



Active Messaging ユーザーズ・ガイド

Adaptive Server® Enterprise

15.7

ドキュメント ID : DC01277-01-1570-01

改訂 : 2011 年 9 月

Copyright © 2012 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

Sybase の商標は、**Sybase trademarks ページ** (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

Java および Java 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

IBM および Tivoli は、International Business Machines Corporation の米国およびその他の国における登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

第 1 章	概要	1
	Active Messaging の概念	1
	リアルタイムの自動決定機能	2
	メッセージング・モデル	3
	JMS	3
	WebSphere MQ メッセージング・モデル	3
	メッセージ・フォーマット	4
	JMS のメッセージ・プロパティ	4
	MQ メッセージ・トピック	5
	メッセージ・セレクタ	5
第 2 章	Active Messaging について	7
	キューとの間のメッセージの送受信	7
	JMS トピックとのメッセージのパブリッシュとコンシューム	8
	メッセージ・プロパティの使用	9
	メッセージング・インタフェースのプレビュー	9
	MQ の概要	11
	SSL を使用したチャネルのセキュリティ保護	14
	MQ パブリッシュ/サブスクライブ	15
	トピックの構文	18
	パブリッシャとサブスクライバの ID	19
	MQ パブリッシュ/サブスクライブの例	20
	MQ クラスタ・キュー・オブジェクトの使用	26
	リモート・キュー・オブジェクトの使用	27
	テキスト・メッセージングの使用	28
	テキスト・メッセージと JMS	28
	テキスト・メッセージと MQ	29
	Adaptive Server Cluster Edition のサポート	30
	ログインのリダイレクト	30
	拡張高可用性	31
	スレッド・カーネルに対する Active Messaging のサポート	32
	国際化のサポート	32
	トランザクション・メッセージの動作	33
	トランザクション・メッセージングの設定オプション	34

	MQ のセキュリティ	35
	MQ キュー・マネージャへの接続	35
	Adaptive Server ホスト・マシンへの MQ クライアントのインストール	36
	MQ の権限	37
	MQ 情報のクエリ	37
第 3 章	SQL リファレンス	39
	メッセージ関連のグローバル変数	40
	<msgheader> と <msgproperties> ドキュメント	47
	Adaptive Server 固有のメッセージ・プロパティ	48
	キーワード	49
	ストアド・プロシージャ	50
	組み込み関数	50
	構文セグメント	51
	sp_configure 'enable real time messaging'	51
	sp_engine	55
	sp_msgadmin	59
	msgconsume	71
	msgpropcount	74
	msgproplist	75
	msgpropname	77
	msgproptype	78
	msgpropvalue	80
	msgpublish	81
	msgrecv	85
	msgsend	99
	msgsubscribe	133
	msgunsubscribe	135
	endpoint	138
	option_string	141
	sizespec	142
	timespec	143
第 4 章	サンプル	145
	Sybase ディレクトリ	145
	SQL を使用したコード・サンプルの使用	146
	Java/JDBC を使用したコード・サンプルの使用	146
	用語解説	147
	索引	151

概要

このマニュアルでは、読者がデータベース管理におけるメッセージング・システムの基礎知識を持っていることを前提としていますが、この章では基本的なメッセージングの概念とモデルを紹介し、マニュアル全体で使用されている用語の一部について説明します。

説明の大部分は、Adaptive Server[®] に固有のメッセージング機能に関連しています。このマニュアルでは、この機能をリアルタイム・メッセージング・サービスと呼びます。

トピック名	ページ
Active Messaging の概念	1
リアルタイムの自動決定機能	2
メッセージング・モデル	3
メッセージ・フォーマット	4
メッセージ・セクタ	5

Active Messaging の概念

メッセージングとは、2つ以上のソフトウェア・アプリケーションによる情報の交換のことです。メッセージは、独立した情報のパッケージです。

多くの Adaptive Server ユーザが固有のアプリケーション環境でメッセージングおよびキューイング・システムまたはパブリッシュ/サブスクライブ・システムを使用しています。これらのアプリケーションは、メッセージ指向のミドルウェアと呼ばれています。多くの場合、同じアプリケーションでデータベース・オペレーションとメッセージング処理を結合しています。

Tibco Enterprise Message Service (EMS)、EAServer Java Messaging System (JMS)、IBM WebSphere MQ (MQ)、Sonic Software の SonicMQ を使用している Adaptive Server では、Active Messaging によってこのようなアプリケーションの開発が容易になります。

注意 EMS は Tibco による JMS (Java Messaging Service) 実装です。このマニュアルでは、特に明記されていない限り、JMS は一般に Tibco EMS、EAServer JMS、SonicMQ JMS を指します。

メッセージング・システムでは、送信者と受信者を分離できます。受信側のアプリケーションが実行されていなくても、アプリケーションからメッセージを送信できるという点で、メッセージング・システムは非同期システムとして機能します。

JMS と MQ は、クライアントとメッセージ・プロバイダとの通信方法を定義する API です。メッセージの送信者と受信者は両方とも、メッセージ・プロバイダに対してクライアントとして機能します。

メッセージング・システムはメッセージ・プロバイダが提供します。メッセージング・プロバイダは、メッセージの記憶領域を集中化または分散するアーキテクチャ、あるいはその複合型のアーキテクチャを実装できます。

Active Messaging では、組み込み関数を使用して、SQL 文でメッセージング処理を実行します。

また、Adaptive Server データベースでのトランザクション (データ変更) を取得し、次のような方法で外部アプリケーションにイベントとして渡すこともできます。

- Tibco, EA Server、または Sonic Software が提供している JMS メッセージ・バス
- WebSphere MQ が提供しているメッセージ・キュー・インタフェース (MQI: Message Queue Interface)

リアルタイムの自動決定機能

データベースの管理において、特定のイベントに対応するリアルタイムの自動決定機能が必要な場合があります。リアルタイムとは、イベントが発生したときに、イベントをキューに入れるのではなく、データベースがイベントに関して決定できるということです。レコードの変更などのイベントは、他の変更との関連で評価され、最も効率のよい応答が選択される必要があります。したがって、有効な意思決定支援システムには、次の要素が必要です。

- 遅延時間が少ないこと。リアルタイムのエンタープライズ・システムを可能にします。
- イベントおよびそれに関連するデータを記述する自動システム。
- アプリケーションのコストを低減する技術。遅延時間を少なくします。

これらのビジネス・ニーズは、Tibco, EA Server JMS メッセージ・バス、または IBM WebSphere MQ を使用して Active Messaging で解決できます。

メッセージング・モデル

この項では、JMS と MQ のメッセージング・モデルについて説明します。

JMS

JMS では、2 つのメッセージング・モデルを定義しています。

- [パブリッシュ/サブスクライブ \(トピック\)](#)
- [ポイントツーポイント \(キュー\)](#)

パブリッシュ/サブスク ライブ (トピック)

パブリッシュ/サブスクライブ (pub/sub) モデルは、1 対多モデルです。メッセージを送信するアプリケーションを「メッセージ・プロデューサ」と呼び、メッセージを受信するアプリケーションを「メッセージ・コンシューマ」と呼びます。メッセージ・コンシューマは、トピックに送信されるメッセージに興味を示すためのサブスクリプションを設定します。トピックは、このメッセージ・モデルの送信先です。

pub/sub モデル内に設定できるサブスクリプションは 2 種類あります。

- **持続的** – メッセージ・コンシューマ・アプリケーションが接続されていないときでもメッセージ・コンシューマのためにメッセージを保持します。メッセージを保持するのは、Adaptive Server ではなくメッセージ・プロバイダです。
- **非持続的** – コンシューマ・アプリケーションがメッセージ・プロバイダに接続されているときのみメッセージを保持します。

ポイントツーポイント (キュー)

「メッセージ送信者」と呼ばれるアプリケーションによって送信されたメッセージを読むことができるのが「メッセージ受信者」と呼ばれる受信側アプリケーションだけであるという点で、ポイントツーポイント・モデルは 1 対 1 モデルです。ポイントツーポイント・メッセージの送信先はキューです。キューには、複数の Active Message 受信者が含まれている場合がありますが、メッセージング・プロバイダはメッセージが 1 つのメッセージ受信者だけに渡されるようにします。

WebSphere MQ メッセージング・モデル

すべての MQ メッセージング・モデルはポイントツーポイントです。つまり、メッセージは常にキュー・マネージャによって管理されているキューから送受信されます。

MQ pub/sub は、MQ キューをベースにしたパブリッシュ/サブスクライブ・モデルであり、メッセージは異なるタイプのオブジェクトではありません。MQ pub/sub との対話では、MQ キューが使用されます。

すべてのメッセージが MQ pub/sub **ブローカ**のブローカ・コマンド・キューに送信されます。これらのメッセージには、パブリッシャまたはサブスクライバの登録や、メッセージの削除やメッセージの更新要求などの制御メッセージが含まれます。

パブリッシャはストリーム・キューにパブリケーションを送信します。MQ pub/sub ブローカは、メッセージに興味があるすべてのサブスクライバにメッセージを配布します。パブリッシャは、トピックを使用してメッセージを説明します。これは、メッセージの内容を説明するサブジェクトです。

サブスクライバは、興味のあるトピックを1つ以上指定することによって、特定のストリーム・キューに送信されるメッセージに興味を示します。このようなメッセージがストリーム・キューに送信されたら、MQ pub/sub ブローカは、サブスクライバの登録時にサブスクライバが指定したローカル・キューにメッセージをコピーします。

メッセージ・フォーマット

MQ および JMS のメッセージ・フォーマットは次のように構成されます。

- **メッセージ・ヘッダー** – 標準によって指定された情報の固定サイズの部分と可変サイズの部分から構成されています。この情報のほとんどは、メッセージ・プロバイダによって自動的に割り当てられます。
- **メッセージ本文** – クライアント・アプリケーションが交換するアプリケーション・データ。

JMS では、**stream** や **map** などの構造化メッセージ・タイプと、**text**、**byte**、**object** などの非構造化メッセージ・タイプを定義しています。

MQ では、メッセージ本文にテキストとバイナリ・データの両方を格納できます。

JMS のメッセージ・プロパティ

Tibco、EAServer、Sonic MQ のメッセージ・プロパティは、メッセージと共に指定できるユーザ定義プロパティです。メッセージ・プロパティには、アプリケーション固有の情報を定義する型があります。メッセージ・コンシューマは、後でこの型を使用して、興味のあるメッセージを選択できます。メッセージ・プロパティの型は、Java ネイティブ型 **int**、**float**、または **String** (クラス) です。

MQ メッセージ・トピック

MQ の pub/sub モデルでは、「トピック」を使用できます。これは、メッセージのサブジェクトです。トピックは、メッセージのルールとフォーマット (RF) ヘッダーに含まれています。JMS とは異なり、MQ のトピックは名前と値のペアではありません。名前とそれに付随する値から構成されていますが、MQ pub/sub メッセージを説明する自由な形式の文字列です。

メッセージ・セレクタ

Tibco および EAServer の JMS メッセージ・セレクタを使用して、メッセージ・コンシューマはメッセージ・ストリームをフィルタ処理し、興味のあるメッセージを選択できます。これらのフィルタは、メッセージ・プロパティとその値を参照する基準を適用できます。メッセージ・セレクタは、SQL-92 `where` 句です。

MQ メッセージの選択では、メッセージ・セレクタとしてメッセージ ID とメッセージの相関 ID のみを使用します。メッセージ・リーダは、メッセージ ID またはメッセージの相関 ID を指定することによって、特定のメッセージを選択して読むことができます。

この章では、Adaptive Server をメッセージ・プロバイダのクライアントに使用できる、Adaptive Server® に固有の Active Messaging の概要について説明します。メッセージ・プロバイダとの間のメッセージの送信と受信には、Transact-SQL™ を使用します。

トピック名	ページ
キューとの間のメッセージの送受信	7
JMS トピックとのメッセージのパブリッシュとコンシューム	8
メッセージ・プロパティの使用	9
メッセージング・インタフェースのプレビュー	9
MQ の概要	11
SSL を使用したチャネルのセキュリティ保護	14
MQ パブリッシュ/サブスクライブ	15
MQ クラスタ・キュー・オブジェクトの使用	26
リモート・キュー・オブジェクトの使用	27
テキスト・メッセージングの使用	28
Adaptive Server Cluster Edition のサポート	30
国際化のサポート	32
トランザクション・メッセージの動作	33
MQ のセキュリティ	35
Adaptive Server ホスト・マシンへの MQ クライアントのインストール	36
MQ の権限	37
MQ 情報のクエリ	37

キューとの間のメッセージの送受信

Transact-SQL アプリケーションは、組み込み関数 `msgsend` 関数と `msgrecv` 関数を使用して、キューにメッセージを送信したり、JMS および MQ のキューからメッセージを読み込んだりすることができます。

アプリケーション・ロジックを使用してメッセージ本文またはペイロードを作成するか、またはリレーショナル・テーブルの文字またはバイナリ・データから直接本文を構成することができます。

リレーショナル・データまたはアプリケーション・ロジックからメッセージ・プロパティの値 (ヘッダまたはユーザ・プロパティ) を構成し、構成したメッセージ・プロパティを送信するメッセージに含めることができます。

JMS または MQ キューから読み込んだメッセージはアプリケーション・ロジックで処理するか、リレーショナル・テーブルに直接挿入することができます。読み込み操作を実行したときに必要なメッセージのみをフィルタするには、メッセージ・セレクタを指定します。

読み込みメッセージのメッセージ・プロパティはアプリケーション・ロジックで個別に処理できます。メッセージ・プロパティの詳細については、「[msgsend](#)」(99 ページ) を参照してください。

JMS トピックとのメッセージのパブリッシュとコンシューム

Transact-SQL アプリケーションは、組み込み関数 `msgpublish` 関数と `msgconsume` 関数を使用して JMS トピックとの間でメッセージをパブリッシュまたはコンシュームできます。

まず、`sp_msgadmin 'register'` を使用してサブスクリプションを登録します。サブスクリプションを登録すると、`msgpublish`、`msgconsume`、`msgsubscribe`、`msgunsubscribe` 関数が参照できる名前が作成されます。サブスクリプションを**持続的**、または**非持続的**として登録し、メッセージ・セレクタを指定して受け取るメッセージを制御すると、必要なメッセージのみを読み込むようにできます。

アプリケーション・ロジックで処理する準備ができるまでメッセージを保持するように JMS プロバイダに指示する場合に `msgsubscribe` を使用します。`msgunsubscribe` を使用すると、このサブスクリプションのメッセージがアプリケーションで不要になったことを JMS プロバイダに通知できます。`msgunsubscribe` では、持続的なサブスクリプションを JMS プロバイダから削除することもできます。

読み込みメッセージのメッセージ・プロパティはアプリケーション・ロジックで個別に処理できます。

`sp_msgadmin` と関数の構文、パラメータおよび使用方法については、「[第 3 章 SQL リファレンス](#)」を参照してください。

メッセージ・プロパティの使用

メッセージを読み込んだら、組み込みの SQL 関数を使用して、Transact-SQL アプリケーション・ロジックでメッセージ・ヘッダとユーザ・プロパティを処理することができます。これらの関数は次の値を返します。

- n 番目のプロパティの名前
- 名前付きプロパティの値
- 名前付きプロパティのタイプ
- プロパティの数
- プロパティのリスト

これらの組み込み関数を使用すると、メッセージ・プロパティの値に基づいて、アプリケーション・ロジックで実行時に処理決定を下すことができます。

- msgproplist
- msgpropname
- msgpropvalue
- msgproptype
- msgpropcount

メッセージング・インタフェースのプレビュー

以下の例は、Transact-SQL メッセージング・インタフェースの概要を示します。

例

例 1 (JMS) メッセージをキューに送信します。

```
select msgsend('hello world',
('eas_jms:iiop://my_eas:7222?queue=queue.sample'
message property 'city=Detroit')
```

