機時間には、AQDequeueOption.WAIT_FOREVER、AQDequeueOption.WAIT_NONE、または 0 より大きい値を使用できます。

JMS-419 ConnectionPoolDataSource が無効です

原因: NULL または無効な ConnectionPoolDataSource が指定されました。

処置: 正しい URL およびユーザー / パスワードとともに、有効な OracleConnectionPoolDataSource オブジェクトを指定してください。

JMS-420 cache_size の値が無効です

原因: 無効な cache_size が指定されました。

処置: キャッシュ・サイズに 0 より大きい値を指定する必要があります。

JMS-421 cache_scheme の値が無効です

原因: 無効なキャッシュ・スキームが指定されました。

処置: 有効なキャッシュ・スキームは、次のとおりです。

- OracleConnectionCacheImpl.DYNAMIC_SCHEME
- OracleConnectionCacheImpl.FIXED_WAIT_SCHEME

JMS-422 無効なタグです - {0}

原因: XML ドキュメントで無効なタグが検出されました。

処置: XML ドキュメントが AQ スキーマに準拠していることを確認してください。

JMS-423 値が無効です

原因: 無効な値が指定されました。

処置: XML ドキュメントに指定された値が AQ スキーマに指定された値に準拠していることを確認してください。

JMS-424 無効なメッセージ・ヘッダーです

原因: 指定されたメッセージ・ヘッダーが NULL または無効です。

処置: 有効なメッセージ・ヘッダーを指定してください。

JMS-425 プロパティ名の指定が必要です。

原因: プロパティ名が NULL でした。

処置: NULL 以外のプロパティ名を指定してください。

JMS-426 プロパティが存在しません

原因: 無効なプロパティ名が指定されました。プロパティが存在しません。

処置: プロパティが存在しません。

JMS-427 サブスクライバ名を指定する必要があります

原因: サブスクライバ名が NULL でした。

処置: NULL 以外のサブスクリプション名を指定してください。

JMS-428 有効なメッセージを指定する必要があります

原因: メッセージが NULL でした。

処置: NULL 以外のメッセージを指定してください。

JMS-429 登録オプションを指定する必要があります

原因: 登録オプションが NULL です。

処置: NULL 以外の登録オプションを指定してください。

JMS-430 データベース・リンクを指定する必要があります

原因: データベース・リンクが NULL です。

処置: NULL 以外の登録オプションを指定してください。

JMS-431 順序番号を指定する必要があります。

原因: 登録オプションが NULL です。

処置: NULL 以外の登録オプションを指定してください。

JMS-432 ステータスを指定する必要があります

原因: ステータス・オプションが NULL です。

処置: NULL 以外の登録オプションを指定してください。

JMS-433 ユーザーが認証されていません

原因: ユーザーが認証されていません。

処置: サーブレットに接続する前にユーザーが Web サーバーで認証されていることを確認してください。

JMS-434 無効なデータ・ソースです

原因: データ・ソースが NULL または無効です。

処置: データベースに接続する有効なデータ・ソースを指定してください。

JMS-435 スキーマの位置が無効です

原因: スキーマの位置が NULL または無効です。

処置: スキーマの有効な URL を指定してください。

JMS-436 AQ 例外

原因: AQ Java レイヤーでエラーが発生しました。

処置: 詳細は、AQxmlException 内のメッセージとリンクしている例外を参照してください。

JMS-437 無効な宛先です

原因: 無効な宛先オブジェクトが指定されました。

処置: 有効な宛先(キュー/トピック)オブジェクトを指定してください。

JMS-438 AQ エージェント{0}は有効なデータベース・ユーザーに対してマップされていません

原因: 指定された AQ エージェントが要求された操作を実行する権限を持つデータベース・ユーザーにマップされていません。

処置: `dbms_aqadm.enable_db_access` を使用して、必要なキューの権限を持つデータベース・ユーザーにエージェントをマップしてください。

JMS-439 無効なスキーマ・ドキュメントです

原因: 指定されたスキーマ・ドキュメントが有効ではありません。

処置: スキーマ・ドキュメントの有効な URL を指定してください。

JMS-440 無効な操作: エージェント{0}は複数のデータベース・ユーザーに対してマップします

原因: 同じセッションの複数のデータベース・ユーザーに AQ エージェントがマップされました。

処置: AQ エージェントを 1 つのデータベース・ユーザーにのみマップしてください。このエージェントにマップされるデータベース・ユーザーの `aq$internet_users` ビューを確認してください。

JMS-441: {0}は NULL にできません

原因: 指定したパラメータが NULL でした。

処置: NULL 以外の値を指定してください。

JMS-442: エージェント名およびアドレスは NULL にできません

原因: 名前とアドレス・パラメータの両方が NULL と指定されました。

処置: 名前とアドレスに NULL 以外の値を指定してください。

JMS-443: IMMEDIATE 可視性モードはこのキュー/項目ではサポートされていません

原因: IMMEDIATE 可視性モードは JMS 型のキュー/トピックではサポートされていません。

処置: JMS 型のキュー/項目には、ON_COMMIT またはデフォルトの可視性モードを使用してください。

JMS-444: この機能はサポートされていません

原因: 要求された機能はまだサポートされません。

処置: この機能をサポートする将来のリリースをお待ちください。

JMS-445: 宛先の別名を指定する必要があります

原因: 宛先の別名に NULL が指定されました。

処置: NULL 以外の宛先の別名を指定してください。

JMS-446: エージェントの別名を指定する必要があります

原因: エージェントの別名に NULL が指定されました。

処置: NULL 以外のエージェントの別名を指定してください。

JMS-447: LDAP サーバーへのアクセス時にエラーが発生しました

原因: LDAP サーバーへのアクセス中にエラーが発生しました。

処置: LDAP サーバーが稼働中であること、およびサーブレットに指定された環境パラメータが正しいことを確認してください。

JMS-448: 無効なコンテンツ・タイプです

原因: コンテンツ・タイプが無効です。

処置: コンテンツ・タイプは text/xml、または application/x-www-form-urlencoded (パラメータ名は aqxmldoc) である必要があります。

用語集

ADT

抽象データ型です。

API

[\[Application Program Interface\]](#)を参照してください。

Application Program Interface

アプリケーション・プログラムがオペレーティング・システムおよびその他のサービスにアクセスする場合のコール規則です。

近似CSCN

メッセージをコミット時間キューにエンキューしたトランザクションがコミットされる時点の、データベースの現在のSCNに基づくシステム変更番号の近似値です。

非同期

マルチタスキング・システム内で、独立してバックグラウンドでの実行を継続できるプロセスを非同期プロセスと呼びます。その他のプロセスは非同期プロセスの終了前に開始できます。[\[同期\]](#)の反対語です。

BFILE

オペレーティング・システム内に常駐するデータベース表領域の外に存在する外部のバイナリ・ファイルです。

バイナリ・ラージ・オブジェクト

内容がバイナリ・データで構成される[ラージ・オブジェクト](#)のデータ型です。このデータは、データ構造がデータベースに認識されないため、RAW型とみなされます。

BLOB

[\[バイナリ・ラージ・オブジェクト\]](#)を参照してください。

ブロードキャスト

[メッセージ・プロデューサ](#)に、メッセージ・[コンシューマ](#)の識別情報が伝達されない[パブリッシュ・サブスクライブ](#)・モードです。このモードは、ラジオ放送局またはテレビ放送局と類似しています。

バッファ・キュー

バッファ・キューは、バッファ属性を持つメッセージ(バッファ済メッセージ)のキューイングおよびメモリー内でのマテリアライズをサポートしています。バッファ済メッセージ専用のメモリーが新規メッセージ用に必要になる、またはバッファ済メッセージがメモリー内で指定の期間を超えると、そのバッファ済メッセージは基礎となるキュー表にスワップされます。バッファ済メッセージ用のメモリーは、スト

リーム・プールと呼ばれる別のプールから取得されます。データベースがバウンズされた場合、バッファ済メッセージはリカバリできません。バッファ済属性が設定されていないメッセージは、基礎となる永続キュー表に永続メッセージとしてキューイングされます。

標準

対象の通常または標準の状態、または方法です。

キャラクタ・ラージ・オブジェクト

値がデータベース・キャラクタ・セットに対応する文字データで構成された[ラージ・オブジェクト](#)のデータ型です。キャラクタ・ラージ・オブジェクトは、Oracle Text検索エンジンで索引付けおよび検索できます。

CLOB

[「キャラクタ・ラージ・オブジェクト」](#)を参照してください。

ConnectionFactory

ConnectionFactoryは、管理者によって定義された接続構成パラメータの集合をカプセル化します。クライアントはこれを使用して[Java Message Service](#)プロバイダとの接続を確立します。

コミット時間キュー

メッセージが近似CSCN値で順序付けされるキューです。

コンシューマ

メッセージの[デキュー](#)が可能なユーザーまたはアプリケーションです。

データ操作言語

データ操作言語(DML)文ではデータベースのデータを操作します。たとえば、表の行の問合せ、挿入、更新および削除はいずれもDML操作で、表またはビューのロックおよびSQL文の実行計画の調査もDML操作です。

Database Configuration Assistant

データベースの作成と削除およびデータベース・テンプレートの管理を行うためのOracle Databaseツールです。

DBCA

[「Database Configuration Assistant」](#)を参照してください。

デキュー

キューから[メッセージ](#)を取り出すことです。

DML

[「データ操作言語」](#)を参照してください。

エンキュー

[メッセージ](#)をキューに配置することです。JMS用語では、エンキューを[送信](#)といいます。

例外キュー

なんらかの理由により取得および処理されない場合、メッセージは例外[キュー](#)に転送されます。

IDAP

[「Internet Data Access Presentation」](#)を参照してください。

索引構成表

通常の表では、データは順不同のコレクションとして格納されますが、索引構成表のデータは、Bツリー索引構造に主キーでソートされて格納されます。Bツリーの各索引エントリには、索引構成表の行の主キーの列値のみでなく、非キーの列値も格納されます。

Internet Data Access Presentation

Oracle Database Advanced Queuing操作のための[Simple Object Access Protocol](#)(SOAP)仕様です。iDAPによってSOAPリクエストの本体にXMLメッセージ構造が定義されます。iDAP構造の[メッセージ](#)は、インターネット上でHTTPを使用して転送されます。

プロセス間通信

1つのプロセスと別のプロセスとの間でのデータ交換のことで、プロセスは同じコンピュータ内である場合と、ネットワーク上である場合があります。リクエストに対するレスポンスを保証するプロトコルが暗示されます。

IOT

[「索引構成表」](#)を参照してください。

IPC

[「プロセス間通信」](#)を参照してください。

Java Database Connectivity

Javaプログラムからリレーショナル・データベースに接続するための業界標準のJavaインタフェース。Sun Microsystemsが定義しました。

Java Message Service

Sun Microsystems、オラクル社、IBM社およびその他のベンダーが定義したメッセージ交換機能標準です。JMSは、JMSクライアントが企業のメッセージ関連製品の機能にアクセスする方法を定義する、インタフェースおよび対応するセマンティクスの集合です。

Java Naming and Directory Interface

Javaプログラムをネーミングおよびディレクトリ・サービスに接続するSunのプログラミング・インタフェースです。

Java Virtual Machine

コンパイル済Javaバイトコードをプラットフォームのマシン言語に変換し実行するJavaインタプリタです。JVMは、クライアント、ブラウザ、中間層、イントラネットの他、Oracle Application Server 10gなどのアプリケーション・サーバーまたはOracle Database 10gなどのデータベース・サーバーで実行できます。

JDBC

[「Java Database Connectivity」](#)を参照してください。

JDBCドライバ

特定のデータベースへのアクセスを可能にする[Java Database Connectivity](#)のベンダー固有レイヤーです。Oracle Databaseでは、Thin、[OCI](#)およびKPRBの3種類のJDBCドライバが使用できます。

JMS

[「Java Message Service」](#)を参照してください。

JMSコネクション

クライアントとそのJMSプロバイダ間のアクティブなコネクションです。通常は、クライアントとプロバイダのサービス・デーモン間のオープンなTCP/IPソケット(または一連のオープン・ソケット)を指します。

JMSメッセージ

JMSメッセージは、ヘッダー、1つ以上のオプション・プロパティおよびメッセージ・ペイロードで構成されます。

JMSセッション

メッセージの作成および処理用の単一スレッドのコンテキストです。

JMSトピック

その他のOracle Database Advanced Queuingインタフェースのマルチ・コンシューマ・キューと同等です。

JNDI

[「Java Naming and Directory Interface」](#)を参照してください。

Jnnn

ジョブ・キュー・プロセスです。

JServer

Oracle Databaseのメモリー領域で実行されるJava Virtual Machineです。

JVM

[\[Java Virtual Machine\]](#)を参照してください。

ラージ・オブジェクト

[BFILE](#)、[BLOB](#)、[CLOB](#)および[NCLOB](#)の各オブジェクトで構成されるSQLデータ型のクラスです。

LDAP

[\[Lightweight Directory Access Protocol\]](#)を参照してください。

Lightweight Directory Access Protocol

標準の拡張可能ディレクトリ・アクセス・プロトコル。LDAPクライアントおよびサーバーが通信に使用する共通言語です。Oracle Internet Directoryなど、業界標準のディレクトリ製品をサポートする設計規則のフレームワーク。

LOB

[\[ラージ・オブジェクト\]](#)を参照してください。

ローカル・コンシューマ

ローカル・[コンシューマ](#)は、[プロデューサ](#)がそのメッセージをエンキューしたキューから[メッセージ](#)をデキューします。

論理変更レコード

取得プロセスまたはユーザー・アプリケーションによってREDOログから取得され、データベース変更を記述する特定の書式を持つオブジェクトです。取得プロセスは、論理変更レコード(LCR)のみを含むメッセージをANYDATAキューにエンキューします。パフォーマンス向上のために、これらのLCRは常に[バッファ・キュー](#)に格納されます。

メッセージ

[キュー](#)に挿入する情報および取り出す情報の最小単位です。メッセージは、制御情報(メタデータ)とペイロード(データ)で構成されます。

マルチキャスト

[メッセージ・プロデューサ](#)に、各[コンシューマ](#)の識別情報が伝達される[パブリッシュ・サブスクライブ](#)モードです。このモードは、Point-to-Multipointとも呼ばれます。

各国語キャラクタ・ラージ・オブジェクト

値がデータベースの各国語キャラクタ・セットに対応する文字データで構成された[ラージ・オブジェクト](#)のデータ型です。

NCLOB

[「各国語キャラクタ・ラージ・オブジェクト」](#)を参照してください。

非永続

非永続キューは、メッセージをメモリーに格納します。これらは、通常、現在接続されているすべてのユーザーに通知を送信する[非同期](#)メカニズムを提供するために使用されます。非永続キューは、Oracle Database Advanced Queuing 10gリリース 2 (10.2)では非推奨です。バッファ・メッセージを使用することをお勧めします。

非トランザクション

一度に1つの[メッセージ](#)のみをエンキューおよびデキューできます。

オブジェクト型

オブジェクト型には、データ構造と、そのデータの操作に必要な関数およびプロシージャがカプセル化されています。CREATE TYPE文を使用してオブジェクト型を定義する場合は、実在のオブジェクトに対応する抽象テンプレートを作成します。

OCI

[「Oracle Call Interface」](#)を参照してください。

Oracle JMS

[「Oracle Java Message Service」](#)を参照してください。

OLTP

[「オンライン・トランザクション処理」](#)を参照してください。

オンライン・トランザクション処理

オンライン・トランザクション処理システムは、トランザクション処理の高速化および信頼性の向上を目的として最適化されています。データ・ウェアハウス・システムと比較して、OLTPの相互作用で扱われる行が少なく、表のグループが大きいのが通例です。

Oracle Call Interface

Oracle Databaseでのデータ操作および[スキーマ](#)操作を可能にするアプリケーション・プログラミング・インタフェースです。

Oracle Java Message Service

Oracle Java Message Service (Oracle JMS)には、[Java Message Service](#) (JMS)標準に基づいた、Oracle Database Advanced Queuing用のJava [API](#)が用意されています。Oracle JMSは、標準のJMSインタフェースをサポートしている他、標準以外のOracle Database Advanced Queuing管理操作や他のOracle Database Advanced Queuing機能をサポートする拡張機能を備えています。

プロデューサ

メッセージの[エンキュー](#)が可能なユーザーまたはアプリケーションです。

伝播

あるキューのメッセージを別の(ローカルまたはリモート)キューにコピーすることです。

パブリッシュ/サブスクライブ

[プロデューサ](#)から、1つ以上のマルチ・コンシューマ・キューにエンキューされた[メッセージ](#)が、複数のサブスクライバによってデキューされるメッセージ交換の方式です。パブリッシュ・サブスクライブ・メッセージには、広範囲な[ブロードキャスト](#)と呼ばれるモード、または対象範囲が狭い[マルチキャスト](#)と呼ばれるモードを設定できます。

QMNC

キュー・モニター・コーディネータです。システムの負荷に応じて、qXXXというスレーブを動的に起動します。スレーブは様々なバックグラウンド・タスクを実行します。

QMnN

キュー・モニター・プロセスです。

キュー

メッセージ・システムがメッセージを格納する場合に使用する抽象的な記憶単位です。

キュー表

キューが格納されるデータベース表です。各キュー表には、デフォルトの[例外キュー](#)があります。

受信者

メッセージを取り出す権限を、エンキュー元またはキュー管理者から付与されたエージェントです。エンキュー元は、メッセージの受信者として[メッセージ](#)を取り出すことができるコンシューマを明示的に指定できます。キュー管理者は、キューからメッセージを取り出すことができる受信者のデフォルト・リストを指定できます。デフォルト・リストで指定される受信者は、[サブスクライバ](#)と呼ばれます。受信者の指定がないメッセージがエンキューされると、そのメッセージはすべてのサブスクライバに送信されます。キュー内の特定のメッセージに特定の受信者(そのキューのサブスクライバであってもなくても可)を指定することで、サブスクライバ・リストをオーバーライドできます。

受信者の名前のみを指定した場合、その受信者は、メッセージをエンキューされたキューからデキューする必要があります。プロトコルの値を0(ゼロ)に設定して、受信者の名前およびアドレスを指定する場合、そのアドレスは、同じデータベースまたは別途インストールしたOracle Databaseの別のキューの名前にする必要があります。受信者の名前がNULLの場合、メッセージはアドレスに指定されたキューに伝播され、アドレスに指定されたキューのサブスクライバによってデキューされます。プロトコル・フィールドの値が0(ゼロ)でない場合、名前およびアドレスのフィールドはシステムにより無視され、メッセージは特定の[コンシューマ](#)によってデキューされます。

リモート・コンシューマ

リモート・[コンシューマ](#)は、[メッセージ](#)がエンキューされたキューとは異なるキューからデキューします。

結果キャッシュ

結果キャッシュは、一連の引数値の評価における結果のキャッシュとして定義できます。[ルール・エンジン](#)の場合、結果キャッシュは、特定の入力引数が指定されたルール・セット、ルールおよび独立した式の評価に関連する結果のキャッシュを意味します

ルール

[サブスクライバ](#)のメッセージ・サブスクライブへの関心を定義するブール式です。これらの式で使用される構文は、SQL問合せのWHERE句に類似しており、メッセージ・プロパティ(現行では、優先順位および相関識別子)、ユーザー・データ・プロパティ(オブジェクト・ペイロードのみ)およびファンクションに関する条件を含めることができます。ルールの評価結果がTRUEであれば、[受信者](#)が指定されていない場合でも、[メッセージ](#)は、そのルールに対応付けられたサブスクライバに送信されます。

ルール・エンジン

ルールを評価するOracle Databaseソフトウェアです。ルールとは、イベントが発生し条件が満たされたときに、クライアントがアクションを実行することを可能にするデータベース・オブジェクトです。ルールはSQL問合せのWHERE句の条件に類似しています。ユーザー作成のアプリケーションも、Oracle Database Advanced QueuingなどのOracle Databaseの機能も、ルール・エンジンのクライアントにすることができます。

スキーマ

表、ビュー、順序、ストアド・プロシージャ、シノニム、索引、クラスタ、データベース・リンクなどの論理構造を含むデータベース・オブジェクトの集合体です。スキーマには、そのスキーマを制御するユーザー名が付けられます。

送信

[エンキュー](#)のJMS用語です。

サーブレット

ネットワーク・サービスの一部として実行され、クライアントのリクエストに応答するJavaプログラムです。通常は、HTTPサーバーのことです。

SGA

[「システム・グローバル領域」](#)を参照してください。

シャード・キュー

システムで管理されているパーティション化を使用して、独立した複数の物理キューに分割される1つの論理キューです。

Simple Object Access Protocol

HTTPを介したXMLを使用してコードを起動する場合の最小限の規則で、World Wide Web Consortiumによって定義

されました。

SOAP

[\[Simple Object Access Protocol\]](#)を参照してください。

サブスクライバ

[キュー](#)からメッセージを取り出す権限を、キュー管理者から付与されたエージェントです。

システム・グローバル領域

Oracle Databaseの1つのインスタンスのデータおよび制御情報を含む、共有メモリー構造のグループです。Oracle Databaseのインスタンスは、SGAとOracle Databaseのプロセスで構成されます。インスタンスを起動すると、Oracle DatabaseによってSGAのメモリーが自動的に割り当てられ、インスタンスを停止すると、オペレーティング・システムからそのメモリーが解放されます。SGAは、1インスタンスにつき1つ割り当てられます。

同期

複数のプロセスが、共通のタイミング信号など、特定のイベントの発生に依存することを同期と呼びます。[\[非同期\]](#)の反対語です。

トランザクション

複数のメッセージをグループ化して同時エンキューまたは同時デキューすることが可能な状態です。

変換

2つのOracleデータ型のマッピングです。ソース・データ型を入力として取り、ターゲット・データ型のオブジェクトを返すSQLファンクションとして表されます。[メッセージ](#)を[キュー](#)に入れる前に正しいデータ型に変換するために、変換は[エンキュー](#)時に指定できます。メッセージを所定の形式で受信することは、[デキュー](#)中に指定できます。[リモート・コンシューマ](#)により指定された場合、メッセージは、宛先のキューに伝播する前に変換されます。

ユーザー・キュー

通常の[メッセージ](#)処理に使用される[キュー](#)です。

VARRAY

順序付けされたデータ要素のセットです。1つの配列内に存在する要素のデータ型はすべて同じです。各要素には、配列内の位置に対応する番号が付けられており、この番号をインデックスと呼びます。配列のサイズは、配列内の要素数によって表されます。Oracle Databaseでは、配列のサイズ変更ができます。

ワイルドカード

文字列比較で、どの文字にも一致する特殊文字または特殊文字列です。

ワークフロー

プロジェクトまたはビジネス取引の開始から終了までの全アクティビティ相互間の一連の関係です。アクティビティは、種類の異なるトリガー・リレーションによって関連付けられています。外部イベントや別のアクティビティによってアクティビティがトリガーされる場合もあります。

索引

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#)

A

- アクセス
 - オブジェクト・タイプ [1](#)
- アクセス制御
 - JMSでの宛先レベル [1](#)
 - キュー・レベル [1](#)
 - システム・レベル [1](#)
 - JMS [1](#)
- サブスクライバの追加 [1](#)
- 管理
 - メッセージ・ゲートウェイ [1](#)
- Oracle Database AQへの管理インタフェース
 - 比較 [1](#)
- AdtMessage
 - 概要 [1](#)
 - 作成 [1](#)
- エージェント
 - AQjmsエージェント
 - 作成 [1](#)
 - メッセージ・ゲートウェイ
 - 概要 [1](#)
 - エージェントの構成 [1](#)
 - モニタリング [1](#)
 - Oracle RAC上でのエージェントの実行 [1](#)
 - エージェントの停止 [1](#)
 - エージェントの起動 [1](#)
- エージェント・ユーザー
 - メッセージ・ゲートウェイ・エージェントの作成 [1](#)
- 別名
 - LDAPサーバーへの追加 [1](#)
 - LDAPサーバーからの削除 [1](#)
 - パラメータ
 - 別名 [1](#)
 - obj_location [1](#)
- ALL_QUEUE_SUBSCRIBERS [1](#)
- ALL_QUEUE_TABLES
 - 現行ユーザーがアクセス可能なキュー表のキュー [1](#)
- ALL_QUEUES [1](#)
- 変更