例 2 (JMS) キューからメッセージを読み込みます。フィルタを使用する場合と使用しない場合を示します。

```
select msgrecv('tibco_jms:tcp://my_jms_host:7222?queue=queue.sample')

select msgrecv
('eas_jms:iiop://my_eas:7222?queue=queue.sample'
message selector 'city=''Detroit''')
```

例 3 (JMS) メッセージをトピックにパブリッシュします。

```
sp_msgadmin register, subscription,sub1,
  'eas_jms:iiop://my_eas:7222?topic=topic.sample'
select msgpublish
  ('hello world', 'sub1' message property 'city=Boston')
```

例 4 (JMS) トピックのメッセージをコンシュームします。

```
select msgconsume('sub1')
```

例 5 (JMS) プロパティの処理を示します。

```
select msgconsume('sub1')
declare @pcount integer
declare @curr integer
declare @pname varchar(100)
select @curr=1
select @pcount = msgpropcount()
while(@curr<=@pcount)
begin
  select @pname=msgpropname(@curr)
  select msgproptype(@pname)
  select msgpropvalue(@pname)
  select @curr=@curr+1
end
```

例 6 (MQ) メッセージをキューに送信します。

```
select msgsend('hello world',
  'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=DEFAULT.QUEUE'
  message header 'priority=2')
```

例 7 (MQ) メッセージをキューから読み込みます。

```
select msgrecv(
  'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=DEFAULT.QUEUE'
  option 'timeout=30ss')
```

例 8 (MQ) パブリッシャを登録し、魚に関するメッセージをパブリッシュします。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEUEi'
  option 'rfhCommand=registerPublisher'
  message header 'topics=fish'
  + ',streamName=ANIMALS.STREAM')
select msgsend('something about a fish',
  'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=ANIMALS.STREAM'
  message header 'topics=fish')
```

例 9 (MQ) サブスクライバを登録し、メッセージを読み込んで、メッセージ・プロパティを処理します。

```
select msgsend(NULL,  
  'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEUE'  
  option 'rfhCommand=registerSubscriber'  
    + ',topics=fish'  
    + ',streamName=ANIMALS.STREAM'  
    + ',queueName=MY_ANIMALS.QUEUE')  
  
select msgrecv(  
  'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=MY_ANIMALS.QUEUE'  
  option 'timeout=30ss')  
  
select msgpropvalue('MPQScompcode', @@msgproperties)
```

MQ の概要

IBM WebSphere MQ を使用すると、さまざまなアプリケーションでキューを使用して、各種のオペレーティング・システム、プロセッサ、アプリケーション・システムで非同期通信を行うことができます。

WebSphere MQ には、メッセージ・キュー・インタフェース (MQI)、共通の低レベル アプリケーション・プログラム・インタフェース (API) があります。アプリケーションは MQI を使用して、キューとのメッセージの送受信を行います。

キュー・マネージャは、キューイング・サービスを提供し、WebSphere MQ で使用されるリソースのセットを所有および管理するシステム・プログラムです。これらのリソースには、キュー、チャンネル、プロセス定義などがあります。

キューはメッセージを保存するために使用されるデータ構造体です。WebSphere MQ では数種類のキュー・オブジェクトを使用できます。

- ローカル・キュー・オブジェクト – アプリケーションが接続されているキュー・マネージャに属するローカル・キューを識別します。各キューがキュー・マネージャに属し、そのキュー・マネージャに対してキューがローカル・キューになるという点で、すべてのキューはローカル・キューということができます。
- リモート・キュー・オブジェクト – アプリケーションが接続されているキュー・マネージャとは異なる、別のキュー・マネージャに属するキューを識別します。このキューは、リモート・キュー・オブジェクトが属するキュー・マネージャに対するローカル・キューとして定義する必要があります。
- エイリアス・キュー・オブジェクト – キューではありませんが、ローカル・キューまたはリモート・キューのオブジェクト・ポインタです。
- モデル・キュー・オブジェクト – 動的キューを作成するテンプレートとして使用するキュー属性のセットを定義します。

すべてのタイプのキュー・オブジェクトをメッセージで送信できますが、メッセージを読み込めるのはローカル・キュー・オブジェクトからのみです。

WebSphere MQ で使用可能なキュー・オブジェクト・タイプ以外にも、キューにはいくつかの概念があります。

- リモート・キュー定義 – 別のキュー・マネージャによって所有されているキューの定義です。キュー自体を指すものではありません。

リモート・キュー定義を使用すると、リモート・キュー、リモート・キュー・マネージャ、または転送キューの名前を指定せずに、メッセージをリモート・キューに入れることができます。

- 定義済みキュー – 管理者が適切な MQ シリーズ・コマンド (MQSC) または WebSphere MQ プログラム可能なコマンド・フォーマット (PCF) コマンドを使用して作成します。定義済みキューは永続的で、使用するアプリケーションに関係なく存在し、WebSphere MQ が再起動を行っても続きます。
- 動的キュー – アプリケーションが MQOPEN 要求を出し、モデル・キューの名前を指定した場合に作成されます。作成されるキューはテンプレート・キュー定義に基づくもので、モデル・キューと呼ばれます。動的キューの属性は、作成元となるモデル・キューから継承されます。
- クラスタ・キュー・オブジェクト – クラスタ・キュー・マネージャによってホストされ、クラスタ内の他のキュー・マネージャが使用できます。

チャネルは WebSphere MQ クライアントと WebSphere MQ サーバ間、または 2 台の WebSphere MQ サーバ間の論理通信リンクです。WebSphere MQ には 2 つのカテゴリのチャネルがあります。

- メッセージ・チャネル – メッセージ・チャネル・エージェント経由で 2 つのキュー・マネージャを接続する一方通行リンクです。
- MQI チャネル – WebSphere MQ クライアントをサーバ・マシン上のキュー・マネージャに接続します。MQCONN または MQCONNX 呼び出しを発行したときに確立されます。MQ チャネルは MQI 呼び出しと応答のみを転送するための双方向リンクです。

MQI チャネル定義には 2 つのチャネル・タイプがあります。

- クライアント接続チャネル – WebSphere MQ クライアントに接続します。
- サーバ接続チャネル – WebSphere MQ クライアント環境で実行されている WebSphere MQ アプリケーションと通信するキュー・マネージャを実行しているサーバに接続します。

MQ チャネルは業界標準の Secure Sockets Layer (SSL) プロトコルをサポートしています。お使いのプラットフォームの MQ のバージョン 5.3 または 6.0 で SSL が使用可能かどうかについては、IBM から提供されている WebSphere MQ のマニュアルを参照してください。

プロセス定義は、入力メッセージによってトリガ・イベントが発生したときに実行されるプロセスを定義します。

WebSphere MQ メッセージは次の 2 つの部分で構成されます。

- メッセージ・ヘッダ – 固定サイズの部分と可変サイズの部分から構成されるメッセージ制御情報。
- メッセージ本文 – 任意のタイプのデータで構成されるアプリケーション・データ (text または binary)。

`rfhCommand` を使用してパブリケーションをパブリッシュしたときに、`msgrcv` によって返されるメッセージ・ペイロードが次の値に設定されている場合、次の意味があります。

- MQRHRF – RF ヘッダがメッセージ本文に含まれています。
- MQRHRH – RF ヘッダがメッセージ本文に含まれていません。

`@@msgproperties` を問い合わせ、RF ヘッダの名前と値のペアを取得することができます。

メッセージ本文に文字が含まれる場合、MQ ネイティブ・サービスまたはユーザ終了ハンドラのいずれかを通じてコードセット変換を使用することができます。メッセージ本文のフォーマットは、メッセージ・ヘッダのフィールドによって定義されます。MQ ではいくつかのフォーマットがサンプルとして提供されていますが、使用可能なすべてのメッセージ本文フォーマットが列挙されるわけではありません。アプリケーションでは任意の名前のフォーマットを入力できます。たとえば、“MQSTR”には文字列データが含まれ、“MQRHRF”には MQ pub/sub のトピックが含まれています。

WebSphere MQ には、次のメッセージ・タイプがあります。

- データグラム – 想定される応答はありません。
- 要求 – 応答が期待されます。
- 応答 – 要求メッセージに対する応答。
- レポート – キュー・マネージャまたは別のアプリケーションのステータス情報が含まれています。

メッセージの送信時には、有効期限、持続性、優先度、相関 ID、応答キューなどさまざまなメッセージ・ヘッダ・プロパティを設定できます。

メッセージをグループ分けすると、メッセージのグループを論理名グループに編成することができます。グループ内で、各論理メッセージをさらにセグメントに分割することができます。グループは名前で識別され、グループ内の各論理メッセージは 1 から始まるシーケンス番号によって識別されます。論理メッセージの各セグメントは、論理メッセージに関連したメッセージ・データのオフセットによって識別されます。セグメント化されたメッセージは MQ pub/sub でサポートされていないため、セグメント化されたメッセージを送信しようとする、エラーが発生します。

キュー内では、メッセージはキューに送信される物理的な順序で表示されます。つまり、異なるグループのメッセージが分散して表示されたり、グループ内のメッセージのシーケンス番号の順序が揃わないことがあります (後者は、2つのアプリケーションが同じグループ ID でシーケンス番号が分割されたメッセージを送信している場合に生じることがあります)。

メッセージを受信したときの読み込みモードは次のいずれかです。

- 破壊的 – メッセージは削除されます。
- 非破壊的 – メッセージは保持されます。これは「ブラウズ」として知られるもので、特定のメッセージをキューから削除する前に、1つまたは複数のメッセージをアプリケーションで詳しく調べることができます。

受信者は、相関 ID またはメッセージ ID などのメッセージ・ヘッダ・プロパティを指定することによって、特定のメッセージを選択することができます。

破壊的、非破壊的のいずれかでメッセージを読み込んだとき、返される順序を物理的または論理的に設定することができます。順序はキュー定義によって定義されます。キューは優先度順、または先入れ先出し順に定義できます。

SSL を使用したチャネルのセキュリティ保護

SSL を使用してメッセージを送受信するには、次の手順に従います。

- 1 キュー・マネージャのプライベート・キーが含まれる、接続されたキュー・マネージャのキー・レポジトリ、および Adaptive Server のデジタル証明書を作成します。
- 2 その Adaptive Server のデジタル証明書が含まれる Adaptive Server、および接続されているキュー・マネージャのキー・レポジトリを作成します。
- 3 接続されているキュー・マネージャ上に SSL 対応のサーバ接続チャネルを作成します。
- 4 「第3章 SQL リファレンス」の「[sp_msgadmin](#)」(59 ページ)で説明した、`sp_msgadmin 'config', 'ibmmq_keystore'` ストアド・プロシージャを使用して、Adaptive Server のキー・レポジトリを設定します。

例 このシナリオでは、WebSphere MQ が Active Messaging で SSL を使用して通信した場合と SSL を使用せずに通信した場合を示します。

キュー・マネージャ “BACH” には2つのサーバ接続チャネルがあります。最初の “CH1” は通常の接続ですが、“CH2” は SSL を必要とするように設定されています。チャネルの SSL 暗号仕様は NULL_MD5 です。

- 1 SSL を有効にせずにキュー・マネージャにメッセージを送信します。

```
select msgsend('a', 'ibmq:CH1/tcp/host1(7654)?qmgr=BACH,queue=Q1')
```

- 2 次に、SSL プロトコルを使用してキュー・マネージャにメッセージを送信します。
 - a キュー・マネージャと Adaptive Server に別々にキー・レポジトリを設定します。Adaptive Server のキー・データベース・ファイルは `/var/mqm/clients/ssl/ASE.kdb` です。キー・レポジトリの設定方法については、IBM から提供されている WebSphere MQ のマニュアルを参照してください。
 - b Adaptive Server のキー・レポジトリを次のように設定します。

```
sp_msgadmin 'config', 'ibmmq_keystore', '/var/mqm/clients/ssl/ASE'
```

- c SSL を使用してメッセージを送信します。

```
select msgsend('e', 'ibm_mq:CH2(ssl:sslcip=NULL_MD5)
/tcp/host1(7654)?qmgr=BACH,queue=Q1')
```

MQ パブリッシュ／サブスクライブ

WebSphere MQ パブリッシュ／サブスクライブは、ブローカ処理を使用してサブスクリプション解析を実行する MQ キュー上で使用されます。最も簡単に説明すると、次のようになります。

- パブリッシャはメッセージを送信するアプリケーションです。
- サブスクライバはメッセージを受信するアプリケーションです。
- 次のキューが関係しています。
 - コントロール・キュー – パブリッシャとサブスクライバが pub/sub ブローカにサブスクライバ登録およびキャンセルなどのディレクティブを送信します。
 - ストリーム・キュー – パブリッシャがメッセージを直接送信します。pub/sub ブローカはストリーム・キューからメッセージを読み込み、適切なサブスクライバのキューに配布します。
 - サブスクライバ・キュー – サブスクライバがメッセージを直接読み込みます。

注意 アプリケーションのタイプによっては、さらに多くのキューが関与することもあります。

- pub/sub ブローカは、コントロール・キューに送信された MQRFH メッセージに応答します。これらのコマンド・メッセージは、ストリーム・キューに到着したメッセージを pub/sub ブローカが処理する方法を制御します。たとえばサブスクライバは、特定のトピックの関心事項を登録できます。

- パブリッシャはストリーム・キューにメッセージを直接送信します。
- pub/sub ブローカはストリーム・キューからメッセージを読み込み、メッセージをコピーするサブスクライバ・キューを判断します。これはサブスクライバで関心事項が登録されたトピックによって異なります。
- サブスクライバはサブスクライバ・キューからメッセージを直接読み込みます。

サブスクライバは「サブスクリプション」を登録します。サブスクリプションが登録されると、1つ以上の「トピック」に関心があることになります。

例

MQ pub/sub プロセスを示す次の例では、次の変数を使用します。

```
declare @BROKER      varchar(100)
declare @STREAM     varchar(100)
declare @SUBQ       varchar(100)
declare @QM         varchar(100)
select @QM          = 'ibm_mq:channel1/tcp/host1(9876)?qmgr=QM'
select @BROKER      = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @STREAM      = 'ANIMALS'
select @SUBQ        = 'MY_ANIMALS'
```

- 1 パブリッシャは魚に関するトピックを持つ「ANIMALS」(動物)のパブリケーションを送信するように登録します。

```
select msgsend(NULL,
  @QM + ',queue=' + @BROKER
  option 'rfhCommand=registerPublisher'
  message header 'topics=fish,streamName=' + @STREAM)
```

- 2 サブスクライバは魚に関するトピックを持つ「ANIMALS」(動物)のパブリケーションを受信するように登録します。サブスクライバは MY_ANIMALS に関するパブリケーションを受け取ります。

```
select msgsend(NULL,
  @QM + ',queue=' + @BROKER
  option 'rfhCommand=registerSubscriber'
  message header 'topics=fish'
  + ',streamName=' + @STREAM
  + ',queueName=' + @SUBQ')
```

- 3 パブリッシャは魚に関する「ANIMALS」(動物)のパブリケーションをパブリッシュします。MQ pub/sub ブローカは MY_ANIMALS のパブリケーションを自動的に転送します。

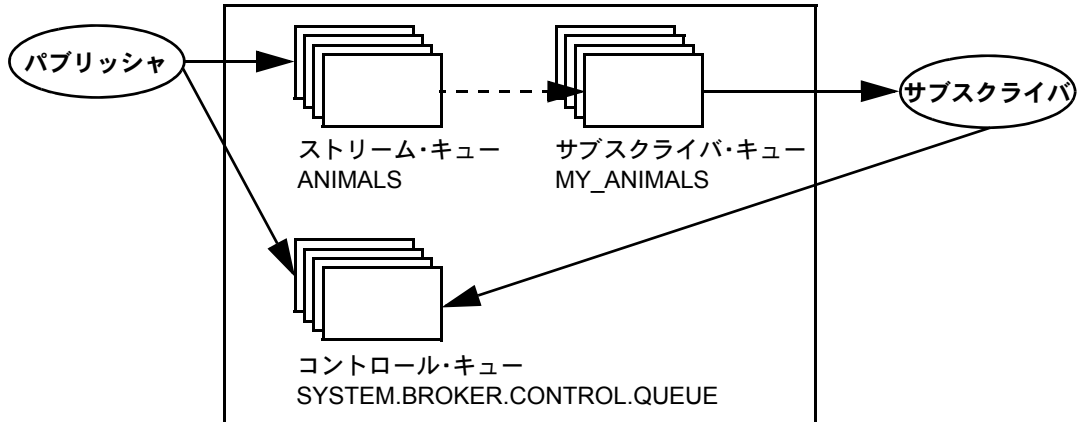
```
select msgsend('something about fish',
  @QM + ',queue=' + @STREAM
  option 'rfhCommand=publish'
  message header 'topics=fish')
```

4 サブスクライバは MY_ANIMALS から転送されたメッセージを読みます。

```
select msgrecv(@QM + ',queue=' + @SUBQ option 'timeout=30ss')
```

図 2-1 に MQ pub/sub プロセスの例を示します。

図 2-1: MQ パブリケーション/サブスクリプションプロセス



1つのメッセージは1つまたは複数のトピックを持つことができます。WebSphere MQ pub/sub モデルでは、トピックで以下の例に示すような階層的な命名規則を使用することをお勧めします。サブスクライバでは関心事項トピックを指定するときに*や?などのワイルドカードを指定できます。

次は、トピックの例です。

```
Sport
Sport/Soccer
Sport/Tennis
```

次に、サブスクライバが関心事項のトピックを指定する方法の例を示します。

```
Sport/*          - Any topic about sports.
*/Soccer       - Any topics about soccer.
*/Soccer/Trades - Any topics about soccer where a 'trade' is involved.
```

保持されたパブリケーションは、MQ pub/sub ブローカがすべてのサブスクライバにメッセージを配信した後もメッセージのコピーを保持するパブリケーションのタイプです。パブリケーションは、コピーがすべてのサブスクライバに配信された後で削除されるのが普通です。保持されたパブリケーションを使用すると、サブスクライバは、MQ pub/sub ブローカによる配信に頼らずに、保持されているパブリケーションを非同期で要求することができます。これらのタイプのメッセージには通常ステータス情報が含まれており、ステータス・パブリケーションとも呼ばれます。

トピックの構文

- トピックは通常、「sport/baseball」などのように、「トピック/サブトピック」の形式を取ります。
- トピック内で“*”や“?”などのワイルドカードを指定できます。
- 複数のトピックを指定する際は、トピックをコロンで区切ります。たとえば、“topic1:topic2:topic3:”などのように区切ります。
- トピックにスペースやカンマが含まれている場合は、トピック リスト全体を引用符で囲みます。トピックは **message header** 句や **message property** 句に文字列として表示されることがあるため、オプションの文字列が引用符付きのスカラ値として渡された場合、囲んだ引用符を二重にしてエスケープする必要があります。さらにトピックに二重引用符が埋め込まれている場合は、埋め込まれた二重引用符を四重引用符でエスケープする必要があります。次に例を示します。

```
-- Topic has embedded spaces, we need to quote with escaped quotes
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='Sport/Football/Hometown Bulldogs''')

-- Topic has embedded spaces, we can quote with double quotes
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics="Sport/Football/Hometown Bulldogs"')

-- Topic has embedded spaces and embedded double quotes, the inner
-- double quotes need to be escaped.
set quoted_identifier off
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics="quoted ""topic"" here"')

-- Topic has embedded spaces and embedded double quotes, double the
-- quotes around the topic, and quadruple the embedded quotes.
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property "topics=""quoted """"topic"""" here""")
```

- トピックにスペースや引用符が埋め込まれている場合、トピックは MQRF ヘッダの引用符で囲まれています。トピックに引用符が埋め込まれている場合、引用符は MQRF ヘッダに配置される前にエスケープされます。

次の例では、MQRF ヘッダに **“Sport/Football/Hometown Bulldogs”** として配置されているトピックが1つあります。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='Sport/Football/Hometown Bulldogs''')
```

次の例では、MQRF ヘッダに **“Books/”“Recipes Of Spain”** として配置されているトピックが1つあります。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='Books/'Recipes Of Spain''')
```

- トピックは“:”を使用することでエスケープできます。一重でエスケープされていない後続の“:”は無視されます。

次の例には、**“baseball”**、**“baseball/anytown”**、**“baseball/scores”** の3つのトピックがあります。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=baseball:baseball/anytown:baseball/scores')
```

次の例には、**“subject1”**、**“subject:2”**、**“subject3”** の3つのトピックがあります。2つのコロン(“:”)を使用して、埋め込まれた“:”をエスケープします。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=subject1:subject::2:subject3')
```

パブリッシャとサブスクライバの ID

デフォルトでは、パブリッシャまたはサブスクライバの ID の構成要素は次のとおりです。

- キュー名。
- キュー・マネージャ名。
- 相関識別子 (オプション)。相関識別子を使用することで、同じキューを使用する異なるパブリッシャとサブスクライバを区別できます。各パブリッシャとサブスクライバには異なる相関識別子を割り当てることができるため、複数のアプリケーションでキューを共有したり、1つのアプリケーションで異なるサブスクリプションを発信源とするパブリケーションを区別したりすることができます。

MQ パブリッシュ/サブスクライブの例

パブリッシャの例

Adaptive Server セッションがパブリッシャになり、“topicA”と“topicB”をパブリッシュします。“topicB”のパブリケーションは、保持されるパブリケーションとしてパブリッシュされます。保持されたパブリケーションは削除されます。

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER      varchar(100)
-- @STREAM has the stream queue name
declare @STREAM      varchar(100)
-- @CORRELID has the generated correlation id
declare @CORRELID    varchar(100)

-- Put Queue manager name, broker and stream queue names into variables
select @QM          = 'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER      = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @STREAM      = 'Q1.STREAM'

-- Register the publisher, only for topicA
select msgsend(NULL, @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerPublisher'
               message header 'correlationAsId=generate'
                           + ',topics=topicA'
                           + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Save the generated correlation id
select @CORRELID = @@msgcorrelation

-- Send two publications on topicA
select msgsend('topicA, publication 1', @QM + ',queue=' + @STREAM
               option 'rfhCommand=publish'
               message header 'correlationAsId=yes'
                           + ',correlationId=' + @CORRELID
                           + ',topics=topicA')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014803

select msgsend('topicA, publication 2', @QM + ',queue=' + @STREAM
               option 'rfhCommand=publish'
               message header 'correlationAsId=yes'
                           + ',correlationId=' + @CORRELID
                           + ',topics=topicA')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014805

-- Add another topic for this publisher
select msgsend(NULL, @QM + ',queue=' + @BROKER
```



```
option 'rfhCommand=registerPublisher'
message header 'correlationAsId=yes'
              + ',correlationId=' + @CORRELID
              + ',topics=topicB'
              + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014807

-- Publish a retained message on topicB
select msgsend('topicB, retained publication 1', @QM + ',queue=' + @STREAM
              option 'rfhCommand=publish'
              message header 'correlationAsId=yes'
                              + ',correlationId=' + @CORRELID
                              + ',topics=topicB'
                              + ',retainPub=yes')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014809

-- Publish a second retained publication on topicB
-- This one will replace the current retained publication on topicB.
select msgsend('topicB, retained publication 2', @QM + ',queue=' + @STREAM
              option 'rfhCommand=publish'
              message header ',correlationAsId=Yes'
                              + ',correlationId' + @CORRELID
                              + ',topics=topicB'
                              + ',retainPub=yes')
-----
0x414d51204652414e4349532e514d202041a3ebfb2001480b

-- Delete the retained publication on topicB
select msgsend(NULL, @QM + ',queue=' + @STREAM
              option 'rfhCommand=deletePublication'

              message header 'topics=topicB'
                              + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb2001480d

-- Deregister the publisher, for all topics.
select msgsend(NULL, @QM + ',queue=' + @BROKER
              option 'rfhCommand=deregisterPublisher'
              message header 'correlationAsId=yes'
                              + ',correlationId=' + @CORRELID
                              + ',deregAll=yes'
                              + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb2001480f
```

サブスクライバの例

この例では、Adaptive Server セッションが“topicA”と“topicB”をサブスクライブします。“topicB”のパブリケーションは保持されるパブリケーションとしてパブリッシュされます。このサブスクライバは、pub/sub ブローカから更新を要求して、保持されたパブリケーションを処理します。

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER      varchar(100)
-- @SUBQUEUE has the subscriber queue name
declare @SUBQUEUE    varchar(100)
-- @STREAM has the stream queue name
declare @STREAM      varchar(100)
-- @CORRELID has the generated correlation id
declare @CORRELID    varchar(100)

-- Put broker and subscriber queue names into variables
select @QM          = 'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER      = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @SUBQUEUE    = 'Q1.SUBSCRIBER'
select @STREAM      = 'Q1.STREAM'

-- Register the subscriber, only for topicA
select msgsend(NULL, @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerSubscriber'
               message header 'correlationAsId=generate'
                          + ',topics=topicA'
                          + ',streamName=' + @STREAM
                          + ',queueName=' + @SUBQUEUE)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Save the generated correlation id
select @CORRELID = @msgcorrelation

-- Add another topic for this subscriber
-- we will explicitly request update for publications on this topic.
select msgsend(NULL, @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerSubscriber'
               message header 'CorrelationAsId=yes'
                          + ',correlationId=' + @CORRELID
                          + ',topics=topicB'
                          + ',streamName=' + @STREAM
                          + ',queueName=' + @SUBQUEUE
                          + ',pubOnReqOnly=yes')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014803

-- The publisher now publishes messages in the following order:
-- topicA, topicB (*), topicA, topicB (*)
-- ( '*' denotes a retained publication )
```

```

-- Get the first message on the subscriber queue, it will be on topicA.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicA

-- Get the second message on the subscriber queue, it will be on topicA.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicA

-- Request the broker to now send retained publications on topicB
select msgsend(NULL, @QM + ',queue=' + @BROKER
    option 'rfhCommand=requestUpdate'
    message header 'CorrelationAsId=yes'
        + ',correlationId=' + @CORRELID
        + ',topics=topicB'
        + ',streamName=' + @STREAM
        + ',queueName=' + @SUBQUEUE)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014805

-- Get the next message on the subscriber queue, it will be on topicB.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicB

-- Get the next message on the subscriber queue, it will be on topicB.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicB

-- Deregister the subscriber, for all topics.
select msgsend(NULL, @QM + ',queue=' + @BROKER
    option 'rfhCommand=deregisterSubscriber'
    message header 'CorrelationAsId=yes'
        + ',correlationId=' + @CORRELID
        + ',deregAll=yes'
        + ',streamName=' + @STREAM
        + ',queueName=' + @SUBQUEUE)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014807

```

ブローカ応答の例

この例は、要求/応答メッセージングを使用して、pub/sub ブローカからの応答を確認する方法を示します。サブスクリプションは user1 によって登録され、pub/sub ブローカ応答が確認されます。同じサブスクリプションが再度 user2 によって登録されますが、サブスクリプション名が異なるため、pub/sub ブローカからのエラー応答が発生します。

user1 によって実行されるクエリ:

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER      varchar(100)
-- @SUBQUEUE has the subscriber queue name
declare @SUBQUEUE    varchar(100)
-- @REPLY has the reply queue name
declare @REPLY       varchar(100)

-- Put broker, subscriber and reply queue names into variables
select @QM          = 'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER      = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @SUBQUEUE    = 'Q1.SUBSCRIBER'
select @REPLY       = 'Q1.REPLY'

-- Register the subscriber.
select msgsend(NULL, @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerSubscriber, msgType=request'
               message header 'correlationAsId=generate'
                               + ',topics=topicA'
                               + ',streamName=Q1.STREAM'
                               + ',queueName=Q1.SUBSCRIBER'
                               + ',replyToQueue=Q1.REPLY')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Read the response
select msgrecv(@QM + ',queue=' + @REPLY option 'timeout=30ss')
-----
NULL

-- Check @@msgproperties
select @@msgproperties
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<msgproperties
  MQPSReasonText="&apos;MQRC_NONE&apos;"
  MQPSReason="0"
  MQPSCompCode="0">
</msgproperties>

-- Check MQPSCompCode
if (msgpropvalue('MQPSCompCode', @@msgproperties) != "0")
begin
  print "registerSubscriber failed"
end
```

user2 によって実行されるクエリ:

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER      varchar(100)
-- @SUBQUEUE has the subscriber queue name
declare @SUBQUEUE    varchar(100)
-- @REPLY has the reply queue name
declare @REPLY       varchar(100)

-- Put broker, subscriber and reply queue names into variables
select @QM=          'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER=      'SYSTEM.BROKER.CONTROL.QUEUE'
select @SUBQUEUE=    'Q1.SUBSCRIBER'
select @REPLY=       'Q1.REPLY'

-- Register the subscriber
select msgsend(NULL, @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerSubscriber, msgType=request'
               message header 'correlationAsId=generate'
                           + ',topics=topicA'
                           + ',streamName=Q1.STREAM'
                           + ',queueName=Q1.SUBSCRIBER'
                           + ',replyToQueue=Q1.REPLY')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Read the response
select msgrecv(@QM + ',queue=' + @REPLY option 'timeout=30ss')
-----
NULL

-- Check @@msgproperties
select @@msgproperties
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<msgproperties
  MQPSUserId="'&apos;user2 &apos;";
  MQPSReasonText="'&apos;MQRCCF_DUPLICATE_IDENTITY&apos;";
  MQPSReason="3078"
  MQPSCompCode="2"
</msgproperties>