- AQエージェント [1](#)
- 伝播 [1](#)
- キュー [1](#)
- キュー表 [1](#)
- サブスクライバ [1](#)
- 変換 [1](#)
- ANYDATAデータ・タイプ
 - キュー表 [1](#)
- アプリケーション開発
 - 概要 [1](#)
 - クライアント/サーバー通信 [1](#)
 - インターネット操作 [1](#)
 - パブリッシュ/サブスクライブ [1](#)
 - サード・パーティのメッセージ [1](#)
 - ワークフロー [1](#)
- AQ_ADMINISTRATOR_ROLE
 - LDAP [1](#)
 - registerConnectionFactory [1](#)
 - 定義 [1](#)
 - JMSに必要な [1](#)
- AQ_MsgProperties [1](#)
- AQ_TM_PROCESSESパラメータ [1](#)
- AQ_USER_ROLE
 - 定義 [1](#)
 - JMSに必要な [1](#)
- AQ\$_AGENT [1](#)
- AQ\$_AGENT_LIST_T [1](#)
- AQ\$_POST_INFO_LIST [1](#)
- AQ\$_QUEUE_TABLE_NAME_D [1](#)
- AQ\$_QUEUE_TABLE_NAME_E [1](#)
- AQ\$_QUEUE_TABLE_NAME_H [1](#)
- AQ\$_QUEUE_TABLE_NAME_I [1](#)
- AQ\$_QUEUE_TABLE_NAME_P [1](#)
- AQ\$_QUEUE_TABLE_NAME_S [1](#)
- AQ\$_QUEUE_TABLE_NAME_T [1](#)
- AQ\$_RECIPIENT_LIST_T [1](#)
- AQ\$_REG_INFO_LIST [1](#)
- AQ\$_SUBSCRIBER_LIST_T [1](#)
- AQ\$INTERNET_USERS [1](#)
- AQ\$QUEUE_TABLE_NAME [1](#)
- AQ\$QUEUE_TABLE_NAME_R [1](#)
- AQ\$QUEUE_TABLE_NAME_S [1](#)
- AQエージェント
 - LDAPサーバーへの追加 [1](#)
 - 変更 [1](#)

- 作成 [1](#)
 - 削除 [1](#)
 - パラメータ
 - agent_name [1](#)
 - certificate_location [1](#)
 - enable_anyp [1](#)
 - enable_http [1](#)
 - LDAPサーバーからの削除 [1](#)
 - AQバックグラウンド・アーキテクチャ [1](#), [2](#)
 - AQjmsエージェント
 - 作成 [1](#)
 - AQメッセージ・プロパティ型 [1](#)
 - AQサブレット
 - デプロイ [1](#)
 - HTTPを使用したレスポンス [1](#)
 - AQシステム権限
 - 付与 [1](#)
 - JMS [1](#)
 - 取消し [1](#)
 - JMS [1](#)
 - アーキテクチャ
 - アプリケーション開発 [1](#)
 - インターネット操作 [1](#), [2](#)
 - メッセージ・ゲートウェイ [1](#)
 - 配列
 - デキュー
 - 概要 [1](#)
 - バッファ済メッセージ [1](#)
 - 構文 [1](#)
 - インキュー
 - 概要 [1](#)
 - 構文 [1](#)
 - 非同期通知
 - 概要 [1](#)
 - バッファ済メッセージ [1](#)
 - 指定のポート [1](#)
 - 後のページ [1](#)
 - RAWペイロード配信 [1](#)
 - 信頼性 [1](#)
 - タイムアウト [1](#)
 - JMSでの非同期受信 [1](#)
-

- [デキューのバッチ処理 1](#)
- [エンキューのバッチ処理 1](#)
- BFILEオブジェクト
 - [伝播 1](#)
- Booleanメッセージ・プロパティ(JMS)
 - [取得 1](#)
 - [設定 1](#)
- ブロードキャスト
 - [定義 1](#)
- バッファ済メッセージ
 - [概要 1](#)
 - [デキュー 1](#)
 - [オプション 1](#)
 - [エンキュー 1](#)
 - [例外処理 1](#)
 - [フロー制御 1](#)
 - [listen_delivery_mode 1](#)
 - [MSG_STATEパラメータ 1](#)
 - [通知 1](#)
 - [順序 1](#)
 - [伝播 1](#)
 - [キュー・ツー・キュー伝播 1](#)
 - [制限事項 1](#)
 - [チューニング 1](#)
 - [サポートされている型 1](#)
 - [ビュー 1](#)
 - [可視性 1](#)
 - [メッセージ・ゲートウェイ 1](#)
 - [Oracle JMS 1](#)
 - [Oracle Real Application Clusters 1](#)
- [バッファ・キュー 1](#)
- Byteメッセージ・プロパティ(JMS)
 - [取得 1](#)
 - [設定 1](#)
- BytesMessage
 - [概要 1](#)
 - [作成 1](#)
 - [例 1](#)

C

- CLASSPATH
 - [メッセージ・ゲートウェイ 1](#)
- クローズ

- JMSコネクション [1](#)
- JMSセッション [1](#)
- メッセージ・コンシューマ [1](#)
- MessageProducer [1](#)
- コミット
 - JMSセッション中の全操作 [1](#)
 - トランザクション [1](#)
- コミット時間順序付け
 - 概要 [1](#)
 - 例 [1](#)
 - 要件 [1](#)
- 互換性
 - 概要 [1](#)
 - Oracle Real Application Clusters [1](#)
 - キュー表の移行 [1](#)
 - 非永続キュー [1](#)
 - セキュリティ [1](#)
- 同時プロセス
 - Oracle Database AQ用のチューニング [1](#)
- 接続(JMS)
 - 作成
 - デフォルトのConnectionFactoryパラメータ [1](#), [2](#)
 - ユーザー名/パスワード [1](#), [2](#)
 - JMSセッションからのJDBCコネクションの取得 [1](#)
- ConnectionFactory
 - 取得
 - LDAP [1](#)
 - オブジェクト [1](#)
 - 登録
 - データベースを介してJDBCコネクション・パラメータを使用 [1](#)
 - データベースを介してJDBC URLを使用 [1](#)
 - LDAPを介してJDBCコネクション・パラメータを使用 [1](#)
 - LDAPを介してJDBC URLを使用 [1](#)
 - 登録解除
 - LDAPを介してLDAPで [1](#)
 - データベースを介してLDAPで [1](#)
 - データベースを介して [1](#)
 - LDAPを介して [1](#)
 - JNDIを使用した検索 [1](#)
- 変換
 - JMSメッセージ, 説明 [1](#)
 - メッセージ・ヘッダー [1](#)
 - JMS以外のメッセージ, 説明 [1](#)
 - TIB/Rendezvousメッセージ [1](#)
 - WebSphere MQメッセージ [1](#)

- 相関識別子
 - 概要 [1](#), [2](#)
 - トランザクション・グループ [1](#)
 - 仮想プライベート・データベース [1](#)
 - デキュー条件として [1](#)
 - メッセージ・プロパティとして [1](#)
 - MessageSelectorとして [1](#)
 - 指定によるデキュー [1](#)
 - JMSで取得 [1](#)
 - JMSで設定 [1](#)
 - キュー表の索引 [1](#)
- 作成
 - AQエージェント [1](#)
 - AQjmsエージェント [1](#)
 - DurableSubscriber [1](#), [2](#), [3](#), [4](#)
 - JMS AdtMessage [1](#)
 - JMS BytesMessage [1](#)
 - JMSコネクション [1](#), [2](#), [3](#), [4](#)
 - JMS MapMessage [1](#)
 - JMSメッセージ [1](#)
 - JMS ObjectMessage [1](#)
 - JMSセッション [1](#), [2](#)
 - JMS StreamMessage [1](#)
 - JMS TextMessage [1](#)
 - メッセージ・ゲートウェイ管理ユーザー [1](#)
 - メッセージ・ゲートウェイ・エージェント・ユーザー [1](#)
 - メッセージ・ゲートウェイの伝播サブスクライバ [1](#)
 - 非永続キュー [1](#)
 - JMSのPoint-to-Pointキュー [1](#)
 - JMSのPoint-to-Pointシャード・キュー [1](#)
 - 伝播 [1](#)
 - JMSでのトピックのパブリッシュ・サブスクライバ [1](#)
 - QueueBrowser [1](#), [2](#), [3](#)
 - QueueConnection [1](#), [2](#), [3](#), [4](#)
 - QueueReceiver [1](#), [2](#)
 - キュー [1](#)
 - JMS [1](#)
 - QueueSender [1](#)
 - QueueSession [1](#)
 - キュー表 [1](#)
 - JMS [1](#)
 - サブスクライバ [1](#)
 - TIB/Rendezvousリンク [1](#)
 - TopicConnection [1](#), [2](#), [3](#)
 - TopicPublisher [1](#)

- [TopicSession 1](#)
 - [変換 1](#)
 - [WebSphere MQ Base Javaリンク 1](#)
 - [WebSphere MQ JMSリンク 1](#)
-

D

- データベース
 - [アクセスの無効化 1](#)
 - [アクセスの有効化 1](#)
- データベース接続
 - [メッセージ・ゲートウェイ接続情報の構成 1](#)
 - [メッセージ・ゲートウェイ 1](#)
- [データ・ポンプ 1](#)
- データ型
 - [LONG VARCHAR 1](#)
- [DBA_ATTRIBUTE_TRANSFORMATIONS 1](#)
- [DBA_QUEUE_SCHEDULES 1](#)
- [DBA_QUEUE_SUBSCRIBERS 1](#)
- [DBA_QUEUE_TABLES](#)
 - [データベース内の全キュー表 1](#)
- [DBA_QUEUES 1](#)
- [DBA_SUBSCR_REGISTRATIONS](#)
 - [すべてのサブスクリプション登録 1](#)
- [DBA_TRANSFORMATIONS 1](#)
- [DBMS_AQ.BUFFERED 1](#)
- [DBMS_AQ.PERSISTENT 1](#)
- [DBMS_AQ.PERSISTENT_OR_BUFFERED 1](#)
- [DBMS_AQADMプロシージャ](#)
 - [ADD_ALIAS_TO_LDAP 1](#)
 - [ADD_SUBSCRIBER 1](#)
 - [ALTER_AQ_AGENT 1](#)
 - [ALTER_PROPAGATION_SCHEDULE 1](#)
 - [ALTER_QUEUE 1](#)
 - [ALTER_QUEUE_TABLE 1](#)
 - [ALTER_SUBSCRIBER 1](#)
 - [CREATE_AQ_AGENT 1](#)
 - [CREATE_NP_QUEUE 1](#)
 - [CREATE_QUEUE 1](#)
 - [CREATE_QUEUE_TABLE 1](#)
 - [CREATE_TRANSFORMATION 1](#)
 - [DEL_ALIAS_FROM_LDAP 1](#)
 - [DISABLE_DB_ACCESS 1](#)
 - [DISABLE_PROPAGATION_SCHEDULE 1](#)

- DROP_AQ_AGENT [1](#)
- DROP_QUEUE [1](#)
- DROP_QUEUE_TABLE [1](#)
- DROP_TRANSFORMATION [1](#)
- ENABLE_DB_ACCESS [1](#)
- ENABLE_PROPAGATION_SCHEDULE [1](#)
- GRANT_QUEUE_PRIVILEGE [1](#)
- GRANT_SYSTEM_PRIVILEGE [1](#)
- MIGRATE_QUEUE_TABLE [1](#)
- MODIFY_TRANSFORMATION [1](#)
- PURGE_QUEUE_TABLE [1](#)
- REMOVE_SUBSCRIBER [1](#)
- REVOKE_QUEUE_PRIVILEGE [1](#)
- REVOKE_SYSTEM_PRIVILEGE [1](#)
- SCHEDULE_PROPAGATION [1](#)
- START_QUEUE [1](#)
- STOP_QUEUE [1](#)
- UNSCHEDULE_PROPAGATION [1](#)
- VERIFY_QUEUE_TYPES [1](#), [2](#)
- DBMS_AQIN [1](#)
- DBMS_AQプロシージャ
 - BIND_AGENT [1](#)
 - DEQUEUE [1](#)
 - DEQUEUE_ARRAY [1](#)
 - ENQUEUE [1](#)
 - ENQUEUE_ARRAY [1](#)
 - LISTEN [1](#)
 - POST [1](#)
 - REGISTER [1](#)
 - UNBIND_AGENT [1](#)
 - UNREGISTER [1](#)
- DBMS_MGWADMパッケージ
 - 概要 [1](#)
 - ADD_SUBSCRIBER [1](#), [2](#)
 - ALTER_AGENT [1](#)
 - ALTER_MSGSYSTEM_LINK [1](#), [2](#)
 - ALTER_SUBSCRIBER [1](#)
 - CREATE_MSGSYSTEM_LINK [1](#), [2](#), [3](#), [4](#)
 - DB_CONNECT_INFO [1](#), [2](#)
 - DISABLE_PROPAGATION_SCHEDULE [1](#)
 - DOMAIN_QUEUE [1](#)
 - DOMAIN_TOPIC [1](#)
 - ENABLE_PROPAGATION_SCHEDULE [1](#)
 - JMS_CONNECTION [1](#)
 - JMS_QUEUE_CONNECTION [1](#)