-- Check MQPSCompCode
if (msgpropvalue('MQPSCompCode', @@msgproperties) != "0")
begin
print "registerSubscriber failed"
end
```

MQ クラスタ・キュー・オブジェクトの使用

Active Messaging を使用すると、Adaptive Server をクライアントに使用して、WebSphere MQ クラスタ機能と通信できます。msgsend を使用して、キュー・マネージャに接続されている任意のクラスタのすべてのクラスタ・キューにメッセージを送信することができます。

注意 msgrecv 関数はリモート・キュー接続をサポートしていません。

クラスタでは、同じキューのインスタンスをホストしている複数のキュー・マネージャを持つことができます。たとえば、MASTER_MQ1 と SLAVE_MQ1 という 2 つのキュー・マネージャは、両方で CQ1 というクラスタ・キューをホストできます。両方のキュー・マネージャはクラスタの INV_CQ1 をジョインするため、CQ1 クラスタ・キューの 2 つのインスタンスが INV_CQ1 クラスタに入ることになります。

リモート・キュー・マネージャを指定するには、終了ポイント構文セグメントの remote_qmgr を使用します。複数のインスタンスを保持するクラスタ・キューにメッセージを送信するときに、送信先のインスタンスがどれでも構わない場合や、クラスタ・キュー・インスタンス間の負荷のバランスを取る必要がない場合は、この remote_qmgr オプションを無視できます。そのような場合、WebSphere MQ は自動的に負荷のバランスを取ります。

- 接続されているキュー・マネージャ上にインスタンスがある場合は、WebSphere MQ によって自動選択されます。
- 接続されているキュー・マネージャ上にインスタンスがない場合は、WebSphere MQ によって適したインスタンスが特定されます。

デフォルトのアルゴリズムを使用しない場合は、クラスタ負荷の終了を設定します。終了は、データベースのトリガに似た WebSphere MQ の機能です。終了とその設定方法の詳細については、IBM WebSphere MQ のマニュアルを参照してください。

同じキューを持つ複数のインスタンスでクラスタを使用すると、正しいキューのコピーをホストする任意のキュー・マネージャにメッセージを送信できます。ただし、順次整合性を維持しなければならない複数のメッセージを持っているユーザには、これは悪影響となります。たとえば顧客がベンダに次のメッセージを送信したとします。

- 1 「100 個のウィジェットを送ってください」と午前 9 時 00 分に送信しました。
- 2 「50 個のウィジェットを送ってください」と午前 9 時 30 分に送信しました。
- 3 「最初のお願いをキャンセルしてください」と午前 10 時に送信しました。

この例では、顧客が購入したいウィジェットの最終数量が 50 個 (100 + 50 - 100 = 50) であることをベンダが把握するには、メッセージの順序を維持する必要があります。メッセージ 2 がメッセージ 1 よりも先に到着してしまうと、ベンダは顧客が 100 個のウィジェットを購入したいと勘違いしてしまいます。

この問題には、`msgsend` 関数で `option_string` 型の `clustQBinding` を指定して、同じインスタンスにこれらのメッセージを配置することで対処できます。`clustQBinding` のオプションは `bind`、`nobind`、`default` です。これらのオプションの詳細および例については、「[msgsend](#)」(99 ページ) の参照ページを参照してください。

リモート・キュー・オブジェクトの使用

次の状況で、`msgsend remote_qmgr` オプションを使用してリモート・キュー・マネージャの名前を指定し、リモート・キュー・オブジェクトにメッセージを送信することができます。

- ローカル・キュー・マネージャとリモート・キュー・マネージャが1つのクラスタ内にあり、ローカル・キュー・マネージャにリモート・キュー・マネージャのクラスタ・キュー・マネージャ定義が保存されている場合。
- ローカル・キュー・マネージャに転送キューがあり、転送キューの名前がリモート・キュー・マネージャの転送キューと同じ場合。
- ローカル・キュー・マネージャにキュー・マネージャ・エイリアスがあり、キュー・マネージャ・エイリアスの名前がリモート・キュー・マネージャのキュー・マネージャ・エイリアスと同じ場合。

注意 Adaptive Server はリモート・キュー・マネージャをターゲット・キュー・マネージャに設定し、キューをターゲット・キューに設定します。Adaptive Server が関連した転送キューにメッセージを送信するとすぐに、メッセージがターゲット・キューに送信されていなくても、Adaptive Server によって正常ステータスが返されます。

WebSphere MQ によるキュー・マネージャ間のメッセージの送信方法の詳細については、IBM のマニュアルを参照してください。

転送キューにメッセージが入れられると、ローカル・キュー・マネージャはサブネット内にリモート・キュー・マネージャ定義がないかを探します。ローカル・キュー・マネージャには、次の種類があります。

- クラスタの完全レポジトリ – ローカル・キュー・マネージャにはリモート・クラスタ・キュー・マネージャの定義が含まれています。
- 部分レポジトリ – ローカル・キュー・マネージャはリモート定義の場所を把握していないことがあり、その場合は WebSphere MQ でエラーが返されます。ただしエラーが発生しても、ローカル・キュー・マネージャはリモート・キュー・マネージャの場所をたずねません。

ローカル・キュー・マネージャでリモート・キュー・マネージャ定義が見つかったと、ローカル・キュー・マネージャはクラスタ転送キューを使用してリモート・キュー・マネージャにメッセージを送信します。送信されると、リモート・キュー・マネージャはメッセージをターゲット・キューに送信します。このため、ターゲット・キューがクラスタ・キューでなくても操作は成功します。

その他の状況では、関連した転送キューのチャンネルがメッセージを受信し、チャンネルの接続先のキュー・マネージャに送信します。そのようなチャンネルが存在しない場合やチャンネルが開始されていない場合、チャンネルが開始するまでメッセージは転送キューに保存されます。

テキスト・メッセージングの使用

JMS および WebSphere MQ は、いずれもバイト・メッセージとテキスト・メッセージを処理できます。

テキスト・メッセージと JMS

JMS でメッセージを送受信すると、Active Messaging はメッセージ・ペイロードのデータ型を自動検出し、バイト・メッセージまたはテキスト・メッセージのいずれかで適切に処理します。メッセージを送信する際は、JMS は `char`、`varchar`、`unicar`、`univarchar`、`text`、`unitext` を有効なテキスト・メッセージ・タイプとして認識します。

例 1 テキスト・メッセージを JMS メッセージング・バスに送信します。

```
declare @msg varchar(1024)
select @msg = 'abcd'
select msgsend(@msg,
  'tibco_jms:tcp://my_jms:7222?queue=sample,user=admin')
```

例 2 テキスト・メッセージを JMS メッセージング・バスから受信します。

```
select msgrecv('tibco_jms:tcp://my_jms:7222?
  queue=sample,user=admin', returns varchar(1024))
```

例 3 バイト・メッセージを JMS メッセージング・バスに送信します。

```
declare @msg varbinary(1024)
select @msg = 'abcd'
select msgsend(@msg,
  'tibco_jms:tcp://my_jms:7222?queue=sample,user=admin')
```

例 4 バイト・メッセージを JMS メッセージング・バスから受信します。

```
select msgrecv('tibco_jms:tcp://my_jms:7222?
  queue=sample,user=admin', returns varbinary(1024))
```


テキスト・メッセージと MQ

WebSphere MQ でメッセージを受信した場合、“formatName” メッセージ・プロパティが“MQSTR”に設定されている場合にのみ MQ はメッセージをテキスト・メッセージと見なします。それ以外の場合、MQ はメッセージをバイト・メッセージとして処理します。

例 1 テキスト・メッセージを WebSphere MQ に送信します。

```
declare @msg varchar(1024)
select @msg = 'abc'
select msgsend(@msg, 'ibm_mq:channel1/TCP/host1(7654)?
qmgr=QM,queue=Q1,alter_user=yes',message property "formatName=MQSTR")
```

例 2 テキスト・メッセージを WebSphere MQ から受信します。

```
select msgrecv('ibm_mq:channel1/TCP/host1(7654)?
qmgr=QM,queue=Q1,alter_user=yes',
option 'bufferLength=20000k,timeout=60000',
returns varchar(1024))
```

例 3 バイト・メッセージを WebSphere MQ に送信します。

```
declare @msg varbinary(1024)
select @msg = 'abc'
select msgsend(@msg, 'ibm_mq:channel1/TCP/host1(7654)?
qmgr=QM,queue=Q1,alter_user=yes')
```

例 4 バイト・メッセージを WebSphere MQ から受信します。

```
select msgrecv('ibm_mq:channel1/TCP/host1(7654)?
qmgr=QM,queue=Q1,alter_user=yes',
option 'bufferLength=20000k,timeout=60000',
returns varbinary(1024))
```

例 5 WebSphere MQ では、ペイロードが UTF8 でエンコードされていれば、バイト・ペイロードをテキスト・メッセージとして送信できます。この例では、テキスト・メッセージ“abc”はバイト・ペイロード 0x616263 に基づいています。これはテキスト“abc”の UTF8 エンコードが 0x616263 になるからです。

```
declare @msg varbinary(1024)
select @msg = 0x616263
select msgsend(@msg, 'ibm_mq:channel1/TCP/host1(7654)?
qmgr=QM,queue=Q1,alter_user=yes',
message property "formatName=MQSTR")
```

Adaptive Server Cluster Edition のサポート

Active Messaging は、Adaptive Server Cluster Edition の次のクライアント・テクノロジーをサポートしています。

- 「ログインのリダイレクト」(30 ページ) – ログインを確認する前に、受信クライアント接続を別のインスタンスにリダイレクトできるインスタンス機能。ログイン・リダイレクトはログイン手順中に行われます。クライアント・アプリケーションは、リダイレクトされた通知を受け取りません。
- 「拡張高可用性」(31 ページ) – 拡張フェールオーバー設定で、Adaptive Server は接続時に「高可用性対応」クライアントに対してフェールオーバー・アドレスのリストを提供します。これにより、接続先のインスタンスが失敗したときに、高可用性対応クライアントまたはアプリケーションが複数回フェールオーバーを行うことができます。

これらのクライアントでは、`interfaces` ファイルまたはディレクトリ・サービスに `HAFILOVER` エントリは必要ありません。ただし、`interfaces` ファイルまたはディレクトリ・サービスに `HAFILOVER` エントリがある場合、クライアントは Adaptive Server がフェールオーバー・アドレス、または接続先サーバのリストを送信するまでこのエントリを使用し続けます。クライアントは常に、Adaptive Server が提供する最新のリストを使用します。

注意 Active Messaging 機能は、既存のクライアントが1つのクラスタ・インスタンスから別のインスタンスに送信されたときに発生する接続マイグレーションには対応していません。

ログインのリダイレクト

ログインのリダイレクトは、Adaptive Server Workload Manager が論理クラスタ設定およびクラスタの現在の負荷に基づいて、受信接続を特定のインスタンスに送信する場合に使用されます。

ログインのリダイレクトは、ログイン時にロード時の考慮事項の関係で、クライアントが別のインスタンスにログインするように指示された場合に発生します。

クライアントのリダイレクトのために追加の設定を実行する必要はありません。リダイレクトは自動的に行われます。

次の例では、“mycluster” というクラスタで実行されているノード “blade1” と “blade2” に “ase1” と “ase2” というインスタンスがあります。

```
ase1
  query tcp ether blade1 19786
ase2
  query tcp ether blade2 19786
```

```
mycluster
  query tcp ether bladel 19786
  query tcp ether blade2 19786
```

たとえば Active Messaging が“ase1”と“ase2”で有効になっているときにアプリケーション・サーバが“ase1”に接続を試みたところ“ase1”が使用不可能だった場合、このログインは“ase2”インスタンスにリダイレクトしてメッセージング操作を実行します。

『Cluster ユーザーズ・ガイド』を参照してください。

拡張高可用性

Adaptive Server は接続時に「HA 対応」クライアントに対してフェールオーバー・アドレスのリストを提供します。これにより、接続先のインスタンスが使用不可能になるたびに、高可用対応クライアントまたはアプリケーションで複数回フェールオーバーを行うことができます。フェールオーバー・リストがクライアントに送信されていない場合、クライアントは interfaces ファイルの HAFAILOVER エントリ情報を使用します。

次の例では、ログイン時にネットワーク障害が発生した場合、インスタンスが拡張高可用性リストを送信する前に、HA 対応クライアントがフェールオーバーを行うことができます。

```
ase1
  query tcp ether bladel 19786

ase2
  query tcp ether blade2 19786

mycluster
  query tcp ether bladel 19786
  query tcp ether blade2 19786
  hafailover mycluster
```

クライアント・アプリケーションではクラスタのインスタンスへの接続を確立できるまでクエリの各行を試みるため、HAFAILOVER エントリはクラスタ・エイリアスをサーバ名に使用します。クラスタ環境で拡張高可用性を有効にするには、『Cluster ユーザーズ・ガイド』を参照してください。

拡張フェールオーバーには Open Client 15.0 ESD #3 以降が必要です。Cluster Edition のクライアント・ライブラリには ESD #8 が含まれています。

Open Client は拡張フェールオーバーに CS_PROP_EXTENDEDFAILOVER プロパティを使用します。『Client-Library/C リファレンス・マニュアル』を参照してください。

スレッド・カーネルに対する Active Messaging のサポート

スレッド・カーネルで Active Messaging を使用している場合は、次のパラメータを設定しないでください。

- max online Q engines
- number of Q engines at startup
- max online engines
- number of engines at startup

少なくとも1つのスレッドを使用するには `syb_blocking_pool` を設定する必要があります。さらに、パフォーマンス上の理由によって、少なくとも Active Messaging を実行している Adaptive Server セッションと同等にする必要があります。たとえば、Active Messaging を設定する前の `syb_blocking_pool` のスレッド数が2である場合、Active Messaging を設定すると、3つの追加 Adaptive Server セッションは Active Messaging を使用します。Sybase® では、`syb_blocking_pool` のスレッド数を $2+3=5$ になる5に設定することをおすすめします。

注意 この設定は、15.7 より前のバージョンの Adaptive Server と互換性があります。

国際化のサポート

Adaptive Server バージョン 15.0.2 ESD #1 以降では、次のような場合に Adaptive Server とメッセージング・バス間のメッセージの送受信の国際化をサポートしています。

- 送信者のサーバ文字セットが GB18030 (簡体字中国語) を使用するように設定されている場合 - 送信者は中国語のメッセージをメッセージング・バスに送信できます。
- 受信者のサーバ文字セットが Big5 (繁体字中国語) を使用するように設定されている場合 - 受信者は中国語のメッセージをメッセージング・バスから受信できます。

例 **例 1** 現在の文字セットを設定し、1 台の Adaptive Server のメッセージング・バスに GB18030 文字セットの中国語を送信します。

```
1> sp_configure "default character set id"
2> go
```

Parameter Name	Default	Memory Used	Config Value	Run Value	Unit	Type
default character set id	1	0	173	173	id	static

```
(1 row affected)
(return status = 0)
```

```
1> declare @msg varchar(1024)
2> select @msg = 0xd6d0cec4
3> select msgsend(@msg, 'ibm_mq:channel1/TCP/host1(7654)?
    qmgr=QM,queue=Q1,alter_user=yes',message property "FormatName=MQSTR")
```

例 2 Big5 文字セットを実行中の別の Adaptive Server のメッセージング・バスから中国語のメッセージを受信します。

```
1> sp_configure "default character set id"
2> go
```

Parameter Name	Default	Memory Used	Config Value	Run Value	Unit	Type
default character set id	1	0	161	161	id	static

```
(return status = 0)
```

```
1> declare @msg varchar(1024)
2> select @msg = msgrecv('ibm_mq:channel1/TCP/host1(7654)?
    qmgr=QM,queue=Q1,alter_user=yes',
3> option 'bufferLength=100k,timeout=60000',
4> returns varchar(16384))
5> select convert(varbinary(1024), @msg)
6> go
```

```
-----
0xa4a4a4e5
```

出力“0xa4a4a4e5”は Big5 文字セットで中国語の「CHINESE」をバイナリ表現したものです。

トランザクション・メッセージの動作

デフォルトでは、すべてのメッセージング操作 (msgsend, msgrecv, msgpublish, msgconsume, msgsubscribe, msgunsubscribe) は、データベース・トランザクションがロールバックすると、ロールバックします。ただし、msgsend または msgrecv を使用して失敗したメッセージング操作は、親データベース・トランザクションには影響しません。

- トランザクションに含まれるプロセスで msgsend または msgpublish が実行されると、プロセスによってトランザクションがコミットされるまで、結果のメッセージはメッセージ・バスに表示されません。これは SQL update や insert を実行するのとは異なります。SQL update および insert コマンドをトランザクションで実行するプロセスでは、コミット前にコマンドの効果をすぐに確認できます。

- トランザクションで `msgsend` または `msgpublish` を実行してメッセージを送信するプロセスでは、トランザクションがコミットされるまで、`msgrecv` または `msgconsume` を使用してメッセージを読み込むことはできません。

トランザクション・メッセージングの設定オプション

トランザクション動作は `set transactional messaging` コマンドによって制御されます。このコマンドでは3つの操作モードが提供されるため、トランザクションでメッセージング機能を使用するときに好きな動作を選択できます。

```
set transactional messaging [ none | simple | full ]
```

- none** – メッセージング操作とデータベース操作は相互に影響を及ぼしません。この例では、`insert` の成否に関係なく、`msgsend` が実行されてメッセージはメッセージ・バスに送信されます。

```
begin tran
  msgsend (...)
  insert (...)
rollback
```

- simple** (デフォルト設定) – メッセージング操作に影響を与えるデータベース操作を実行しますが、メッセージング操作はデータベース・トランザクションに影響を与えません。この例では、`msgsend` が失敗しても `insert` はアポートしません。