- [JMS_TOPIC_CONNECTION 1](#)
- [MQSERIES_BASE_JAVA_INTERFACE 1](#)
- [REGISTER_FOREIGN_QUEUE 1, 2](#)
- [REMOVE_MSGSYSTEM_LINK 1](#)
- [RESET_SUBSCRIBER 1](#)
- [SHUTDOWN 1](#)
- [STARTUP 1](#)
- [UNREGISTER_FOREIGN_QUEUE 1](#)
- [DBMS_MGWMSG.LCR_TO_XML 1](#)
- 遅延
 - [デキュー中 1](#)
 - [JMSで指定 1](#)
- デキューの条件
 - [仮想プライベート・データベース 1](#)
 - [キュー表の索引 1](#)
- デキュー
 - [バッファ済メッセージ 1](#)
 - [マルチ・コンシューマ 1](#)
 - [同時プロセス 1](#)
 - [機能 1](#)
 - [IDAPクライアント・リクエスト 1](#)
 - [リクエストに対するIDAPサーバー・レスポンス 1](#)
 - [メッセージ配列 1, 2](#)
 - [メッセージ 1](#)
 - [メッセージの状態 1](#)
 - [メソッド 1](#)
 - モード
 - [概要 1](#)
 - [メッセージのナビゲーション 1](#)
 - [オプション 1](#)
 - [バッファ済メッセージ 1](#)
 - パラメータ
 - [array_size 1](#)
 - [dequeue_options 1, 2](#)
 - [遅延を伴う再試行 1](#)
 - [トランザクション保護 1](#)
 - [メッセージの待機 1](#)
- 宛先(JMS)
 - [変更 1](#)
 - [削除 1](#)
 - [開始 1](#)
 - [停止 1](#)
- 無効化
 - [データベース・アクセス 1](#)
 - [伝播 1](#)

- doubleメッセージ・プロパティ(JMS)
 - 取得 [1](#)
 - 設定 [1](#)
 - 削除
 - AQエージェント [1](#)
 - 伝播 [1](#)
 - キュー [1](#)
 - キュー表 [1](#)
 - 変換 [1](#)
 - DurableSubscriber
 - 概要 [1](#)
 - 作成
 - JMSトピック [1](#), [2](#)
 - Oracleオブジェクト型のトピック [1](#), [2](#)
 - サブスクライブの解除
 - ローカル・サブスクライバ [1](#)
 - リモート・サブスクライバ [1](#)
-

E

- 電子メール通知 [1](#)
- 有効化
 - データベース・アクセス [1](#)
 - 伝播 [1](#)
- インキュー
 - バッファ済メッセージ [1](#)
 - クライアント・リクエスト [1](#)
 - 相関識別子 [1](#)
 - 機能 [1](#)
 - IDAPクライアント・リクエスト [1](#)
 - リクエストに対するIDAPサーバー・レスポンス [1](#)
 - メッセージ配列 [1](#), [2](#)
 - メッセージの期限切れ [1](#)
 - メッセージのグループ化 [1](#)
 - メッセージ・プロパティ [1](#)
 - メッセージ [1](#)
 - オプション [1](#)
 - パラメータ
 - array_size [1](#)
 - enqueue_options [1](#)
 - message_properties [1](#)
 - message_properties_array [1](#)
 - payload [1](#)
 - payload_array [1](#)

- [メッセージの優先順位および順序付け](#) 1
 - [送信元の識別](#) 1
- 列挙定数
 - [概要](#) 1
 - [delivery_mode](#) 1
 - [操作インタフェース](#) 1
- 環境変数
 - [CLASSPATH](#) 1
 - [メッセージ・ゲートウェイ](#) 1
 - [MGW_PRE_PATH](#) 1
 - [ORACLE_SID](#) 1
- エラー状態
 - [メッセージ・ゲートウェイ](#) 1
- エラー処理
 - [エラー・メッセージ](#) 1
 - [IDAP](#) 1
 - [伝播](#) 1
- [エラー・メッセージ](#) 1
 - [メッセージ・ゲートウェイ](#) 1
 - [メッセージ・ゲートウェイ・エージェント](#) 1
- エラー(JMS)
 - [コードの取得](#) 1
 - [番号の取得](#) 1
- [イベント・ジャーナル](#) 1
- 例外(JMS)
 - [JMS例外にリンクされている例外, 取得](#) 1
 - 例外リスナー
 - [取得](#) 1
 - [設定](#) 1
 - [スタック・トレースの出力](#) 1
- 例外処理
 - [バッファ済メッセージ](#) 1
 - [例外キュー](#) 1, 2
 - [メッセージ・ゲートウェイ](#) 1
 - [JMSでの伝播](#) 1
- 例外キュー
 - [概要](#) 1
 - [JMS](#) 1
- 期限切れ
 - [エンキュー中に設定](#) 1
 - [JMSで指定](#) 1
- エクスポート
 - キュー表
 - [概要](#) 1
 - [データ・ポンプ](#) 1

- [モード 1](#)
 - [複数の受信者 1](#)
-

F

- floatメッセージ・プロパティ(JMS)
 - [取得 1](#)
 - [設定 1](#)
 - フロー制御
 - [概要 1](#)
-

G

- 取得(JMS)
 - [ConnectionFactory 1](#)
 - [相関識別子 1](#)
 - [エラー・コード 1](#)
 - [エラー番号 1](#)
 - [例外 1](#)
 - [JDBC接続 1](#)
 - [JMSコネクション 1](#)
 - [メッセージ識別子 1](#)
 - [非シャード・キュー表 1](#)
 - [OracleOCIConnectionPool 1](#)
 - [QueueConnectionFactory 1](#)
 - [LDAP 1](#)
 - [JDBCコネクション・パラメータ 1](#)
 - [JDBC URL 1](#)
 - [LDAPのキュー 1](#)
 - [TopicConnectionFactory 1](#)
 - [JDBCコネクション・パラメータ 1](#)
 - [JDBC URL 1](#)
 - [LDAPのトピック 1](#)
- GLOBAL_AQ_USER_ROLE
 - [registerConnectionFactory 1, 2](#)
- 権限付与
 - [AQシステム権限 1](#)
 - [JMS 1](#)
 - [キュー権限 1](#)
 - [JMS 1](#)
 - [JMSでのトピック権限 1](#)
- グループ化
 - [メッセージ 1](#)
- [GV\\$AQ 1](#)

H

- HTTP
 - AQ操作 [1](#)
 - AQサブレットのレスポンス [1](#)
 - クライアント・リクエスト [1](#)
 - ヘッダー [1](#)
 - 伝播 [1](#)
 - 伝播に使用 [1](#)
 - レスポンス [1](#)
-

I

- I/O
 - Oracle Database AQに構成 [1](#)
- IDAP
 - クライアント・リクエスト
 - コミット・トランザクション [1](#)
 - デキュー [1](#)
 - エンキュー [1](#)
 - 登録 [1](#)
 - トランザクションのロールバック [1](#)
 - エラー処理 [1](#)
 - メッセージ [1](#)
 - 通知 [1](#)
 - リクエストおよびレスポンス文書 [1](#)
 - サーバー・レスポンス
 - コミット・トランザクション [1](#)
 - デキュー・リクエスト [1](#)
 - エンキュー・リクエスト [1](#)
 - 登録リクエスト [1](#)
 - トランザクションのロールバック [1](#)
- インポート
 - キュー表
 - 概要 [1](#)
 - データ・ポンプ [1](#)
 - IGNOREパラメータ [1](#)
 - 複数の受信者 [1](#)
- 受信ボックス [1](#)
- 索引
 - Oracle Database AQ用のチューニング [1](#)
- INIT.ORAパラメータ [1](#)
- 初期化パラメータ

- [メッセージ・ゲートウェイ 1](#)
- [integerメッセージ・プロパティ\(JMS\)](#)
 - [取得 1](#)
 - [設定 1](#)
- [Oracle Database AQへのインタフェース](#)
 - [概要 1](#)
 - [管理 1](#)
 - [AQ XMLサブレット 1](#)
 - [比較 1, 2](#)
 - [JMS 1](#)
 - [OCCI 1](#)
 - [OCI 1](#)
 - [OCI security 1](#)
 - [操作 1](#)
 - [PL/SQL 1](#)
- [Internet Data Access Presentation](#)
 - [概要 1](#)
- [インターネット操作](#)
 - [アプリケーション開発 1](#)
 - [AQサブレットのレスポンス 1](#)
 - [アーキテクチャ 1, 2](#)
 - [クライアント・リクエスト 1](#)
 - [AQサブレットのデプロイ 1](#)
 - [IDAPクライアント・リクエスト](#)
 - [コミット・トランザクション 1](#)
 - [デキュー 1](#)
 - [エンキュー 1](#)
 - [登録 1](#)
 - [トランザクションのロールバック 1](#)
 - [IDAPIエラー 1](#)
 - [IDAP通知 1](#)
 - [IDAPリクエストおよびレスポンス文書 1](#)
 - [IDAPサーバー・レスポンス](#)
 - [コミット・トランザクション 1](#)
 - [デキュー・リクエスト 1](#)
 - [エンキュー・リクエスト 1](#)
 - [登録リクエスト 1](#)
 - [トランザクションのロールバック 1](#)
 - [Internet Data Access Presentation 1](#)
 - [JMS型 1](#)
 - [電子メールによる通知 1](#)
 - [オブジェクト型キュー 1](#)
 - [ペイロード 1](#)
 - [伝播 1](#)
 - [RAWキュー 1](#)

- SOAP
 - 本体 [1](#)
 - エンベロープ [1](#)
 - メッセージ構造 [1](#)
 - メソッドの起動 [1](#)
 - ユーザー認証 [1](#)
-

J

- Java EEの準拠 [1](#)
- Javaプロパティ
 - メッセージ・ゲートウェイ [1](#)
 - oracle.mgw.batch_size [1](#)
 - oracle.mgw.polling_interval [1](#)
 - oracle.mgw.tibrv.advMsgCallback [1](#)
 - oracle.mgw.tibrv.encoding [1](#)
 - oracle.mgw.tibrv.intraProcAdvSubjects [1](#)
- JDBC接続
 - JMSセッションからの取得 [1](#)
 - データベースを介したJDBCパラメータによるConnectionFactoryの登録 [1](#)
 - LDAPを介したConnectionFactoryの登録に使用 [1](#)
- JDBC OCIドライバ
 - JMSに必要 [1](#)
- JDBC Thinドライバ
 - JMSに必要 [1](#)
- JDBC URL
 - LDAPを介したJDBC URLによるConnectionFactoryの登録 [1](#)
 - データベースを介した登録 [1](#)
- JMS
 - 概要 [1](#)
 - Oracle Real Application Clusters [1](#)
 - 非同期受信 [1](#)
 - バッファ済メッセージ [1](#)
 - 接続 [1](#)
 - 例外キュー [1](#)
 - 必要なJDBC OCIドライバ [1](#)
 - 必要なJDBC Thinドライバ [1](#)
 - メッセージ本体 [1](#)
 - メッセージ・コンシューマ機能 [1](#)
 - メッセージ・ヘッダー [1](#)
 - MessageProducerの機能 [1](#)
 - メッセージ・プロパティ [1](#)
 - メッセージ・タイプ [1](#)
 - 非シャード・キュー表

- 作成 [1](#)
- Point-to-Point機能 [1](#)
- 伝播スケジュール [1](#)
- パブリッシュ・サブスクライブ機能 [1](#)
- キュー, 作成 [1](#)
- キュー表
 - 取得 [1](#)
- 受信者リスト [1](#)
- セッション [1](#)
- 統計ビューのサポート [1](#)
- 構造化ペイロード [1](#)
- トラブルシューティング [1](#)
- JMS_DeliveryMode [1](#)
- JMS_NoLocal [1](#)
- JMSコネクション
 - 概要 [1](#)
 - クローズ [1](#)
 - 取得 [1](#)
 - OracleOCIConnectionPoolの取得 [1](#)
 - 開始 [1](#)
 - 停止 [1](#)
- JMS相関識別子
 - 設定 [1](#)
- JMS宛先
 - 概要 [1](#)
 - 管理 [1](#)
 - メソッド [1](#)
 - JMSセッションを使用した取得 [1](#)
 - JNDIを使用した検索 [1](#)
- JMSの例
 - BytesMessage [1](#)
 - MapMessage [1](#)
 - 設定 [1](#)
 - StreamMessage [1](#)
 - TextMessage [1](#)
- JMSメッセージ・プロパティ
 - ブール [1](#), [2](#)
 - byte [1](#), [2](#)
 - double [1](#), [2](#)
 - float [1](#), [2](#)
 - integer [1](#), [2](#)
 - long [1](#), [2](#)
 - object [1](#), [2](#)
 - short [1](#), [2](#)
 - string [1](#), [2](#)

- JMSメッセージ
 - 参照 [1](#)
 - TopicBrowser [1](#)
 - 相関識別子 [1](#)
 - 作成
 - AdtMessage [1](#)
 - BytesMessage [1](#)
 - JMSメッセージ [1](#)
 - MapMessage [1](#)
 - ObjectMessage [1](#)
 - StreamMessage [1](#)
 - TextMessage [1](#)
 - 遅延, 指定 [1](#)
 - 期限切れ, 指定 [1](#)
 - グループ化 [1](#)
 - メッセージ・コンシューマ, クローズ [1](#)
 - メッセージ識別子 [1](#)
 - メッセージ・リスナー
 - JMSセッションで指定 [1](#)
 - メッセージ・コンシューマで指定 [1](#)
 - MessageProducer, クローズ [1](#)
 - メッセージ・プロパティ
 - 取得 [1](#)
 - 設定 [1](#)
 - 受信時のナビゲート [1](#)
 - 受信のナビゲーション・モード, 指定 [1](#)
 - 優先度
 - デフォルトの設定 [1](#)
 - 優先順位および順序付け [1](#)
 - メッセージ・ゲートウェイでの伝播
 - インバウンド [1](#)
 - アウトバウンド [1](#)
 - 公開
 - 受信者リストの指定 [1](#)
 - 配信モード、優先順位およびTime-To-Liveの指定 [1](#)
 - トピックの指定 [1](#)
 - 最小限の指定 [1](#)
 - QueueBrowser, 作成 [1](#), [2](#), [3](#)
 - QueueReceiver, 作成 [1](#), [2](#)
 - 受信
 - 概要 [1](#)
 - 非同期 [1](#), [2](#)
 - 変換を使用して宛先から [1](#)
 - 同期 [1](#), [2](#), [3](#)
 - メッセージ・コンシューマ [1](#), [2](#)

- リモート・サブスクライバ, 作成 [1](#)
 - 取出しを伴わないメッセージの削除 [1](#)
 - 遅延を伴う再試行 [1](#)
 - QueueSenderを使用した送信 [1](#), [2](#)
 - TimeToLive
 - デフォルトの設定 [1](#)
 - TopicBrowser, 作成 [1](#), [2](#)
 - TopicReceiver, 作成 [1](#), [2](#)
 - JMS AQでの変換 [1](#)
 - JMS伝播
 - 概要 [1](#)
 - 変更 [1](#)
 - 無効化 [1](#)
 - 有効化 [1](#)
 - 例外処理 [1](#)
 - RemoteSubscriber [1](#)
 - スケジューリング [1](#) [2](#)
 - スケジュールの解除 [1](#)
 - JMSパブリッシュ/サブスクライブ
 - 設定 [1](#)
 - JMSセッション
 - 概要 [1](#)
 - クローズ [1](#)
 - 全操作のコミット [1](#)
 - 作成 [1](#), [2](#)
 - JDBCコネクションの取得 [1](#)
 - 全操作のロールバック [1](#)
 - メッセージ・リスナーの指定 [1](#)
 - Destinationオブジェクトの取得に使用 [1](#)
 - JMSシャード・キュー [1](#)
 - JMS型のキュー/トピック [1](#)
 - JMSタイプ
 - インターネット操作 [1](#)
 - JNDI
 - ConnectionFactoryオブジェクトの検索に使用 [1](#)
 - Destinationオブジェクトの検索に使用 [1](#)
 - JOB_QUEUE_PROCESSES [1](#)
-

L

- LDAP
 - AQ_ADMINISTRATOR_ROLE [1](#)
 - キュー/トピック・コネクション・ファクトリ [1](#)
 - ConnectionFactoryの登録 [1](#)

- [ConnectionFactoryの登録解除 1](#)
 - LDAPサーバー
 - [別名の追加 1](#)
 - [AQエージェントの追加 1](#)
 - [別名の削除 1](#)
 - [AQエージェントの削除 1](#)
 - リンク
 - [変更 1](#)
 - [メッセージ・ゲートウェイ・リンクの構成 1](#)
 - [MGW_LINKSビュー 1](#)
 - [MGW_MQSERIES_LINKビュー 1](#)
 - [MGW_TIBRV_LINKSビュー 1](#)
 - [削除 1](#)
 - [TIB/Rendezvous, 作成 1](#)
 - [WebSphere MQ Base Java, 作成 1](#)
 - [WebSphere MQ JMS, 作成 1](#)
 - listener.ora
 - [メッセージ・ゲートウェイ用の変更 1, 2](#)
 - [TIB/Rendezvous用の変更 1](#)
 - [WebSphere MQ用の変更 1](#)
 - リスニング
 - [概要 1](#)
 - [アプリケーション開発 1](#)
 - パラメータ
 - [agent_list 1](#)
 - [listen_delivery_mode 1](#)
 - [wait 1](#)
 - [構文 1](#)
 - LOB
 - [伝播 1](#)
 - [log_directory 1](#)
 - [log_level 1](#)
 - ログ・ファイル
 - [メッセージ・ゲートウェイ 1](#)
 - 論理変更レコード
 - [メッセージ・ゲートウェイ 1](#)
 - longメッセージ・プロパティ(JMS)
 - [取得 1](#)
 - [設定 1](#)
 - [LONG VARCHARデータ型 1](#)
-

M

- 管理

- 非永続キュー [1](#)
- 伝播 [1](#), [2](#)
- キュー [1](#), [2](#)
- キュー表 [1](#)
- サブスクライバ [1](#)
- 変換 [1](#)
- MapMessage
 - 概要 [1](#)
 - 作成 [1](#)
 - 例 [1](#)
- メッセージ・ヘッダー
 - メッセージ・ゲートウェイでの変換 [1](#)
 - WebSphere MQのマッピング [1](#)
- メッセージ識別子
 - 概要 [1](#)
 - トランザクション・グループ [1](#)
 - JMSで取得 [1](#)
- MessageListener [1](#)
- MessageProducer
 - クローズ [1](#)
 - 機能 [1](#), [2](#)
 - デフォルトの優先順位の設定 [1](#)
 - デフォルトのTimeToLiveの設定 [1](#)
- メッセージ・プロパティ
 - TIB/Rendezvous [1](#)
 - JMSでメッセージ型とともに使用 [1](#)
- メッセージ
 - 配列のデキュー [1](#), [2](#)
 - 配列のエンキュー [1](#), [2](#)
 - JMSでの本体 [1](#)
 - JMSでのブラウズ [1](#), [2](#)
 - 相関識別子
 - 概要 [1](#)
 - 相関識別子 [1](#)
 - JMS [1](#)
 - JMSで作成 [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
 - JMSでのリモート・サブスクライバの作成 [1](#)
 - 遅延, JMSで指定 [1](#)
 - デキュー
 - 機能 [1](#)
 - メソッド [1](#)
 - モード [1](#)
 - 構文 [1](#)
 - 同時プロセス [1](#)
 - エンキュー

- [機能 1](#)
 - [オプション 1](#)
 - [構文 1](#)
- [例外キュー 1](#)
- 期限切れ
 - [概要 1](#)
 - [JMSで指定 1](#)
- [フォーマットの変換 1](#)
- [グループ化 1](#)
 - [JMS 1](#)
- [メッセージ・ゲートウェイでのヘッダーの変換 1](#)
- [JMSでのヘッダー 1](#)
- [JMSでの履歴と保存 1](#)
- 識別子
 - [概要 1](#)
- [JMSメッセージの変換 1](#)
- [JMSメッセージ・プロパティ](#)
 - [取得 1](#)
- [JMSメッセージ・プロパティ, 設定 1](#)
- [JMSでのメッセージ・コンシューマ, クローズ 1](#)
- [JMSでのメッセージ識別子 1](#)
- [JMSでのMessageProducer, クローズ 1](#)
- [JMSでのナビゲート 1](#)
- [デキュー中のナビゲーション 1](#)
- [受信時のナビゲーション 1](#)
- [ナビゲーション・モード, JMSで指定 1](#)
- [JMS以外のメッセージの変換 1](#)
- [否認防止 1](#)
- [オブジェクト型のサポート 1](#)
- 順序付け
 - [バッファ済メッセージ 1](#)
- [伝播中の順序付け 1](#)
- [ペイロードの制限事項 1](#)
- 永続性
 - [セキュリティ 1](#)
 - [メタデータの分析 1](#)
 - [スケジューリング 1](#)
- [優先順位, JMSで設定 1](#)
- [優先順位および順序付け 1](#)
 - [JMS 1](#)
- [伝播中の優先順位 1](#)
- 伝播
 - [エラー 1](#)
 - [機能 1](#)
 - [受信ボックスと送信ボックス 1](#)

- LOB [1](#)
 - リモート・コンシューマ [1](#)
 - スケジューリング [1](#)
 - 統計 [1](#)
 - HTTPの使用 [1](#)
 - Oracle RACの使用 [1](#)
 - プロパティ [1](#)
 - JMS [1](#)
 - JMSでのパブリッシュ [1](#), [2](#), [3](#), [4](#)
 - QueueBrowser, 作成 [1](#), [2](#), [3](#)
 - QueueReceiver, 作成 [1](#), [2](#)
 - JMSで受信 [1](#)
 - JMSで同期的に受信 [1](#), [2](#), [3](#)
 - 受信者
 - 概要 [1](#)
 - リモート・サブスクライバ, JMSで作成 [1](#)
 - JMSでの取出しを伴わないメッセージの削除 [1](#)
 - 保存と履歴 [1](#)
 - デキュー中の再試行 [1](#)
 - JMSでの遅延間隔をおいた後の再試行 [1](#)
 - 送信元の識別 [1](#)
 - JMSで送信 [1](#), [2](#)
 - デキュー中の状態 [1](#)
 - サード・パーティの伝播サポート [1](#)
 - TIB/Rendezvousの変換 [1](#)
 - TimeToLive, JMSで設定 [1](#)
 - TopicBrowser, 作成 [1](#), [2](#), [3](#), [4](#)
 - TopicReceiver, 作成 [1](#), [2](#)
 - 追跡 [1](#)
 - トランザクション保護 [1](#)
 - 変換 [1](#)
 - JMS [1](#)
 - JMSでプロパティとともに型を使用 [1](#)
 - デキュー中の待機 [1](#)
 - WebSphere MQの変換 [1](#)
 - XML変換 [1](#)
- MessageSelector
 - 概要 [1](#)
 - JMSのメッセージ型
 - 概要 [1](#)
 - AdtMessage [1](#)
 - BytesMessage [1](#)
 - MapMessage [1](#)
 - ObjectMessage [1](#)
 - StreamMessage [1](#)

- TextMessage [1](#)
- メッセージ・ゲートウェイ
 - 概要 [1](#)
 - 管理 [1](#)
 - 管理ユーザー
 - 作成 [1](#)
 - エージェント
 - 概要 [1](#)
 - 構成 [1](#)
 - エラー・メッセージ [1](#)
 - 停止 [1](#)
 - 開始 [1](#)
 - エージェント・ユーザー
 - 作成 [1](#)
 - JMS [1](#)
 - Oracle以外のメッセージ・システム [1](#)
 - アーキテクチャ [1](#)
 - バッファ済メッセージ [1](#)
 - 標準型 [1](#)
 - データベース接続 [1](#)
 - データベース接続情報, 構成 [1](#)
 - 環境変数 [1](#)
 - エラー状態 [1](#)
 - エラー・メッセージ [1](#)
 - 例外処理 [1](#)
 - 機能 [1](#)
 - Oracle RAC環境 [1](#)
 - 初期化ファイル [1](#)
 - 概要 [1](#)
 - 初期化パラメータ [1](#)
 - Oracle Databaseとの統合 [1](#)
 - Javaプロパティ [1](#)
 - リンク
 - 変更 [1](#)
 - ロード [1](#)
 - ログ・ファイル [1](#)
 - 論理変更レコード [1](#)
 - メッセージの変換(JMS) [1](#)
 - メッセージの変換(JMS以外) [1](#)
 - メッセージ・システム・リンク
 - 構成 [1](#)
 - listener.oraの変更 [1](#), [2](#)
 - エージェント・ステータスの監視 [1](#)
 - Oracle以外のメッセージ
 - 構成プロパティ [1](#)

- オプションのリンク構成プロパティ [1](#)
- Oracle以外のメッセージ・キュー
 - 構成 [1](#)
- Oracle以外のキュー
 - 登録解除 [1](#)
- オプションの外部キュー構成プロパティ [1](#)
- オプションのサブスクライバ構成プロパティ [1](#)
- 伝播 [1](#)
- 伝播の無効化 [1](#)
- 伝播の有効化 [1](#)
- 伝播のリセット [1](#)
- 伝播 [1](#)
 - モニタリング [1](#)
- 伝播スケジュール
 - 削除 [1](#)
- 伝播サブスクライバ
 - 作成 [1](#)
 - 削除 [1](#)
- 伝播サブスクライバ [1](#)
- Oracle以外のキューの登録 [1](#)
- リンクの削除 [1](#)
- リソース制限 [1](#)
- Oracle RAC上でのエージェントの実行 [1](#)
- TIB/Rendezvous用の設定 [1](#)
- WebSphere MQ用の設定 [1](#)
- サード・パーティのメッセージの設定 [1](#)
- 設定
 - 手順 [1](#)
 - 検証 [1](#)
- アンロード [1](#)
- Oracle以外のキューのビュー [1](#)
- ビュー [1](#)
- リンクのビュー [1](#)
- メッセージ・ゲートウェイ・ユーザー
 - MGW_AGENT_ROLE [1](#)
- MGW_ADMINISTRATOR_ROLE
 - メッセージ・ゲートウェイ管理ユーザー [1](#)
 - 作成 [1](#)
- MGW_AGENT_OPTIONS
 - サプリメンタル・オプションおよびプロパティ [1](#)
- MGW_AGENT_ROLE [1](#)
 - メッセージ・ゲートウェイ・ユーザー [1](#)
 - 作成 [1](#)
- MGW_BASIC_MSG_T [1](#)
- MGW_FOREIGN_QUEUES [1](#)