```
begin tran
  insert (...)
  msgsend (...)
commit
```

この例では、`msgsend` はロールバックされます。

```
begin tran
  insert (...)
  msgsend (...)
rollback
```

- full** – すべてのトランザクション動作が行われます。このモードでは、メッセージング操作とデータベース操作は相互に影響を及ぼします。メッセージング操作が失敗すると、トランザクションはロールバックします。データベース・トランザクションが失敗すると、メッセージング操作はロールバックします。

```
begin tran
  select @message=msgrecv(Q1,...)
  insert t2 values (@message,...)
  select msgsend ( t2.status,...)
commit tran
```

- トランザクション・メッセージングが **full** または **simple** に設定されている場合、メッセージを送信またはパブリッシュするコミットされていないトランザクションを同じトランザクション内に読み込むことはできません。

Transact-SQL アプリケーションでは、アプリケーションの条件に応じて、優先モードを指定できます。

注意 トランザクションの内部で **set transactional messaging** を使用することはできません。

MQ のセキュリティ

この項では、セキュリティと MQ について説明します。

MQ キュー・マネージャへの接続

MQ 終了ポイントでは、JMS を使用するときのようにユーザ名とパスワードを指定することはできません。MQ キュー・マネージャへのすべての接続は、Adaptive Server プロセスが実行されているユーザとして行われます。MQ キュー・マネージャへの接続後、Adaptive Server は操作を実行している Adaptive Server ログインとしてキューを開こうとします。このため、ユーザは次のことを行う必要があります。

- MQ キュー・マネージャが実行されているマシンにユーザ・アカウントを持つ必要があります。そのようなアカウントがない場合、ユーザは **msgsend** 関数の **alter_user=yes** オプションを使用してメッセージング操作を実行する必要があります。msgsend リファレンス・ページの表 3-11 (108 ページ) を参照してください。
- 表 2-2 (37 ページ) に記載の MQ 権限を持っている必要があります。

注意 Adaptive Server の **messaging_role** では引き続き Real Time Data Services 組み込み関数を実行する必要があります。

さらに、**sp_msgadmin** の **'register, login'** および **'default, login'** 関数では、キュー・マネージャで終了ポイントが指定されている場合、Adaptive Server ログインを登録したり、デフォルトの Adaptive Server ログインを作成したりすることはできません。「**sp_msgadmin**」(59 ページ) を参照してください。

Adaptive Server ホスト・マシンへの MQ クライアントのインストール

Adaptive Server ホスト・マシンには MQ クライアント・ソフトウェアをインストールする必要があります。

Adaptive Server は MQ クライアント共有ライブラリを動的にロードします。共有ライブラリのインストール場所を表 2-1 に示します。

表 2-1: MQ クライアント共有ライブラリとディレクトリ

プラットフォーム	ディレクトリ	ライブラリ名
Solaris 32	<i>/opt/mqm/lib</i>	<i>libmqmcs.so, libmqic.so</i>
Solaris 64	<i>/opt/mqm/lib64</i>	<i>libmqmcs.so, libmqic.so</i>
Solaris AMD64	<i>/opt/mqm/lib64</i>	<i>libmqmcs.so, libmqic.so</i>
Linux 32	<i>/opt/mqm/lib</i>	<i>libmqic_r.so</i>
Linux AMD64	<i>/opt/mqm/lib64</i>	<i>libmqic_r.so</i>
HPPA 64	<i>/opt/mqm/lib64</i>	<i>libmqic.sl</i>
HPIA 64	<i>/opt/mqm/lib64</i>	<i>libmqic.so</i>
AIX 64	<i>/usr/mqm/lib64</i>	<i>libmqic_r.a(mqic_r.o)</i>
Windows 32	<i>c:\Program Files\IBM\Websphere MQ\bin</i>	<i>MQIC32.DLL</i>

- HP、HPIA、Linux、Linux AMD、Solaris、Solaris AMD – Adaptive Server はライブラリを */opt/mqm/lib* からロードするため、MQ のために `LD_LIBRARY_PATH` を設定する必要はありません。
- IBM – Adaptive Server を起動する前に、`$LIBPATH` に */usr/mqm/lib64* を含めるように設定してください。
- Windows – Adaptive Server を起動する前に、`%PATH%` にライブラリを含めるように設定してください。

MQ の権限

MQ の設定には、次の UNIX ユーザ・アカウントとユーザ・グループ (原則/グループ) 権限が必要です。

表 2-2: MQ の原則/グループおよびその権限

MQ の原則 / グループ	MQ の権限
データ・サーバ実行プログラムを実行する OS ログイン	キュー・マネージャに connect, altusr, inq, setid
任意のメッセージング操作を実行している代替ユーザの OS ログイン	キューに inq
任意のメッセージングの read 操作を実行している代替ユーザの OS ログイン	キューに get
任意のメッセージングの browse 操作を実行している代替ユーザの OS ログイン	キューに browse
任意のメッセージングの send 操作を実行している代替ユーザの OS ログイン	キューに put
replyToQueue として指定された代替ユーザの動的キューの OS ログイン	キュー・マネージャに crt, dlt, モデル・キューに get, inq

注意 メッセージをリモート・キューに送信すると、WebSphere MQ は転送キュー上のユーザ権限を確認します。

msgsend で alter_user=yes を指定すると、代替ユーザは Adaptive Server を実行しているオペレーティング・システムのログインになります。alter_user を指定しないと、代替ユーザは MQ 操作を実行している Adaptive Server ログインになります。

MQ 情報のクエリ

Adaptive Server バージョン 15.0.2 ESD #1 以降を実行している場合、sp_msgadmin スタアド・プロシージャの show オプションを使用して、指定のキュー・マネージャの MQ オブジェクトについて、Adaptive Server に次の情報のクエリを行うことができます。

- キュー・マネージャの名前
- キュー・マネージャに属するすべてのキューとクエリ・タイプ
- キュー・マネージャに属するすべてのチャンネルとそのタイプ

❖ ***sp_msgadmin 'show'* を使用するための WebSphere MQ の準備**

- 1 WebSphere MQ で、照会を行うキュー・マネージャを開始します。
- 2 キュー・マネージャの MQ リスナが実行中であることを確認します。
- 3 キュー・マネージャのコマンド・サーバを開始します。
- 4 キュー・マネージャに SYSTEM.ADMIN.COMMAND.QUEUE というキューがあることを確認します。

これらの手順の実行方法の詳細は、IBM Publication Center (<http://www.elink.ibm.com/publications/servlet/pbi.wss>) から提供されている WebSphere MQ のマニュアルを参照してください。

この章では、Active Messaging を管理する際に使用できるグローバル変数、ストアド・プロシージャ、関数、および構文セグメントについて説明します。

トピック名	ページ
メッセージ関連のグローバル変数	40
<msgheader> と <msgproperties> ドキュメント	47
Adaptive Server 固有のメッセージ・プロパティ	48
キーワード	49
ストアド・プロシージャ	50
組み込み関数	50
構文セグメント	51
sp_configure 'enable real time messaging'	51
sp_engine	55
sp_msgadmin	59
msgconsume	71
msgpropcount	74
msgproplist	75
msgpropname	77
msgproptype	78
msgpropvalue	80
msgpublish	81
msgrecv	85
msgsend	99
msgsubscribe	133
msgunsubscribe	135
endpoint	138
option_string	141
sizespec	142
timespec	143

メッセージ関連のグローバル変数

これらのグローバル変数は、送受信された最近のメッセージからのメッセージ情報に対するアクセス権を持つアプリケーション・プログラムを提供します。

@@msgcorrelation

送信されたか、読み取られた最後のメッセージの相関が含まれます。

- (MQ) MQ は、`@@msgcorrelation` が印刷可能な文字で構成されているかどうかを確認しません。現在のサーバの文字セットでの `@@msgcorrelation` にアプリケーション・プログラムを依存させないでください。アプリケーション・プログラムは、後続のメッセージのセレクタとしてのみ `@@msgcorrelation` を使用する必要があります。`@@msgcorrelation` がアプリケーションに返された場合は、これを varbinary データ型に変換します。
- (JMS) `@@msgcorrelation` には、送受信された最新のメッセージの correlation ID が含まれます。

@@msgheader

受信された最新のメッセージのメッセージ・ヘッダ情報が含まれます。この変数のフォーマットは XML です。このフォーマットの詳細については、「`<msgheader>` と `<msgproperties>` ドキュメント」(47 ページ)を参照してください。

`@@msgheader` を設定する関数には、`msgrecv` と `msgconsume` があります。

表 3-1 は、MQ における `@@msgheader` グローバル変数の有効なフィールド名とその説明を示します。表 3-2 (42 ページ) は、JMS における `@@msgheader` のフィールドと説明を示します。

表 3-1: MQ @@msgheader フィールドと説明

プロパティ名	説明
ApplIdentityData	ID に関連するアプリケーション・データ。
ApplOriginData	オリジンに関連するアプリケーション・データ。
CodedCharSetId	数値コード化された文字セットの識別子。
CorrelId	相関識別子。
Encoding	メッセージ内のバイナリ・データのコード化。Encoding フィールドのフラグのビットマスク。
DecimalEncoding	メッセージ・ペイロードの小数点数のコード化であり、Encoding フィールドから抽出された合成プロパティ。指定する値は次のとおりです。 <ul style="list-style-type: none"> • BigEndian – 小数点数はビッグ・エンディアン。 • LittleEndian – 小数点数はリトル・エンディアン。 • Undefined – 小数点数はビッグ・エンディアンまたはリトル・エンディアンとして定義されていない。
Feedback	フィードバック・ステータス。

プロパティ名	説明
FloatEncoding	<p>バイロードの浮動小数点数のコード化であり、Encoding フィールドから抽出された合成プロパティ。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • BigEndian – 浮動小数点数はビッグ・エンディアン。 • LittleEndian – 浮動小数点数はリトル・エンディアン。 • Undefined – 浮動小数点数はビッグ・エンディアンまたはリトル・エンディアンとして定義されていない。
Format	<p>メッセージ・データのフォーマット名。MQ 定義のフォーマット名またはアプリケーション定義のフォーマット名の場合もあります。</p>
GroupId	<p>グループ識別子。</p>
IntegerEncoding	<p>バイロードの整数のコード化であり、Encoding フィールドから抽出された合成プロパティ。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • BigEndian – 整数はビッグ・エンディアン。 • LittleEndian – 整数はリトル・エンディアン。 • Undefined – 整数のエディアンは定義されていない。
LastMsgInGroup	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • true – メッセージはグループの最後のメッセージ。 • false – メッセージはグループの最後のメッセージではない。
MsgId	<p>メッセージ識別子。</p>
MsgInGroup	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • true – メッセージはグループの一部。 • false – メッセージはグループの一部ではない。
MsgSeqNumber	<p>メッセージのシーケンス番号。</p>
MessageType	<p>次の場合を除き、小数点数の形式のメッセージ・タイプ。</p> <ul style="list-style-type: none"> • request – メッセージは要求メッセージ。 • reply – メッセージは応答メッセージ。 • datagram – メッセージはデータグラム・メッセージ。 • report – メッセージは報告メッセージ。
NegativeActionNotification	<p>Report フィールドから抽出された合成プロパティ。受信アプリケーションは、負のアクション通知 (NAN) レポートを生成します。</p> <ul style="list-style-type: none"> • yes – 受信アプリケーションは、NAN レポート・メッセージを生成し、ReplyToQ フィールドと ReplyToQMgr フィールドで指定された送信先にこれを送信する。 • no – 受信アプリケーションは、NAN レポート・メッセージを生成しない。
Persistence	<p>メッセージの継続性。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • persistent – メッセージは、継続的なメッセージ。 • non-persistent – メッセージは、非継続的なメッセージ。
PositiveActionNotification	<p>Report フィールドから抽出された合成プロパティ。受信アプリケーションは、正のアクション通知 (PAN) レポートを生成します。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – 受信アプリケーションは、PAN レポート・メッセージを生成し、ReplyToQ フィールドと ReplyToQMgr フィールドで指定された送信先にこれを送信する。 • no – 受信アプリケーションは、PAN レポート・メッセージを生成しない。

プロパティ名	説明
PutAppName	メッセージをキューに入れるアプリケーションの名前。
PutAppType	メッセージをキューに入れるアプリケーションのタイプ。
PutDate	メッセージがキューに入れられた日付。
PutTime	メッセージがキューに入れられた時刻。
ReplyCorrelationId	<p>Report フィールドから抽出された合成プロパティ。レポート・メッセージの相関 ID として何をを使うかを示します。</p> <ul style="list-style-type: none"> • msgId – レポート・メッセージの相関 ID は、受信メッセージのメッセージ ID に設定される。 • correlationId – レポート・メッセージの相関 ID は、受信メッセージの相関 ID に設定される。
ReplyMsgId	<p>Report フィールドから抽出された合成プロパティ。レポート・メッセージのメッセージ ID として何をを使うかを示します。</p> <ul style="list-style-type: none"> • new – レポート・メッセージのメッセージ ID として新しいメッセージ ID を使用する。 • original – レポート・メッセージのメッセージ ID として受信したメッセージ ID を使用する。
ReplyToQ	応答キューの名前。
ReplyToQMgr	応答キュー・マネージャの名前。
Report	メッセージのレポート・オプション。これは、MQRO * フラグのビットマップです。
UserIdentifier	ユーザ識別子。

表 3-2: JMS @msgheader フィールドと説明

プロパティ名	説明
correlation	メッセージからの相関 ID。
destination	メッセージの送信先の名前。
encoding	メッセージのコード化の名前。
messageid	メッセージからのメッセージ ID。
mode	<p>メッセージの配信モード。次のいずれかになります。</p> <ul style="list-style-type: none"> • persistent • non-persistent
priority	メッセージの優先度。
redelivered	メッセージの再配信ステータス。
replyto	メッセージの応答名。
timestamp	メッセージのタイムスタンプ。
ttl	メッセージが存在する期間を示すメッセージの有効期間の値。
type	メッセージのタイプ。

@@msgid

送受信された最新のメッセージ ID が含まれます。

MQ は、**@@msgid** が印刷可能な文字で構成されているかどうかを確認しません。現在のサーバの文字セットでの **@@msgid** にアプリケーション・プログラムを依存させないでください。アプリケーション・プログラムは、後続のメッセージのセレクタとしてのみ **@@msgid** を使用する必要があります。**@@msgid** がアプリケーションに返される場合は、これを **varbinary** データ型に変換する必要があります。

変数を設定する関数は、次のとおりです。

- (JMS) `msgsend`、`msgpublish`、`msgrecv`、`msgconsume`
- (MQ) `msgsend`、`msgrecv`

@@msgproperties

受信された最新のメッセージのメッセージ・プロパティ情報が含まれます。この変数のフォーマットは XML です。このフォーマットの詳細については、「[<msgheader> と <msgproperties> ドキュメント](#)」(47 ページ)を参照してください。

- (JMS) **@@msgproperties** は、メッセージのユーザ・プロパティです。
- (MQ) 次のようになります。
 - メッセージに 1 つまたは複数の MQRF ヘッダが含まれる場合、MQRF ヘッダの名前と値のペアが **@@msgproperties** に挿入されます。
 - MQRF ヘッダの名前と値のペアはユニークではない名前を持つ可能性があるため、“_ddd”を追加して名前をユニークにします。ここで、ddd はユニークにするための整数の拡張子です。たとえば、MQRF ヘッダに次のようなトピックがあるとします。

```
MQPSTopic    */baseball
MQPSTopic    */baseball/world series
MQPSTopic    */sports
```

@@msgproperties におけるこれらのプロパティは、次のようになります。