- 外部キュー [1](#)
- MGW_GATEWAY [1](#), [2](#)
 - 構成およびステータスに関する情報 [1](#)
- MGW_JOBS
 - メッセージ・ゲートウェイ伝播ジョブ [1](#)
- MGW_LINKS [1](#)
 - メッセージ・システム・リンクの名前および型 [1](#)
- MGW_MQSERIES_LINK [1](#)
- MGW_MQSERIES_LINKS
 - WebSphere MQメッセージ・システム・リンク [1](#)
- MGW_PRE_PATH [1](#)
- MGW_SCHEDULES
 - スケジュールに関する情報 [1](#)
- MGW_SUBSCRIBERS
 - サブスクライバに関する情報 [1](#)
- MGW_TIBRV_LINKS [1](#)
 - TIB/Rendezvousメッセージ・システム・リンク [1](#)
- MGW_TIBRV_MSG_T [1](#)
- mgw.ora
 - 概要 [1](#)
 - コメント行 [1](#)
 - 環境変数 [1](#)
 - Javaプロパティ [1](#)
 - TIB/Rendezvous用の変更 [1](#)
 - WebSphere MQ用の変更 [1](#)
 - パラメータ [1](#)
 - 設定 [1](#)
- 移行
 - キュー表 [1](#)
- 変更
 - メッセージ・ゲートウェイ用のlistener.ora [1](#), [2](#)
 - 変換 [1](#)
- 監視
 - メッセージ・ゲートウェイ [1](#)
 - 伝播 [1](#)
 - メッセージ・ゲートウェイ・エージェントのステータス [1](#)
- MQ_BrokerControlQueue [1](#)
- MQ_BrokerPubQueue [1](#)
- MQ_BrokerQueueManager [1](#)
- MQ_BrokerVersion [1](#)
- MQ_ccsid [1](#)
- MQ_CharacterSet [1](#)
- MQ_JmsDurSubQueue [1](#), [2](#)
- MQ_JmsTargetClient [1](#)
- MQ_openOptions [1](#)

- [MQ_PubAckInterval 1](#)
 - [MQ_ReceiveExit 1](#)
 - [MQ_ReceiveExitInit 1](#)
 - [MQ_SecurityExit 1](#)
 - [MQ_SecurityExitInit 1](#)
 - [MQ_SendExit 1](#)
 - [MQ_SendExitInit 1](#)
 - [MsgBatchSize 1](#)
 - マルチキャスト
 - [定義 1](#)
 - [マルチ・コンシューマによるデキュー 1](#)
-

N

- names
 - キュー
 - [長さ 1](#)
 - [大文字と小文字の混合使用 1](#)
 - キュー表
 - [長さ 1](#)
 - [大文字と小文字の混合使用 1](#)
- ナビゲーション
 - [デキュー中 1](#)
 - モード
 - [FIRST_MESSAGE 1](#)
 - [NEXT_MESSAGE 1](#)
 - [NEXT_TRANSACTION 1](#)
 - [JMSでモードを指定 1](#)
- 非永続キュー
 - [互換性 1](#)
 - [作成 1](#)
 - [管理 1](#)
 - [通知 1](#)
 - [制限事項 1](#)
- 否認防止
 - [概要 1](#)
- 通知
 - [概要 1](#)
 - [バッファ済メッセージ 1](#)
 - [指定のポート 1](#)
 - [電子メール 1](#)
 - [IDAP 1](#)
 - [非永続キュー 1](#)
 - パラメータ

- [post_count 1](#)
- [post_list 1](#)
- [reg_count 1](#)
- [reg_list 1](#)
- [転送 1](#)
- [後のページ 1](#)
- [RAWペイロード配信 1](#)
- [登録 1](#)
- [信頼性 1](#)
- [タイムアウト 1](#)
- [登録解除 1](#)

O

- [object_name 1](#)
- [ObjectMessage](#)
 - [概要 1](#)
 - [作成 1](#)
- [objectメッセージ・プロパティ\(JMS\)](#)
 - [取得 1](#)
 - [設定 1](#)
- [オブジェクト・タイプ](#)
 - [アクセス 1](#)
 - [サポート 1](#)
 - [シノニム 1](#)
- [OCCI](#)
 - [Oracle Database AQへのインタフェース 1](#)
 - [Oracle Type Translator 1](#)
- [OCI](#)
 - [Oracle Database AQへのインタフェース 1](#)
 - [Oracle Type Translator 1](#)
- [Oracle Database AQへの操作インタフェース 1](#)
- [オプション](#)
 - [デキュー 1](#)
 - [エンキュー 1](#)
- [ORACLE_SID](#)
 - [メッセージ・ゲートウェイ 1](#)
- [oracle.mgw.batch_size 1](#)
- [oracle.mgw.polling_interval 1](#)
- [oracle.mgw.tibrv.advMsgCallback 1](#)
- [oracle.mgw.tibrv.encoding 1](#)
- [oracle.mgw.tibrv.intraProcAdvSubjects 1](#)
- [Oracle AQのビュー 1](#)
- [Oracle Enterprise Manager](#)

- Oracle Database AQ [1](#)
 - サポート [1](#)
 - Oracle Internet Directory
 - Oracle Database AQ [1](#)
 - Oracle Database AQ統合 [1](#)
 - Oracle JMS
 - 概要 [1](#)
 - Java EEの準拠 [1](#)
 - Oracle Messaging Gatewayのビュー [1](#)
 - Oracleオブジェクト型(ADT)のキュー
 - インターネット操作 [1](#)
 - OracleOCIConnectionPool
 - JMSコネクションからの取得 [1](#)
 - Oracle RAC
 - バッファ済メッセージ [1](#)
 - メッセージ・ゲートウェイの構成 [1](#)
 - Oracle Database AQ併用時のパフォーマンス [1](#)
 - キュー・サービス名 [1](#)
 - メッセージ・ゲートウェイ・エージェントの実行 [1](#)
 - Real Application Clusters
 - JMS [1](#)
 - メッセージ伝播 [1](#)
 - サポート [1](#)
 - Oracle Type Translator [1](#)
 - 順序付け
 - コミット時間 [1](#)
 - 伝播中 [1](#)
 - JMSでのメッセージ [1](#)
 - エンキュー中に指定 [1](#)
 - 送信ボックス [1](#)
-

P

- パラメータ
 - admin_option [1](#)
 - agent_list [1](#)
 - agent_name [1](#), [2](#)
 - 別名 [1](#)
 - AQ_TM_PROCESSES [1](#)
 - array_size [1](#), [2](#)
 - attempts [1](#)
 - attribute_number [1](#)
 - certificate [1](#)
 - certificate_location [1](#)

- [comment](#) [1](#), [2](#)
- [互换性](#) [1](#)
- [互换性](#) [1](#)
- [consumer_name](#) [1](#)
- [correlation](#) [1](#), [2](#)
- [db_username](#) [1](#)
- [delay](#) [1](#)
- [delivery_mode](#) [1](#), [2](#), [4](#)
- [deq_condition](#) [1](#)
- [dequeue](#) [1](#), [2](#)
- [dequeue_mode](#) [1](#)
- [dequeue_options](#) [1](#), [2](#)
- [dest_queue_name](#) [1](#)
- [destination](#) [1](#), [2](#)
- [destination_queue](#) [1](#)
- [duration](#) [1](#)
- [enable_anyp](#) [1](#)
- [enable_http](#) [1](#)
- [enqueue](#) [1](#), [2](#)
- [enqueue_options](#) [1](#)
- [enqueue_time](#) [1](#)
- [exception_queue](#) [1](#)
- [expiration](#) [1](#)
- [from_schema](#) [1](#)
- [from_type](#) [1](#)
- [grant_option](#) [1](#)
- [grantee](#) [1](#)
- [latency](#) [1](#)
- [listen_delivery_mode](#) [1](#)
- [log_directory](#) [1](#)
- [log_level](#) [1](#)
- [max_retries](#) [1](#)
- [message_grouping](#) [1](#)
- [message_properties](#) [1](#)
- [message_properties_array](#) [1](#)
- [MSG_STATE](#) [1](#)
- [msgid](#) [1](#)
- [multiple_consumers](#) [1](#)
- [name](#) [1](#)
- [namespace](#) [1](#)
- [navigation](#) [1](#)
- [next_time](#) [1](#)
- [obj_location](#) [1](#)
- [original_msgid](#) [1](#)
- [OWNER_INSTANCE](#) [1](#)

- payload [1](#), [2](#)
- payload_array [1](#)
- post_count [1](#)
- post_list [1](#)
- primary_instance [1](#)
- priority [1](#)
- purge_condition [1](#)
- purge_options [1](#)
- queue_name [1](#)
- queue_payload_type [1](#)
- queue_table [1](#), [2](#)
- queue_to_queue [1](#)
- queue_type [1](#)
- recipient_list [1](#)
- reg_count [1](#)
- reg_list [1](#)
- relative_msgid [1](#)
- REMOTE_LISTENER [1](#)
- retention_time [1](#)
- retry_delay [1](#)
- rule [1](#)
- secondary_instance [1](#)
- secure [1](#)
- sender_id [1](#)
- sequence_deviation [1](#)
- sort_list [1](#)
- src_queue_name [1](#)
- start_time [1](#)
- 状態 [1](#)
- storage_clause [1](#)
- streams_pool_size [1](#)
- to_schema [1](#)
- to_type [1](#)
- transaction_group [1](#)
- 変換 [1](#), [2](#), [3](#)
- user_property [1](#)
- visibility [1](#), [2](#)
- wait [1](#), [2](#)
- ペイロード
 - インターネット操作 [1](#)
 - 制限事項 [1](#)
 - 構造化 [1](#)
 - メッセージ・ゲートウェイでの変換 [1](#)
 - XMLType [1](#)
- パフォーマンス

- 概要 [1](#)
- バッファ済メッセージ [1](#)
- 同時プロセス [1](#)
- I/Oの構成 [1](#)
- Oracle Database AQおよびOracle RAC [1](#)
- 永続メッセージ [1](#)
- 伝播のチューニング [1](#)
- キュー表の索引 [1](#)
- シリアル・プロセス [1](#)
- 共有サーバー [1](#)
- 記憶域パラメータ [1](#)
- 永続メッセージ
 - バッファ済との比較 [1](#)
 - パフォーマンス [1](#)
 - チューニング [1](#)
- Point-to-Pointメッセージ
 - 概要 [1](#)
- ポート
 - 通知用に指定 [1](#)
- 通知用の転送 [1](#)
- PreserveMessageID [1](#)
- 優先度
 - 伝播中 [1](#)
 - エンキュー中に指定 [1](#)
- 優先順位(JMS)
 - 概要 [1](#)
 - MessageProducerからの全メッセージに対して設定 [1](#)
- 権限
 - AQシステム権限
 - 付与 [1](#)
 - JMSで付与 [1](#)
 - 取消し [1](#)
 - JMSで取消し [1](#)
 - DBMS_AQIN [1](#)
 - パラメータ
 - admin_option [1](#)
 - grant_option [1](#)
 - grantee [1](#)
 - キュー権限
 - 付与 [1](#)
 - JMSで付与 [1](#)
 - 取消し [1](#)
 - JMSで取消し [1](#)
 - 伝播に必要 [1](#)
 - セキュリティ [1](#)

- トピック権限
 - JMSで付与 [1](#)
 - JMSで取消し [1](#)
- プログラム・インタフェース
 - 概要 [1](#)
 - AQ XMLサブレット [1](#)
 - 比較 [1](#), [2](#)
 - JMS [1](#)
 - OCCI [1](#)
 - OCI [1](#)
 - OCI security [1](#)
 - PL/SQL [1](#)
- 伝播
 - 概要 [1](#)
 - JMS [1](#)
 - 変更 [1](#)
 - JMS [1](#)
 - BFILEオブジェクト [1](#)
 - バッファ済メッセージ [1](#)
 - 作成 [1](#)
 - デバッグ [1](#)
 - 無効化 [1](#)
 - JMS [1](#)
 - メッセージ・ゲートウェイ [1](#)
 - 削除 [1](#)
 - 有効化 [1](#)
 - JMS [1](#)
 - メッセージ・ゲートウェイ [1](#)
 - エラー処理 [1](#)
 - JMS [1](#)
 - 機能 [1](#)
 - 受信ボックスと送信ボックス [1](#)
 - メッセージ・ゲートウェイでのJMSメッセージ
 - インバウンド [1](#)
 - アウトバウンド [1](#)
 - 管理 [1](#), [2](#)
 - LOBを含むメッセージ [1](#)
 - メッセージ・ゲートウェイ
 - 構成 [1](#)
 - モニタリング [1](#)
 - リセット [1](#)
 - サブスクリイバ, 説明 [1](#)
 - サブスクリイバ, 作成 [1](#)
 - サブスクリイバ, 削除 [1](#)
 - 最適化 [1](#)

- パラメータ
 - destination [1](#)
 - destination_queue [1](#)
 - duration [1](#)
 - latency [1](#)
 - next_time [1](#)
 - start_time [1](#)
- メッセージの優先順位および順序付け [1](#)
- 必要な権限 [1](#)
- キューからデータベース・リンクへ
 - 概要 [1](#)
 - スケジューリング [1](#)
 - Oracle RACの使用 [1](#)
- キューからキューへ
 - 概要 [1](#)
 - バッファ済メッセージ [1](#)
 - スケジューリング [1](#)
 - Oracle RACの使用 [1](#)
- リモート・コンシューマ
 - 概要 [1](#)
- スケジュール
 - 概要 [1](#)
 - 変更 [1](#)
 - 構文の作成 [1](#)
 - JMS [1](#), [2](#)
 - メッセージ・ゲートウェイで削除 [1](#)
- スケジューリング
 - JMS [1](#)
- セキュリティ [1](#)
- 統計 [1](#)
- サード・パーティのサポート [1](#)
- TIB/Rendezvous [1](#), [2](#)
- チューニング [1](#)
- スケジュールの解除 [1](#)
 - JMS [1](#)
- HTTPの使用 [1](#), [2](#)
- HTTPおよびHTTPSの使用 [1](#)
- WebSphere MQ [1](#), [2](#)
- メッセージ・ゲートウェイ [1](#)
- Oracle RACの使用 [1](#)
- パブリッシュ/サブスクライブ [1](#)
 - 概要 [1](#)
 - 設定 [1](#), [2](#)
- JMSメッセージのパブリッシュ
 - 受信者リストの指定 [1](#)

- 配信モード、優先順位およびTime-To-Liveの指定 [1](#)
- トピックの指定 [1](#)
- 最小限の指定 [1](#)
- ページ
 - 通知後 [1](#)
- ページ
 - キュー表 [1](#)

Q

- QUEUE_PRIVILEGES [1](#)
- キュー/トピック/コネクション/ファクトリ
 - LDAPで取得 [1](#)
- キュー(JMS)
 - LDAPで取得 [1](#)
- QueueBrowser
 - 概要 [1](#)
 - Oracleオブジェクト型メッセージ用の作成 [1](#)
 - 標準JMS型メッセージ用の作成 [1](#), [2](#)
- QueueConnection
 - デフォルトのConnectionFactoryパラメータを使用して作成 [1](#)
 - オープンしているJDBCコネクションを使用して作成 [1](#)
 - オープンしているOracleOCIConnectionPoolを使用して作成 [1](#)
 - ユーザー名/パスワードを指定して作成 [1](#)
- QueueConnectionFactory
 - 取得
 - LDAP [1](#)
 - JDBCコネクション・パラメータで取得 [1](#)
 - JDBC URLで取得 [1](#)
 - 登録
 - データベースを介してJDBCコネクション・パラメータを使用 [1](#)
 - データベースを介してJDBC URLを使用 [1](#)
 - LDAPを介してJDBCコネクション・パラメータを使用 [1](#)
 - LDAPを介してJDBC URLを使用 [1](#)
 - 登録解除
 - データベースを介して [1](#)
 - LDAPを介して [1](#)
- キュー・モニター・コーディネータ [1](#)
- キュー・モニター・コーディネータ [1](#)
- キュー権限
 - 付与 [1](#)
 - JMS [1](#)
 - 取消し [1](#)
 - JMS [1](#)

- QueueReceiver
 - 概要 [1](#)
 - Oracleオブジェクト型メッセージ用の作成 [1](#)
 - 標準JMS型メッセージ用の作成 [1](#)
- キュー
 - 変更 [1](#)
 - クリーン・アップ [1](#)
 - 作成 [1](#)
 - JMS [1](#)
 - 削除 [1](#)
 - 例外 [1](#)
 - JMS [1](#)
 - リスニング [1](#)
 - 管理の制限事項 [1](#)
 - 管理 [1](#), [2](#)
 - モニター・コーディネータ [1](#)
 - names
 - 長さ [1](#)
 - 大文字と小文字の混合使用 [1](#), [2](#)
 - Oracle以外
 - 構成 [1](#)
 - 登録 [1](#)
 - 非永続 [1](#)
 - 互換性 [1](#)
 - 管理 [1](#)
 - 通知 [1](#)
 - 制限事項 [1](#)
 - パラメータ
 - comment [1](#)
 - dequeue [1](#), [2](#)
 - enqueue [1](#), [2](#)
 - max_retries [1](#)
 - queue_name [1](#)
 - queue_table [1](#)
 - queue_type [1](#)
 - retention_time [1](#)
 - retry_delay [1](#)
 - Point-to-Point
 - JMSで作成 [1](#), [2](#)
 - 制限事項 [1](#), [2](#)
 - secure [1](#)
 - セキュリティ [1](#)
 - 開始 [1](#)
 - 停止 [1](#)
 - サブスクライバ

- [概要 1](#)
- [型, 検証 1](#)
- QueueSender
 - [概要 1](#)
 - [作成 1](#)
 - [メッセージの送信とオプションの指定 1](#)
 - [デフォルトのオプションを使用したメッセージの送信 1](#)
- QueueSession
 - [作成 1](#)
- キュー表
 - [変更 1](#)
 - [作成 1](#)
 - [JMS 1](#)
 - [データ・ポンプ 1](#)
 - [削除 1](#)
 - [エクスポート](#)
 - [モード 1](#)
 - [エクスポート](#)
 - [概要 1, 2](#)
 - [JMSで取得 1](#)
 - [インポート](#)
 - [概要 1, 2](#)
 - [IGNOREパラメータ 1](#)
 - [複数の受信者 1](#)
 - [管理 1](#)
 - [移行 1](#)
 - [複数の受信者](#)
 - [エクスポート1](#)
- names
 - [長さ 1](#)
 - [大文字と小文字の混合使用 1, 2](#)
- パラメータ
 - [comment 1](#)
 - [互換性 1](#)
 - [message_grouping 1](#)
 - [multiple_consumers 1](#)
 - [primary_instance 1](#)
 - [queue_payload type 1](#)
 - [queue_table 1](#)
 - [secondary_instance 1](#)
 - [secure 1](#)
 - [sort_list 1](#)
 - [storage_clause 1](#)
- [ペイロード型 1](#)
- [ページ 1](#)

- [制限事項 1](#)
 - [セキュリティ 1](#)
 - [ソート・キー 1](#)
 - [STORAGE句 1](#)
 - [パフォーマンスのための索引のチューニング 1](#)
 - キューイング
 - [Oracle Database 1](#)
 - [定義 1](#)
-

R

- RAW
 - [通知によるペイロード配信 1](#)
 - [インターネット操作にRAWキューを使用 1](#)
- 受信者
 - [概要 1](#)
 - [JMSの受信者リスト 1](#)
- リカバリ
 - [制限事項 1](#)
- REFペイロード
 - [制限事項 1](#)
- registerConnectionFactory
 - [AQ_ADMINISTRATOR_ROLE 1](#)
 - [GLOBAL_AQ_USER_ROLE 1](#)
 - [LDAPを介してJDBCコネクション・パラメータを使用 1](#)
 - [データベースを介してJDBCコネクション・パラメータを使用 1](#)
 - [LDAPを介してJDBC URLを使用 1](#)
- 登録
 - [通知 1](#)
 - [データベースを介してJDBC URLを使用 1](#)
- 登録
 - [クライアント・リクエスト 1](#)
 - [IDAPクライアント・リクエスト 1](#)
 - [リクエストに対するIDAPサーバー・レスポンス 1](#)
- 信頼性
 - [通知 1](#)
- リモート・コンシューマ
 - [伝播 1](#)
- RemoteSubscriber [1](#), [2](#)
- リモート・サブスクライバ
 - [制限事項 1](#)
- リソース制限
 - [メッセージ・ゲートウェイ 1](#)
- 制限事項

- バッファ済メッセージ [1](#)
 - メッセージ・ペイロード [1](#)
 - 非永続キュー [1](#)
 - Point-in-Timeリカバリ [1](#)
 - キュー管理 [1](#)
 - REFペイロード [1](#)
 - リモート・サブスクライバ [1](#)
 - サブスクライバ [1](#)
 - シノニム [1](#)
 - 仮想プライベート・データベース [1](#)
 - 結果キャッシュ [1](#)
 - 保存
 - メッセージ [1](#)
 - JMS [1](#)
 - 再試行
 - デキュー中 [1](#)
 - 複数のセッションによるデキュー [1](#)
 - 取消し
 - AQシステム権限 [1](#)
 - JMS [1](#)
 - キュー権限 [1](#)
 - JMS [1](#)
 - ロール
 - AQ_ADMINISTRATOR_ROLE [1](#), [2](#), [3](#)
 - AQ_USER_ROLE [1](#), [2](#)
 - GLOBAL_AQ_USER_ROLE [1](#), [2](#)
 - MGW_ADMINISTRATOR_ROLE [1](#), [2](#)
 - MGW_AGENT_ROLE [1](#), [2](#)
 - ロールバック
 - JMSセッション中の全操作 [1](#)
 - RV_discardAmount [1](#)
 - RV_limitPolicy [1](#)
 - RV_maxEvents [1](#)
-

S

- 拡張性
 - 概要 [1](#)
- スケジュール
 - メッセージ・ゲートウェイでの伝播の有効化と無効化 [1](#)
- スケジューリング
 - 伝播スケジュールの説明 [1](#)
 - SCHEDULE_PROPAGATIONを使用して伝播 [1](#)
- 保護キュー [1](#)

- セキュリティ [1](#)
 - JMSでの宛先レベル [1](#)
 - JMSでのシステム・レベル [1](#)
 - 互換性パラメータ [1](#)
 - メッセージの永続性 [1](#)
 - OCIアプリケーション [1](#)
 - 伝播 [1](#)
 - キュー権限 [1](#)
 - キュー表
 - secureパラメータ [1](#)
- 送信元の識別
 - エンキュー中 [1](#)
- シリアル・プロセス
 - Oracle Database AQ用のチューニング [1](#)
- セッション(JMS)
 - 作成 [1](#)、[2](#)
- シャード・キュー [1](#)
- 共有サーバー
 - Oracle Database AQ併用時のパフォーマンス [1](#)
- shortメッセージ・プロパティ(JMS)
 - 取得 [1](#)
 - 設定 [1](#)
- SOAP
 - 本体 [1](#)
 - エンベロープ [1](#)
 - ヘッダー [1](#)
 - メッセージ構造 [1](#)
 - メソッドの起動 [1](#)
- スタック・トレース
 - JMSで出力 [1](#)
- 起動
 - JMSコネクション [1](#)
 - メッセージ・ゲートウェイ・エージェント [1](#)
 - キュー [1](#)
- 統計
 - 伝播 [1](#)
- 停止
 - JMSコネクション [1](#)
 - キュー [1](#)
- 記憶域パラメータ
 - Oracle Database AQのチューニング [1](#)
- StreamMessage
 - 概要 [1](#)
 - 作成 [1](#)
 - 例 [1](#)

- stringメッセージ・プロパティ(JMS)
 - 取得 [1](#)
 - 設定 [1](#)
- 構造化ペイロード [1](#)
 - 概要 [1](#)
 - JMS [1](#)
- サブスクライバ
 - 概要 [1](#)
 - 追加 [1](#)
 - 変更 [1](#)
 - 作成 [1](#)
 - Oracleオブジェクト型メッセージ用のJMSリモート・サブスクライバの作成 [1](#)
 - JMSメッセージ用のリモート・サブスクライバの作成 [1](#)
 - メッセージ・ゲートウェイで作成 [1](#)
 - メッセージ・ゲートウェイの伝播 [1](#)
 - 管理 [1](#)
 - names
 - 大文字と小文字の混合使用 [1](#)
 - 順序 [1](#)
 - パラメータ
 - delivery_mode [1](#)
 - queue_to_queue [1](#)
 - rule [1](#)
 - 変換 [1](#)
 - 削除 [1](#)
 - 制限事項 [1](#)
 - JMSで変換を指定 [1](#), [2](#)
 - DurableSubscribersのサブスクライブの解除 [1](#), [2](#)
- シノニム
 - 制限事項 [1](#)
- SYS.AQ\$_DESCRIPTOR [1](#)
- SYS.AQ\$_POST_INFO [1](#)
- SYS.AQ\$_REG_INFO [1](#)
- SYS.MGW_MQSERIES_PROPERTIES [1](#)
- SYS.MGW_TIBRV_PROPERTIES [1](#)
- システム権限
 - 付与 [1](#)
 - JMS [1](#)
 - 取消し [1](#)
 - JMS [1](#)