```
MQPSTopic    */baseball
MQPSTopic_1  */baseball/world series
MQPSTopic_2  */sports
```

@@msgproperties を設定する関数は、次のとおりです。

- (JMS) `msgrecv`、`msgconsume`
- (MQ) `msgrecv`

RF ヘッダに値のペアが存在する場合に、RF ヘッダから抽出される値のペアを次に示します。

MQPSCCommand	MQPSIntData	MQPSReason	MQPSSubIdentity
MQPSCCompCode	MQPSParmId	MQPSReasonText	MQPSSubName
MQPSCorrelId	MQPSPubOpts	MQPSRegOpts	MQPSSubUserData
MQPSDelOpts	MQPSPubTime	MQPSSeqNum	MQPSSubUserData
MQPSErrorId	MQPSQMgrName	MQPSStreamName	MQPSTopic
MQPSErrorPos	MQPSQName	MQPSStringData	MQPSUserId

認識されない名前は無視されます。RF ヘッダの値が引用符 (") で囲まれている場合、囲んでいる引用符は削除されます。引用符で囲まれた値の中にエスケープされた引用符 (") が含まれる場合、二重引用符は一重引用符に置き換えられます。

`@@msgreplyqmgr`

MQ のみ - 読み取られた最後のメッセージの ReplyToQmgr 名が含まれます。

`@@msgreplytoinfo`

直接送受信されたメッセージで使用されたトピックまたはキューの名前 (`provider_url`、`queue_name`、`topic_name`、`user_name`) が含まれます。永続的な送信先にも一時的な送信先にも設定できます。

`@@msgreplytoinfo` を設定する関数は、次のとおりです。

- (JMS) `msgconsume`、`msgpublish`、`msgrecv`、`msgsend`
- (MQ) `msgrecv`、`msgsend`

JMS のみ - パスワードは、`@@msgreplytoinfo` の値に含まれません。後続の `msgsend` または `msgrecv` 呼び出しで引数としてこの送信先を使用するには、`password=<your password>` を追加します。

MQ のみ - `remote_qmgr` の構文を含むことができます。`@@msgreplytoinfo` は、`@@msgcorrelation` を使用してクラスター・キュー・マネージャに対するサポートを表示する要求/応答メッセージングを示します。

たとえば、Adaptive Server が MASTER_MSCAI キュー・マネージャに接続し、MASTERQ として指定された `replyToQueue` プロパティを使用して SLAVE_MSCAI リモート・キュー・マネージャ上に配置された Q1 にメッセージを送信するとします。`msgsend` を送信すると、その値は `@@msgreplytoinfo` の値になります。

```
select msgsend('d','ibm_mq:CH1/tcp/host1(1105)?
qmgr=MASTER,remote_qmgr=SLAVE,queue=Q1,alter_user=yes',
    message property 'replyToQueue=MASTERQ')
go
select @@msgreplytoinfo
go
```

```
IBM_MQ:CH1/tcp/host1(1105)?qmgr=MASTER,queue=MASTERQ
```


もう1つの Adaptive Server は、キュー・マネージャ SLAVE に接続し、前に送信されたメッセージを Q1 から受け取ります。@@msgreplytoinfo グローバル変数は、remote_qmgr の構文を含むため、この例の場合の応答キューはリモート・キューとなります。

```
select msgrecv('ibm_mq:CH2/tcp/host2(4810)?
qmgr=SLAVE,queue=Q1,alter_user=yes', option 'timeout=100')
go
select @@msgreplytoinfo
go

ibm_mq:CH2/tcp/host2(4810)?qmgr=SLAVE,remote_qmgr=MASTER,queue=MASTERQ
```

注意 構文 remote_qmgr を含む @@msgreplytoinfo を使用して応答メッセージ msgrecv を送信する場合、応答メッセージが正しいリモート・キュー・マネージャに到達するかどうかは WebSphere MQ を設定した方法によって決定されます。「リモート・キュー・オブジェクトの使用」(27 ページ) を参照してください。

@@msgschema

JMS のみ – メッセージのスキーマまたは null 値が含まれます。Adaptive Server プロパティ `ase_message_body_schema` の値が含まれます。msgsend および msgpublish のスキーマ・オプションの説明を参照してください。

@@msgschema を設定する関数には、次のものがあります。msgsend、msgpublish。

@@msgstatus

最後の操作で例外が発生しなかった場合、サービス・プロバイダ例外の整数エラー・コードまたはゼロが含まれます。

@@msgstatus を設定する関数には、次のものがあります。msgsend、msgpublish、msgrecv、msgconsume。

@@msgstatusinfo

最後の msgsend、msgpublish、msgrecv、または msgconsume で例外または空の文字列が発生した場合、サービス・プロバイダ例外のエラー・メッセージまたはゼロが含まれます。

(MQ) 最後のメッセージング操作のプロバイダ・エラー・メッセージが含まれます。MQ クライアント・ライブラリは、ローカライズされたエラー・メッセージを提供しないため、次のようなエラー・メッセージが表示されます。

```
MQ API call failed with reason code '%s' (%d)
```

“%s” は、MQ 原因コードの MQ ニーモニックに置き換えられます。

“%d” は、10 進数の MQ 原因コードに置き換えられます。

変数を設定する関数は、次のとおりです。

- (JMS) msgsend、msgpublish、msgrecv、msgconsume。
- (MQ) msgsend、msgrecv。

`@@msgtimestamp` 最後に送信されたメッセージに含まれる `timestamp` が含まれます。
 変数を設定する関数は、次のとおりです。 `msgsend`、`msgpublish`。

例 **例 1** MQ のみ - `@@msgreplytoinfo` と `@@msgcorrelation` を使用して要求/応答メッセージを表示します。

セッション 1 (要求側)	セッション 2 (受信側)
<pre>select msgsend('sender mmessage', 'ibm_mq:channel1/TCP/host1(5678)' + '?qmgr=QM1' + ',queue=Q100', option 'msgType=request', message property 'correlationId=0x123456' + 'replyToQueue=Q200')</pre>	<pre>select msgrecv('ibm_mq:channel1/TCP/host1(5678)' + '?qmgr=QM1' + ',queue=Q100') select msgsend('receiver reply', @@msgreplytoinfo, option 'msgType=reply' message property 'correlationId=' + @@msgcorrelation)</pre>
<pre>select msgrecv('ibm_mq:channel1/TCP/host1(5678)' + '?qmgr=QM1' + ',queue=Q200' option 'timeout=30ss', + 'correlationID=0x123456')</pre>	

次に例を示します。

- 1 セッション 1 は、要求メッセージを Q100 に送信し、Q200 で応答メッセージを予期します。また、相関を 0x123456 に設定します。
- 2 セッション 2 は、Q100 からメッセージを読み取り、Q200 に応答メッセージを送信し、相関を 0x123456 に指定します。応答キューは、読み取ったばかりのメッセージから取得されます。
- 3 セッション 1 は、Q200 から応答メッセージを読み取り、相関 0x123456 を持つメッセージのみを要求します。

使用方法

- これらのグローバル変数は、長さ 16384 の `char` データ型です。
- `rtrim` を使用して後続ブランクを削除できます。

<msgheader> と <msgproperties> ドキュメント

説明 グローバル変数 `@@msgheader` と `@@msgproperties` は、返されたメッセージのヘッダとプロパティを含む XML `<msgheader>` と `<msgproperties>` ドキュメントを使用して設定されます。この項では、これらのドキュメントのフォーマットを指定します。

`PROPERTY_1`、`PROPERTY_2` などの名前のプロパティの `<msgheader>` と `<msgproperties>` ドキュメントの一般的なフォーマットは、次の構文の項の DTD テンプレートによって記述される形式になります。

構文

```
<!DOCTYPE msgheader [
<!ELEMENT msgheader EMPTY>
<!ATTLIST property_1 CDATA>
<!ATTLIST property_2 CDATA>
etc.
<!DOCTYPE msgproperties [
<!ELEMENT msgproperties EMPTY>
<!ATTLIST property_1 CDATA>
<!ATTLIST property_2 CDATA>
```

例

次の例は、2つの `select` 文の `<msgheader>` または `<msgproperties>` ドキュメントを示します。

```
select msgsend('Sending message with properties',
              'my_jms_provider?queue=queue.sample',
              message property 'color=red, shape=square')
```

```
select msgrecv('my_jms_provider?queue=queue.sample')
```

```
select rtrim (@@msgproperties)
```

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgproperties
  RTMS_MSGBODY_FORMAT='&apos;string&apos;';
  ASE_RTMS_CHARSET='1'
  ASE_RTMS_VERSION='&apos;1.0&apos;';
  ASE_VERSION='&apos;12.5.0.0&apos;';
  shape='&apos;square&apos;';
  color='&apos;red&apos;'; >
</msgproperties>
```

```
select rtrim (@@msgheader)
```

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgheader
  type='&apos;null&apos;';
  timestamp='1080092021000'
  replyto='&apos;queue.sample&apos;';
  redelivered='false'
  priority='4'
  messageid='&apos;ID:E4JMS-SERVER.73018656B39:1&apos;';
  ttl='0'
  destination='&apos;queue.sample&apos;';
  mode='2'
  correlation='&apos;null&apos;';
```

```
encoding='&apos;null&apos; ' >
</msgheader>
```

使用法

- 指定されたメッセージの `<msgheader>` または `<msgproperties>` ドキュメントには、メッセージ・ヘッダまたはメッセージ・プロパティのプロパティごとに1つの属性が含まれます。属性の名前は、プロパティの名前であり、属性の値は、プロパティの文字列値です。
- `<msgheader>` または `<msgproperties>` ドキュメントの属性の値は、XML エンティティに置き換えられます。msgpropvalue と msgpropname は、XML エンティティを属性値に暗黙的に置き換えます。
- msgrecv または msgconsume によって生成された `<msgheader>` または `<msgproperties>` ドキュメントには、プロパティの文字セットを指定する XML 宣言が含まれます。

Adaptive Server 固有のメッセージ・プロパティ

(JMS のみ) デバッグ、モニタリングなどに役立つため、Adaptive Server に固有の定義済みのプロパティが JMS メッセージのプロパティ部分に含まれています。これらのプロパティは通常、別の Adaptive Server から送信されたメッセージ、またはデバッグに役立つ可能性があるメッセージを処理します。

これらのメッセージ・プロパティの多くは、diagserver を実行している場合、または特定のトレース・フラグがオンになっている場合にのみ含まれます。“ASE_” で始まるすべてのプロパティは予約されています。msgsend または msgpublish を使用してこれらのプロパティを設定することはできません。表 3-3 は、これらのメッセージ・プロパティを示します。

表 3-3: JMS における Adaptive Server 固有のメッセージ

プロパティ	説明	使用するとき
ASE_RTMS_CHARSET	送信されたデータの文字セット・コード。	常時
ASE_MSGBODY_SCHEMA	メッセージ本文を記述するスキーマまたは null 値。ユーザが msgsend の一部としてメッセージ・スキーマを送信する場合にのみ、このスキーマは null 以外になります。 ASE_MSGBODY_FORMAT が xml である場合、このプロパティにはペイロードを記述する XML スキーマが含まれます。このスキーマは、値が 16K を超えた場合でもトランケートされません。	常時
ASE_MSGBODY_FORMAT	メッセージ本文のフォーマット。xml、string (サーバ文字セット)、binary、unicode (ネットワーク順序の unichar)。	常時
ASE_ORIGIN	送信元の Adaptive Server の名前。	diagserver とともに存在
ASE_RTMS_VERSION	Active Messaging を使用した Adaptive Server のバージョン。	常時
ASE_SPID	メッセージを送信する SPID。	diagserver とともに存在

プロパティ	説明	使用するとき
ASE_TIMESTAMP	メッセージが送信された時間を示す Adaptive Server のタイムスタンプ。	diagserver とともに存在
ASE_VERSION	メッセージをパブリッシュした Adaptive Server のバージョン。	常時
ASE_VERSIONSTRING	Adaptive Server のバージョン文字列。プラットフォーム、ビルド・タイプなどに関する情報を提供します。デバッグに役立ちます。	diagserver とともに存在

注意 これらのプロパティは、参照目的でのみ示されています。これらのプロパティは、将来、変更される場合があります。

キーワード

表 3-4 は、ASE Active Messaging に固有のキーワードと、これらのキーワードを有効に使用できる関数を示します。

表 3-4: ASE Active Messaging のダブル・キーワードとトリプル・キーワード

JMS または MQ	キーワード	キーワードを使用する有効なコマンドと関数
両方	message header	select msgsend(,, message header,,) select msgpublish(,,message header,,)
両方	message property	select msgsend(,, message property,,) select msgpublish(,,message property,,)
JMS	message selector	select msgrecv(,,message selector,,) select msgconsume(,,message selector,,)
JMS	with retain	select msgunsubscribe(,,with retain,,)
JMS	with remove	select msgunsubscribe(,,with remove,,)
両方	transactional messaging none	set transactional messaging none
両方	transactional messaging simple	set transactional messaging simple
両方	transactional messaging full	set transactional messaging full

ストアド・プロシージャ

ASE Active Messaging とともに使用するストアド・プロシージャを次に示します。

- 「[sp_configure 'enable real time messaging'](#)」 (51 ページ)
- 「[sp_engine](#)」 (55 ページ)
- 「[sp_msgadmin](#)」 (59 ページ)

`sp_msgadmin` とそのオプションは、基本となるメッセージ・プロバイダを設定または管理しません。たとえば、キューとトピックの作成、削除、アクセスは、メッセージング・プロバイダ・レベルで行う必要があります。

注意 `sp_addexceclass` は、`anyengine` と `lastonline` パラメータの MQ Q エンジンを受け入れません。

組み込み関数

この章の組み込み関数の項では、Active Messaging を管理する SQL 関数とオプション文字列の一般的なフォーマットについて説明します。Adaptive Server 固有のメッセージ・プロパティについては、[表 3-3 \(48 ページ\)](#) を参照してください。この章の SQL 関数を次に示します。

- キューとの間でメッセージを送受信する。
- メッセージ・トピックに関連するメッセージをパブリッシュ、サブスクライブ、コンシュームする。
- メッセージ・プロパティを処理する。

この章で示される関数とそれらのページ番号は、次のとおりです。

- 「[msgconsume](#)」 (71 ページ)
- 「[msgpropcount](#)」 (74 ページ)
- 「[msgproplist](#)」 (75 ページ)
- 「[msgpropname](#)」 (77 ページ)
- 「[msgproptype](#)」 (78 ページ)
- 「[msgpropvalue](#)」 (80 ページ)
- 「[msgpublish](#)」 (81 ページ)
- 「[msgrecv](#)」 (85 ページ)
- 「[msgsend](#)」 (99 ページ)
- 「[msgsubscribe](#)」 (133 ページ)

構文セグメント

この章の構文セグメントの項では、リアルタイム・メッセージングの管理に使用される SQL 構文と制約について説明します。

この章で示される構文とそれらのページ番号は、次のとおりです。

- 「[endpoint](#)」 (138 ページ)
- 「[option_string](#)」 (141 ページ)
- 「[sizespec](#)」 (142 ページ)
- 「[timespec](#)」 (143 ページ)

sp_configure 'enable real time messaging'

説明	リアルタイム・メッセージングを有効または無効にするか、現在のメッセージング設定を表示します。
構文	<pre>sp_configure "enable real time messaging" [, enable_or_disable] [, rtm_provider drop instance] [, instance_name]</pre>
パラメータ	<p>enable_or_disable</p> <p>"real time messaging" オプションを有効または無効にするかどうかを指定します。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • 1 – リアルタイム・メッセージングを有効にします。 • 0 – リアルタイム・メッセージングを無効にします。 <p>省略した場合、現在の "real time messaging" 設定が返されます。</p> <p>rtm_provider</p> <p>有効または無効にする Active Messaging プロバイダのタイプを指定します。JVM とクラスタ・サーバを指定する際にこのパラメータを使用します。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • eas_jms – EAServer に対してのみ "real time messaging" を有効または無効にします。 • ibm_mq – IBM MQ に対してのみ "real time messaging" を有効または無効にします。 • sonicmq_jms – SonicMQ JMS に対してのみ "real time messaging" を有効または無効にします。 • tibco_jms – Tibco JMS に対してのみ "real time messaging" を有効または無効にします。

drop instance

1つのインスタンスのメッセージング関連の設定を削除します。

instance_name

クラスタ・サーバ環境を作成するときに指定するインスタンスの名前です。このオプションを指定しない場合、現在のリアルタイム・メッセージングの設定によってクラスタ全体のオプションが指定されます。

例 例1 すべてのプロバイダのリアルタイム・メッセージングを有効にします。