T

- TextMessage

- 概要 [1](#)
- 作成 [1](#)、[2](#)
- 例 [1](#)
- サード・パーティのメッセージ
 - アプリケーション開発 [1](#)
 - メッセージ・ゲートウェイ [1](#)
 - 構成プロパティ [1](#)
 - オプションの外部キュー構成プロパティ [1](#)
 - オプションのリンク構成プロパティ [1](#)
 - オプションのサブスクリバ構成プロパティ [1](#)
 - キュー
 - 構成 [1](#)
 - 登録 [1](#)
 - 登録解除 [1](#)
 - 設定 [1](#)
 - 登録済キューのビュー [1](#)
- TIB/Rendezvous
 - AQ_MsgProperties [1](#)
 - リンク
 - 作成 [1](#)
 - listener.ora, 変更 [1](#)
 - メッセージの変換 [1](#)
 - メッセージ・プロパティのマッピング [1](#)
 - メッセージ・ゲートウェイ, 設定 [1](#)
 - mgw.ora, 変更 [1](#)
 - MsgBatchSize [1](#)
 - PreserveMessageID [1](#)
 - 伝播
 - インバウンド [1](#)
 - アウトバウンド [1](#)
 - RV_discardAmount [1](#)
 - RV_limitPolicy [1](#)
 - RV_maxEvents [1](#)
 - サブジェクト
 - 登録 [1](#)
 - 登録解除 [1](#)
 - システム・プロパティ [1](#)
- タイムアウト
 - 通知 [1](#)
- 時間指定
 - インキュー中 [1](#)
- TimeToLive
 - MessageProducerからの全メッセージに対して設定 [1](#)
- トピック
 - DurableSubscriberの作成 [1](#), [2](#), [3](#), [4](#)

- JMSで作成 [1](#)
- LDAPで取得 [1](#)
- JMSでのトピック権限の付与 [1](#)
- JMSでのトピック権限の取消し [1](#)
- サブスクライバ用の変換の指定 [1](#)
- TopicBrowser [1](#)
 - メッセージのブラウズに使用 [1](#)
 - Oracle型メッセージ・トピック用の作成 [1](#), [2](#)
 - 標準JMS型メッセージ・トピック用の作成 [1](#), [2](#)
- TopicConnection
 - オープンしているJDBCコネクションを使用して作成 [1](#)
 - オープンしているOracleOCIConnectionPoolを使用して作成 [1](#)
 - ユーザー名/パスワードを指定して作成 [1](#)
- TopicConnectionFactory
 - 取得
 - LDAP [1](#)
 - JDBCコネクション・パラメータ [1](#)
 - JDBC URL [1](#)
 - 登録
 - データベースを介してJDBCコネクション・パラメータを使用 [1](#)
 - データベースを介してJDBC URLを使用 [1](#)
 - LDAPを介してJDBCコネクション・パラメータを使用 [1](#)
 - LDAPを介してJDBC URLを使用 [1](#)
 - 登録解除
 - データベースを介して [1](#)
 - LDAPを介して [1](#)
- TopicPublisher
 - 概要 [1](#)
 - 作成 [1](#)
- TopicReceiver
 - 概要 [1](#)
 - Oracleオブジェクト型メッセージ・トピック用の作成 [1](#)
 - 標準JMS型メッセージ・トピック用の作成 [1](#)
- TopicSession
 - 作成 [1](#)
- トランザクション
 - IDAPクライアント・リクエスト
 - コミット [1](#)
 - ロールバック [1](#)
 - デキュー中の保護 [1](#)
- 変換
 - 概要 [1](#)
 - 変更 [1](#)
 - 作成 [1](#)
 - 削除 [1](#)

- リモート・サブスクライバ用, JMSで指定 [1](#)
- トピック・サブスクライバ用, JMSで指定 [1](#)
- 管理 [1](#)
- メッセージ・ゲートウェイ [1](#)
- 変更 [1](#)
- パラメータ
 - attribute_number [1](#)
 - from_schema [1](#)
 - from_type [1](#)
 - to_schema [1](#)
 - to_type [1](#)
 - 変換 [1](#)
- XML [1](#)
- トラブルシューティング
 - JMS [1](#)
- チューニング
 - バッファ済メッセージ [1](#)
 - 永続メッセージ [1](#)
- type_name [1](#)
- タイプ
 - アクセス [1](#)
 - aq\$_purge_options_t [1](#)
 - AQエージェント [1](#)
 - AQエージェント・リスト [1](#)
 - AQ通知記述子 [1](#)
 - AQ転送情報 [1](#)
 - AQ転送情報リスト [1](#)
 - AQ受信者リスト [1](#)
 - AQ登録情報 [1](#)
 - AQ登録情報リスト [1](#)
 - AQサブスクライバ・リスト [1](#)
 - バッファ済メッセージのサポート [1](#)
 - メッセージ・ゲートウェイ [1](#)
 - MGW_BASIC_MSG_T [1](#)
 - MGW_TIBRV_MSG_T [1](#)
 - サポート [1](#)

U

- 登録解除
 - LDAPのConnectionFactory [1](#), [2](#)
 - 通知 [1](#)
- スケジュールの解除
 - 伝播 [1](#)

- JMS [1](#)
 - USER_ATTRIBUTE_TRANSFORMATIONS
 - ユーザー変換ファンクション [1](#)
 - USER_QUEUE_SCHEDULES [1](#)
 - USER_QUEUE_SUBSCRIBERS [1](#)
 - USER_QUEUE_TABLES [1](#)
 - USER_QUEUES [1](#)
 - USER_SUBSCR_REGISTRATIONS
 - ユーザー・サブスクリプション登録 [1](#)
 - USER_TRANSFORMATIONS [1](#)
 - ユーザー認証
 - インターネット操作 [1](#)
 - ユーザー
 - メッセージ・ゲートウェイ・エージェント [1](#)
-

V

- V\$AQ_BACKGROUND_COORDINATOR [1](#)
- V\$AQ_CROSS_INSTANCE_JOBS [1](#)
- V\$AQ_IPC_ACTIVE_MSGS [1](#)
- V\$AQ_IPC_MSG_STATS [1](#)
- V\$AQ_IPC_PENDING_MSGS [1](#)
- V\$AQ_JOB_COORDINATOR [1](#)
- V\$AQ_MESSAGE_CACHE_ADVICE [1](#)
- V\$AQ_MESSAGE_CACHE_STAT [1](#)
- V\$AQ_NONDUR_REGISTRATIONS [1](#)
- V\$AQ_NONDUR_SUBSCRIBER [1](#)
- V\$AQ_NONDUR_SUBSCRIBER_LWM [1](#)
- V\$AQ_NOTIFICATION_CLIENTS [1](#)
- V\$AQ_REMOTE_DEQUEUE_AFFINITY [1](#)
- V\$AQ_SERVER_POOL [1](#)
- V\$AQ_SHARDED_SUBSCRIBER_STAT [1](#)
- V\$AQ_SUBSCRIBER_LOAD [1](#)
- V\$BUFFERED_PUBLISHERS
 - インスタンスにあるすべてのバッファ済パブリッシャ [1](#)
- V\$BUFFERED_SUBSCRIBERS
 - インスタンスにあるすべてのバッファ済キューのサブスクライバ [1](#)
- V\$METRICGROUP
 - メトリック・グループに関する情報 [1](#)
- V\$PERSISTENT_PUBLISHERS
 - データベースにある永続キューのすべてのアクティブなパブリッシャ [1](#)
- V\$PERSISTENT_QMN_CACHE
 - 永続キューのバックグラウンド・タスクのパフォーマンス統計 [1](#)
- V\$PERSISTENT_QUEUES

- データベースにあるすべてのアクティブな永続キュー [1](#)
- V\$PERSISTENT_SUBSCRIBERS
 - データベースにある永続キューのすべてのアクティブなサブスクライバ [1](#)
- V\$PROPAGATION_RECEIVER
 - 受信(宛先)側のバッファ・キュー伝播スケジュール [1](#)
- V\$PROPAGATION_SENDER
 - 送信(ソース)側のバッファ・キュー伝播スケジュール [1](#)
- V\$SUBSCR_REGISTRATION_STATS
 - 通知の診断性 [1](#)
- 検証
 - メッセージ・ゲートウェイの設定 [1](#)
 - キュー・タイプ [1](#)
- ビュー
 - すべての伝播スケジュール [1](#)
 - データベース内のすべてのキュー [1](#)
 - データベース内の全キューのサブスクライバ [1](#)
 - すべての変換ファンクション [1](#)
 - すべての変換 [1](#)
 - インターネット・アクセスが登録されたAQエージェント [1](#)
 - デキュー・アフィニティ・インスタンス・リスト [1](#)
 - シャード・キューのメモリー管理 [1](#)
 - キュー表内のメッセージ [1](#)
 - メッセージ・ゲートウェイ [1](#)
 - メッセージ・ゲートウェイ・エージェント [1](#)
 - メッセージ・ゲートウェイ・リンク [1](#)
 - ユーザー・スキーマの伝播スケジュール [1](#)
 - ユーザーがなんらかの権限を持っているキュー [1](#)
 - ユーザーがキュー権限を持っているキュー [1](#)
 - ユーザー・スキーマのキュー [1](#)
 - キューのサブスクライバ [1](#)
 - キューのサブスクライバおよびそのルール [1](#)
 - ユーザーがキュー権限を持っているキューのサブスクライバ [1](#)
 - ユーザー・スキーマのキューのサブスクライバ [1](#)
 - ユーザー・スキーマのキュー表 [1](#)
 - Oracle以外の登録済キュー [1](#)
 - シャード・キュー・サブスクライバの統計 [1](#)
 - シミュレートされたメトリック [1](#)
 - ユーザー変換 [1](#)
- 仮想プライベート・データベース
 - 制限事項 [1](#)
- 可視性
 - 概要 [1](#), [2](#)
 - バッファ済メッセージ [1](#)
 - デキュー・オプション [1](#)
 - エンキュー・オプション [1](#)

- [ロールバック操作 1](#)
-

W

- 待機
 - [デキュー中 1](#)
- WebSphere MQ
 - [Base Javaリンク, 作成 1](#)
 - [Base Javaキュー](#)
 - [登録 1](#)
 - [登録解除 1](#)
 - [JMS_DeliveryMode 1](#)
 - [JMS_NoLocal 1](#)
 - [JMSリンク, 作成 1](#)
 - [JMSキューまたはトピック](#)
 - [登録 1](#)
 - [登録解除 1](#)
 - [listener.ora, 変更 1](#)
 - [メッセージの変換 1](#)
 - [メッセージ・ヘッダーのマッピング 1](#)
 - [メッセージ・ゲートウェイ, 設定 1](#)
 - [mgw.ora, 変更 1](#)
 - [MQ_BrokerControlQueue 1](#)
 - [MQ_BrokerPubQueue 1](#)
 - [MQ_BrokerQueueManager 1](#)
 - [MQ_BrokerVersion 1](#)
 - [MQ_ccsid 1](#)
 - [MQ_CharacterSet 1](#)
 - [MQ_JmsDurSubQueue 1, 2](#)
 - [MQ_JmsTargetClient 1](#)
 - [MQ_openOptions 1](#)
 - [MQ_PubAckInterval 1](#)
 - [MQ_ReceiveExit 1](#)
 - [MQ_ReceiveExitInit 1](#)
 - [MQ_SecurityExit 1](#)
 - [MQ_SecurityExitInit 1](#)
 - [MQ_SendExit 1](#)
 - [MQ_SendExitInit 1](#)
 - [MsgBatchSize 1](#)
 - [オプションのリンク構成プロパティ 1](#)
 - [PreserveMessageID 1](#)
 - 伝播
 - [インバウンド 1](#)
 - [アウトバウンド 1](#)

- システム・プロパティ [1](#)
-

X

- XA
 - Oracle Database AQとともに使用 [1](#)
- XML [1](#)
 - AQサブレットのデプロイ [1](#)
 - メッセージ・フォーマットの変換 [1](#)