```
sp_configure "enable real time messaging",1
```

リアルタイム・メッセージングを無効にするには、次のコマンドを使用します。

```
sp_configure "enable real time messaging",0
```

例2 MQのみでリアルタイム・メッセージングを有効にします。

```
sp_configure "enable real time messaging", 1 ,ibm_mq
```

リアルタイム・メッセージングを無効にするには、次のコマンドを使用します。

```
sp_configure "enable real time messaging", 0, ibm_mq
```

使用法

このストアド・プロシージャを使用しても以前の設定は上書きされません。たとえば、`tibco_jms` を有効にし、このストアド・プロシージャを実行して MQ を有効にした場合、MQ と `tibco_jms` の両方が有効になります。`tibco_jms` を無効にしても MQ に影響を与えず、MQ は引き続き有効になります。

enable_or_disable パラメータは、次の項目が正しくインストールおよび設定された場合にのみ機能します。

- 使用しているプラットフォームの適切な LD_LIBRARY_PATH
- プロバイダ DLL ライブラリ
- SYBASE ライセンス
- 製品の CD-ROM の SYBASE インタフェース・ライブラリ

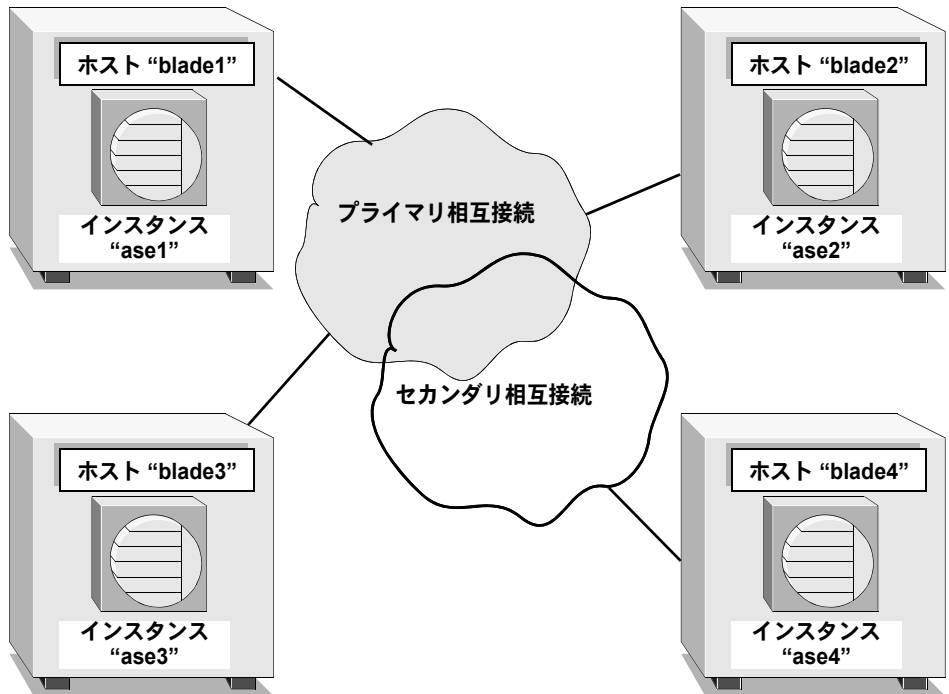
パスとファイル名の詳細については、『ASE インストール・ガイド』を参照してください。

クラスタ環境で sp_configure を使用する

Cluster Edition では、複数の Adaptive Server を設定して共有ディスク・クラスタとして実行できます。複数のマシンが共有ディスクのセットと高速プライベート相互接続 (たとえば、ギガバイト Ethernet) に接続することで、Adaptive Server は複数の物理ホストと論理ホストを使用してスケラブルに性能を向上できます。

次の例で使用されるクラスタ・システムで、クライアントは、“mycluster” という名前の共有ディスク・クラスタに接続します。“mycluster” には、マシン “blade1”、“blade2”、“blade3”、“blade4” でそれぞれ実行されている “ase1”、“ase2”、“ase3”、“ase4” インスタンスが含まれます。これらの例で、各ノードには1つのインスタンスが存在します。

図 3-1: sp_configure の例で説明されているクラスタ環境



例 1 すべての Active Messaging プロバイダに対してリアルタイム・メッセージングを有効にします。

```
sp_configure "enable real time messaging", 1
```

例 2 クライアントが“ase1”にログインした場合、IBM MQ 専用のインスタンス“ase1”を無効にします。

```
sp_configure "enable real time messaging", 0, "ibm_mq", "ase1"
```

“ase1”用に生成されたインスタンス固有の構成オプションの値は 26 です。“real time”の設定値はバイトであり、さまざまなビットによってリアルタイムの機能が異なります。

- 0x1 – すべてが有効。
- 0x2 – tibco_jms が有効。
- 0x4 – ibm_mq が有効。
- 0x8 – eas_jms が有効。
- 0x10 – sonicmq_jms が有効。

例 3 インスタンス “ase1” でインスタンス固有の **enable real time messaging** 設定オプションのステータスを表示します。値は 26 です。

```
sp_configure "enable real time messaging", null, null, ase1
```

例 4 インスタンス “ase1” でインスタンス固有の **enable real time messaging** 設定オプションを削除します。このプロシージャを実行すると、“ase1” はクラスタ全体の **enable real time messaging** 設定オプションの使用を開始し、ステータスは 1 になります。

```
sp_configure "enable real time messaging", 0, "drop instance", "ase1"
```

drop instance を実行すると、新しい設定値は実行値と同じになります。

例 5 インスタンス全体の **enable real time messaging** 設定オプションのステータスを表示します。この値は 1 です。

```
sp_configure "enable real time messaging"
```

sp_engine

説明	Q エンジンをオンラインまたはオフラインにできるようにします。
構文	<pre>sp_engine "online offline can_offline shutdown q_online q_offline q_can_offline q_shutdown", [engine_id]</pre>
パラメータ	<p>can_offline エンジンをオフラインにできるかどうかに関する情報を返します。エンジンをオフラインにできない場合は、エンジンがオフラインになるのを妨げる Adaptive Server セッションの <code>spid</code> が表示されます。このパラメータを使用して Q エンジンを指定することはできません。</p> <p>engine_id エンジンの ID です。 指定するエンジンのタイプは、コマンド (<code>online</code>、<code>q_online</code> など) と一致する必要があります。たとえば、<code>q_offline</code> を使用して Q 以外のエンジンを指定したり、<code>offline</code> を使用して Q エンジンを指定することはできません。 このパラメータは、<code>offline</code>、<code>q_offline</code>、<code>can_offline</code>、<code>q_can_offline</code>、<code>shutdown</code>、<code>q_shutdown</code> で必須です。 このパラメータは、<code>online</code>、<code>q_online</code> では必須ではありません。</p> <p>online エンジンをオンラインにします。<code>sp_configure "max online Q engines"</code> の値を、現在オンラインである Q エンジンの数より大きくしてください。<code>online</code> は予約キーワードなので、引用符を使用してください。このパラメータを使用して Q エンジンを指定することはできません。</p> <p>offline エンジンをオフラインにします。<code>engine_id</code> を使用して、オフラインにするエンジンを指定することもできます。このパラメータを使用して Q エンジンを指定することはできません。</p> <p>q_can_offline Q エンジンをオフラインにできるかどうかに関する情報を返します。エンジンをオフラインにできない場合は、エンジンがオフラインになるのを妨げる Adaptive Server セッションの <code>spid</code> が表示されます。Q エンジンをオフラインにできるかどうかを指定するには、<code>engine_id</code> を使用する必要があります。</p> <p>q_offline Q エンジンをオフラインにします。オフラインにするエンジンを指定するには、<code>engine_id</code> を使用する必要があります。</p> <p>q_online 次の Q エンジンをオンラインにします。</p>

q_shutdown

Q エンジンを強制的にオフラインにします。このエンジンと結び付きがあるタスクがある場合、それらは5分間の待機の後、強制終了されます。**shutdown**は予約キーワードなので、引用符を使用してください。Q エンジンをシャットダウンできるかどうかを指定するには、**engine_id**を使用する必要があります。

shutdown

エンジンを強制的にオフラインにします。このエンジンと結び付きがあるタスクがある場合、それらは5分間の待機の後、強制終了されます。**shutdown**は予約キーワードなので、引用符を使用してください。このパラメータを使用してQ エンジンを指定することはできません。

例**例 1** Q エンジンを手動でオンラインにします。

```
sp_engine 'q_online'
go

(return status=0)

02:00000:00000:2005/06/08 12:52:21.09 kernel Network and device connection limit is
1014.
02:00000:00000:2005/06/08 12:52:21.24 server Initialized Unilib version 7.2.
02:00000:00000:2005/06/08 12:52:21.24 kernel Q engine 2, os pid 20025 online
02:00000:00000:2005/06/08 12:52:21.33 kernel LDAP dynamic libraries successfully
loaded.
02:00000:00000:2005/06/08 12:52:21.38 kernel IBM MQ dynamic libraries successfully
loaded.
```

例 2 Q エンジンをオフラインにします。

```
1> select engine, status from sysengines
2> go

engine status
-----
0 online
1 online_q
2 online_q
(3 rows affected)

1> sp_engine 'q_offline', 1
2> go

(return status = 0)
00:00000:00000:2005/06/08 12:55:54.25 kernel engine
2, os pid 20025 offline

1> select engine, status from sysengines
2> go

engine status
-----
0 online
```

```

    1 online_q
(2 rows affected)

```

例 3 Q エンジンをオフラインにできるかどうかを確認します。

```

1> select engine, status from sysengines
2> go

engine status
-----
0 online
1 online_q
(2 rows affected)

1> sp_engine 'q_can_offline', 1
2> go

spid: 13 has outstanding rtms-connection connections.

```

例 4 Q エンジンを停止します。

```

1> select engine, status from sysengines
2> go

engine status
-----
0 online
1 online_q
(2 rows affected)

1> sp_engine 'q_shutdown', 1
2> go

(return status = 0)

1> select engine, status from sysengines
2> go

engine status
-----
0 online

(1 row affected)

```

使用法

- **online**、**offline**、**can_offline**、**shutdown** は、Q 以外のエンジンにだけ反映されます。これらのパラメータを使用して Q エンジンを指定すると、エラーが表示されます。
- **q_online**、**q_offline**、**q_can_offline**、**q_shutdown** は、Q エンジンにだけ反映されます。これらのパラメータを使用して Q 以外のエンジンを指定すると、エラーが表示されます。
- エンジン 0 を停止にしたり、オフラインしたりすることはできません。

- エンジンのステータスと現在オンラインのエンジンを、次のクエリを使って判断できます。

```
select engine, status from sysengines
       where status = "online"
```

- **max online Q engines** の値が、その時点の **online** ステータスのエンジン数より大きく、CPU が十分に追加エンジンをサポートできる場合にだけ、エンジンをオンラインにできます。
- **engine offline** は、そのエンジンと結び付きがあるサーバ・プロセスがある場合、失敗するか、すぐには有効にならない可能性があります。
- クラスタ環境で、**sp_engine** はローカル・インスタンスのエンジンのみで機能します。

パーミッション

エンジンをオンラインまたはオフラインにするには、システム管理者権限が必要です。

sp_msgadmin

説明

メッセージ関連の情報を設定および管理します。

構文

```
sp_msgadmin 'config', ['jvmlogging', logging_level
    | 'jvmpropertyfile', filepath
    | 'jvmlogfile', filepath
    | 'jvmmaxthreads', thread_number
    | 'jvminthreads', thread_number
    | 'jvmthreadtimeout', thread_timeout
    | 'jvm', jvm_parameter]

sp_msgadmin 'default', 'login', provider_name, provider_login,
    provider_password

sp_msgadmin 'help'
    [, 'list' | 'register' | 'default' | 'remove']

sp_msgadmin 'list',
    [| 'login', provider_name, [login_name]
    | 'provider' [, provider_name]
    | 'subscription' [, subscription_name]]

sp_msgadmin 'register',
    ['provider', provider_name, provider_class,
    messaging_provider_URL
    | 'login', provider_name, local_login, provider_login,
    provider_password [, role_name]
    | 'subscription', subscription_name, endpoint[, selector
    [, delivery_option [, durable_name, client_id]]]]

sp_msgadmin 'remove',
    ['provider', provider_name
    | 'login', provider_name, local_login [, role]
    | 'subscription', subscription_name]

sp_msgadmin 'show',
    showtype, provider[, options_clause]
```

パラメータ

sp_msgadmin 'config'

Java 仮想マシン (JVM)、または MQ SSL を使用する Adaptive Server のキー・レポジトリのファイル・パスに対してさまざまな設定を指定できます。ASE Active Messaging を再び有効にすると、設定された値が有効になります。sp_msgadmin 'config' のオプションを次に示します。

- 'jvmlogging', *logging_level* – 設定したレベルより高いコード内のトレース情報だけを表示するようにメッセージング・サービスを設定できます。

logging_level は、Apache log4j ロギング・システムを使用してレベルを指定します。*logging_level* の値は次のとおりです。

- 'all' – コード内のすべてのトレース情報を返します。
- 'debug' – JVM デバッグ情報を返します。
- 'fatal' – JVM の致命的な情報を返します。
- 'off' – ロギングをオフにします。
- 'info' – *logging_level* のデフォルト値。情報レベルのログ情報を返します。
- 'error' – エラー・ログ情報だけを返します。

Apache log4j Web サイト (<http://logging.apache.org/log4j/>) を参照してください。

- 'jvmpropertyfile', *filepath* – 設定で JVM が使用するプロパティ・ファイルを指定します。

filepath には、環境変数の使用など、プロパティ・ファイルの有効なパスを指定できます。プロパティ・ファイルのデフォルト値は `$$SYBASE/$$SYBASE_ASE/lib/rtms.properties` です。

- 'jvmlogfile', *filepath* – 設定で JVM が使用するログ・ファイルへのパスを定義します。

JVM のログ情報は、コンソールに表示され、1 つのログ・ファイルに書き込まれます。ログ・ファイルがその最大サイズの 5MB に到達するたびに、JVM は、新しいログ・ファイルを自動的に作成し、ファイルの末尾に新しい番号 (*XXX.2*, *XXX.3* など) を追加します。

JVM ログ・ファイル *filepath* のデフォルト値は `$$SYBASE/$$SYBASE_ASE/rtms.log` です。

クラスタ環境で Java Active Messaging サーバを起動すると、実際のログ・ファイルは値と `@@nodename` の組み合わせになります。たとえば、ノード "s1" に対して sp_msgadmin を実行すると、実際の JVM ログ・ファイルは、次のようになります。

`$$SYBASE/$$SYBASE_ASE/jrtms_sl.log:`

```
1> sp_msgadmin 'config', 'jvmlogfile', '$$SYBASE/$$SYBASE_ASE/jrtms.log'
```

設定された JVM ログ・ファイルの条件

- `$$SYBASE/$$SYBASE_ASE/jrtms.log` などのファイル拡張子を持つ場合。ここで、`jrtms.log` のファイル名には、`log` ファイル拡張子名が含まれます。インスタンス “ase1” の実際のファイル名は `$$SYBASE/$$SYBASE_ASE/jrtms_ase1.log` となります。
- `$$SYBASE/$$SYBASE_ASE/jrtms` などのファイル拡張子を持たない場合。ここで、ファイル名はファイル拡張子が付かない `jrtms` です。インスタンス “ase1” の実際のファイル名は `$$SYBASE/$$SYBASE_ASE/jrtms_ase1` となります。

- 'jvmmxthreads', *thread_number* – JVM サーバのスレッド・プールで一度に実行する Java スレッドの最大数を指定します。

thread_number の値は、`jvmmthreads` の値より大きくなければなりません。デフォルト値は 10 です。

- 'jvmmthreads', *thread_number* – JVM サーバのスレッド・プールで一度に実行する Java スレッドの最小数を指定します。*thread_number* の値には 0 以上を指定できますが、`jvmmxthreads` の値より小さくなくてはなりません。デフォルト値は 0 です。
- 'jvmthreadtimeout', *thread_timeout* – 一定期間動作しなかった場合にスレッドが自動的に破棄されるようにします。

thread_timeout は、スレッドが破棄されるまでの時間 (秒単位) です。デフォルト値は 600 (10 分) です。

- 'jvm', *jvm_parameter* – JVM の起動時に Java に渡すパラメータを定義します。

jvm_parameter は、有効な Java パラメータ文字列の名前です。デフォルト値は、“-Xmx500m” です。これは、割り付けられた RAM の 500MB を使用して起動するように Java を指定する汎用フラグです。Java -Xmx フラグの詳細については、Java Web サイト (<http://java.sun.com>) を参照してください。

- 'ibmmq_keystore', *keystore_name* – SSL 経由で WebSphere MQ との間でメッセージを送受信できるように Adaptive Server のキー・レポジトリ・ファイル・パスを設定します。

keystore_name は、キーと証明書が保存されているキー・データベース・ファイルのロケーションです。

sp_msgadmin 'default'

デフォルトを指定します。sp_msgadmin 'list' の場合、指定されたメッセージ・プロバイダのデフォルトのログインを指定する構文のリストを示します。次の選択肢があります。

- 'login' – 'default' とともに使用する場合に、デフォルトのログインを指定します。

注意 endpoint が MQ キュー・マネージャである場合は、sp_msgadmin 'default', 'login' を使用することはできません。

- *provider_name* – 登録するメッセージング・プロバイダです。最大 30 文字の長さを指定できます。
- *provider_login* – メッセージ・プロバイダに接続する際に *local_login* がマッピングするメッセージング・プロバイダのログイン名です。*provider_login* は、*provider_name* からのメッセージを送受信する際にプロバイダが使用するデフォルトのログインでもあります。
- *provider_password* – *provider_login* のパスワードです。

注意 (クラスタ環境のみ) sp_msgadmin default を使用してクラスタ環境のデフォルトのログインを定義する場合は、クラスタ全体に対する設定を使用できます。

sp_msgadmin 'help'[, 'list' | 'register' | 'default' | 'remove']

sp_msgadmin またはそのパラメータに関する構文情報を提供します。

sp_msgadmin 'list'

メッセージ・プロバイダ、ログイン、またはサブスクリプションに関する構文情報のリストを示します。

- 'login'[, *provider_name*,[*login_name*] – 特定のメッセージング・プロバイダのログイン・マッピング、またはすべてのメッセージング・プロバイダのログインに関する情報のリストを示します。
- 'provider'[, *provider_name*] – メッセージ・プロバイダを指定し、特定のメッセージング・プロバイダまたはすべてのメッセージング・プロバイダに関する情報のリストを示します。
- 'subscription'[, *subscription_name*] – 特定のサブスクリプションまたはすべてのサブスクリプションに関する情報のリストを示します。

sp_msgadmin 'register'

メッセージング・プロバイダ、ログイン、またはサブスクリプションを登録します。次の選択肢があります。

- **sp_msgadmin 'register' provider** – メッセージング・プロバイダを登録します。各パラメータの意味は、次のとおりです。
 - *provider_name* – メッセージング・プロバイダの名前です。
 - *provider_class* – 追加するメッセージング・プロバイダのクラスです。有効な値は次のとおりです。
 - EAS_JMS
 - TIBCO_JMS
 - IBM_MQ
 - SONIC_MQ
 - *messaging_provider_URL* – 登録するメッセージング・プロバイダの URL です。
- **sp_msgadmin 'register' 'login'** – ログイン・マッピングを登録します。各パラメータの意味は、次のとおりです。
 - *provider_name* – 前に登録したプロバイダの名前です。最大 30 文字の長さを指定できます。
 - *local_login* – ローカル・ログインにマップする Adaptive Server ログインです。
 - *provider_login* – メッセージ・プロバイダに接続する際に *local_login* がマップするメッセージング・プロバイダのログイン名です。
 - *provider_password* – *provider_login* のメッセージング・プロバイダのパスワードです。
 - *role_name* – SQL 役割名です。*role_name* を指定すると、*local_login* は無視され、*provider_login* と *provider_password* が *role_name* に適用されます。

注意 endpoint が MQ キュー・マネージャである場合は、sp_msgadmin 'register', 'login' を使用することはできません。

- `sp_admin 'register' 'subscription'` – サブスクリプションを登録します。パラメータの意味は、次のとおりです。
 - `subscription_name` – サブスクリプション名です。
 - `endpoint` – サブスクリプションが送信されるトピックです。「msgsend」(99 ページ)の「`endpoint`」を参照してください。

注意 `endpoint` が MQ キュー・マネージャである場合は、`sp_msgadmin 'register', 'subscription'` を使用することはできません。

- `selector` – 対象のメッセージをクライアントが選択できるようにするメッセージ・フィルタです。「msgrecv」(85 ページ)のフィルタを参照してください。
- `delivery_option` – SQL セッションがパブリッシュするメッセージをコンシュームできるかどうかを指定します。有効な値は次のとおりです。
 - `local` – SQL セッションはパブリッシュするメッセージをコンシュームできます。
 - `nonlocal` – SQL セッションはパブリッシュするメッセージをコンシュームできません。
 - `null` – 値が `local` であると想定します。
- `durable_name` – 文字列値です。「`client_id`」を参照してください。
- `client_id` – 持続的としてサブスクリプションを識別する際にメッセージング・プロバイダによって使用される ID です。`client_id` は文字列値です。`client_id` または `durable_name` のどちらかを指定した場合、持続的としてサブスクリプションを指定するもう一方も指定する必要があります。そうしないと、サブスクリプションは非持続的になります。

`client_id` と `durable_name` の組み合わせによって、メッセージ・プロバイダの持続的なサブスクリプションが識別されます。これらの組み合わせはユニークでなければなりません。

`client_id` のユニーク性は、メッセージング・プロバイダ全体に拡張されます。JMS は、特定の `client_id` が指定された時間に一度だけ接続できるようにします。たとえば、あるアプリケーションが、指定された `client_id` を使用して持続的なサブスクリプションを既に持っており、別のアプリケーションと同時に接続する場合、別のアプリケーションによって指定された `client_id` と同じ値を使用することはできません。

持続的なサブスクリプションは、クライアントが接続されていないときでも存在します。メッセージング・プロバイダは、クライアントが接続されていない間でも受信したメッセージを保存します。

非持続的なサブスクリプションは、クライアントが接続されている間だけ存在します。メッセージング・プロバイダは、クライアントが接続されていない間に受信したメッセージを破棄します。

注意 (クラスタ環境のみ) クラスタ環境の `sp_msgadmin 'register'` を使用してプロバイダ、ログイン、サブスクリプションの各情報を登録する場合、登録はクラスタ全体に適用されます。

`sp_msgadmin 'remove'`

メッセージ・プロバイダ、ログイン、またはサブスクリプションを削除します。

- `'provider', provider_name` — 次のパラメータを使用して前に定義されたメッセージング・プロバイダを削除します。

```
sp_msgadmin 'register', 'provider', provider_name
```

`provider_name` は、削除するメッセージング・プロバイダを参照するエイリアスです。

- `'login', provider_name, local_login [, role]` — 次の呼び出しによって定義された Adaptive Server ログインとサービス・プロバイダのログインの間で前に作成されたマッピングを削除します。

```
sp_msgadmin 'register', 'login', local_login, ...
```

ここで、

- `local_login` — ローカル・ログインにマップする Adaptive Server ログインです。
- `role` — 役割です。
- `'subscription', subscription_name` — 次の構文によって作成されたサブスクリプションを削除します。

```
sp_msgadmin 'register' 'subscription', subscription_name, ...
```

sp_msgadmin 'show'

Adaptive Server バージョン 15.0.2 ESD #1 以降を必要とし、指定されたキュー・マネージャで一部の MQ オブジェクトに関する情報を表示します。各パラメータの意味は次のとおりです。

- **showtype** – 表示する WebSphere MQ プロセスまたはオブジェクトを指定できるようにします。
 - **qmgr** – キュー・マネージャの名前です。
 - **queues** – キュー・マネージャに属するすべてのキューとそのタイプです。
 - **channels** – キュー・マネージャに属するすべてのチャンネルとそのタイプです。
- **provider** – メッセージング・プロバイダを指定します。[「endpoint」\(138 ページ\)](#) で説明されているフル・パスのフォーマットを使用します。
- **option_string** – オプションのリストです。[表 3-5](#) を参照してください。

表 3-5: option_string のタイプと値

データ型	値	デフォルト	説明
timeout	0 と $(2^{31}-1)$ の間の timespec	30000 (30 秒)	各応答メッセージで WebSphere MQ Administration Interface が待機する必要がある最大時間(ミリ秒単位)を指定する。
replyqueue	string	なし	コマンド・サーバは応答メッセージをキューに返す。このオプションを定義しない場合、コマンド・サーバはメッセージをオープン の SYSTEM.DEFAULT.MODEL.QUEUE によって作成された動的キューに返します。

例**例 1 (JMS)** JVM のレベルをログに記録します。

```
sp_msgadmin 'config', 'jvmlogging', 'info'
```

例 2 (JMS) プロパティ・ファイルとして `/usr/1.prop` を指定します。

```
sp_msgadmin 'config', 'jvmpropertyfile', '/usr/1.prop'
```

例 3 (JMS) `$$SYBASE/$$SYBASE_ASE/rtms.log` としてログ・ファイル・パスを定義します。

```
sp_msgadmin 'config', 'jvmlogfile', '$$SYBASE/$$SYBASE_ASE/rtms.log'
```

例 4 (JMS) JVM サーバのスレッド・プールのスレッドの最大数を 100 に指定します。

```
sp_msgadmin 'config', 'jvmmaxthreads', 100
```

例 5 (JMS) 自動的に破棄されるまでのスレッドのアイドル時間を 10 分に指定します。

```
sp_msgadmin 'config', 'jvmthreadtimeout', 600
```

例 6 (JMS) -Xmx500m フラグを使用して 500MB の RAM を持つ JVM を起動します。

```
sp_msgadmin 'config', 'jvm', '-Xmx500m'
```

例 7 (JMS) クラスが EAS_JMS で URL が iiop://localhost:7222 の “eas_1” メッセージ・プロバイダを登録します。

```
sp_msgadmin 'register', 'provider',
            'eas_1', 'eas_jms', 'iiop://localhost:7222'
```

例 8 (JMS) 送信または受信するために指定されたメッセージング・プロバイダを使用するときに、マップされていないすべての Adaptive Server ログインに適用されるデフォルトのログインを指定します。

```
sp_msgadmin 'default', 'login', 'my_eas', 'eas_user', 'eas_password'
```

注意 sp_msgadmin 'register', 'provider' を呼び出して、最初に、*provider_name* を登録してください。

例 9 (JMS) デフォルトのログインを指定します。

```
sp_msgadmin 'default', 'login', 'one_jms_provider', 'loginsa',
            'abcdef123456'
```

例 10 (JMS) “loginsa” のログインを使用するユーザの詳細を示します。

```
sp_msgadmin 'list', 'login', 'my_jms_provider', 'loginsa'
```

例 11 (JMS) メッセージング・プロバイダ・ログイン “jms_user1” とメッセージング・プロバイダ名 “my_jms_provider” を使用してログイン “ase_login1” を登録します。

```
sp_msgadmin 'register', 'login', 'my_jms_provider', 'ase_login1',
            'jms_user1', 'jms_user1_password'
```

例 12 (JMS) メッセージング・プロバイダ・ログイン “jms_user1” を持つログインと sa_role パーミッションを持つすべての Adaptive Server ログインで使用される指定されたパスワードを登録します。

```
sp_msgadmin 'register', 'login', 'my_jms_provider', null, 'jms_user1',
            'jms_user1_password', 'sa_role'
```

例 13 (JMS) クラスが TIBCO_JMS で IP が 10.23.233.32:4823 のメッセージング・プロバイダ “my_jms_provider” をアドレスとして登録します。

```
sp_msgadmin 'register', 'provider', 'my_jms_provider', 'TIBCO_JMS',
            'tcp://10.23.233.32:4823'
```

例 14 (JMS) “durable_sub1” という名前の持続的サブスクリプションを登録します。sp_msgadmin 'list' は新しいサブスクリプションに関する情報を表示します。

```
sp_msgadmin 'register', 'subscription', 'durable_sub1',
            'my_jms_provider?topic=topic.sample', null, null, 'durable1', 'client1'
sp_msgadmin 'list', 'subscription', 'durable_sub1'
```

例 15 (JMS) “subscription_1” を非持続的サブスクリプションとして登録します。

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
            'my_jms_provider?topic=topic.sample'
```

注意 最初に、sp_msgadmin register, provider を使用して “my_jms_provider” を登録する必要があります。

例 16 (JMS) デフォルトのログインを削除します。

```
sp_msgadmin 'remove', 'login', 'my_jms_provider'
```

例 17 (JMS) メッセージング・プロバイダ “my_jms_provider” に関連付けられている Adaptive Server ログイン “ase_login1” を削除します。

```
sp_msgadmin 'remove', 'login', 'my_jms_provider', 'ase_login1'
```

例 18 (JMS) “my_jms_provider” の役割 sa_role のすべてのログインを削除します。

```
sp_msgadmin 'remove', 'login', 'my_jms_provider', null, 'sa_role'
```

例 19 (MQ) SSL を使用できるように Adaptive Server のキー・レポジトリを設定します。ここでキー・データベースのファイル・パスは /var/mqm/clients/ssl/KeyringClient.kdb です。

```
sp_msgadmin 'config', ibmmq_keystore,
            'var/mqm/clients/ssl/KeyringClient'
```

例 20 (MQ) クラスが IBM_MQ で URL が chan11/TCP/host1(5678) のメッセージング・プロバイダ “mq_provider_1” を登録します。

```
sp_msgadmin 'register', 'provider', 'mq_provider_1', 'ibm_mq',
            'chan11/TCP/host1(5678)'
```

例 21 (MQ) 受信ポートが 3150 のマシン “bigcrunch” のキュー・マネージャ名を表示します。

```
sp_msgadmin 'show', 'QMGR', 'ibm_mq:/tcp/bigcrunch(3150)'
```

Name

TEST

例 22 (MQ) キュー・マネージャの名前を表示します。キュー・マネージャは、受信ポートが 3150 であるマシン “bigcrunch” 上に存在します。応答メッセージは Q1 キューに配置され、Adaptive Server が応答メッセージを待機する時間は 20 ミリ秒です。

```
sp_msgadmin 'show', 'QMGR', 'ibm_mq:channel1/tcp/bigcrunch(3150)',
            'timeout=20, replyqueue=Q1'
```


例 23 (MQ) キュー・マネージャ上のすべてのキューを表示します。応答メッセージは Q1 キューに配置され、Adaptive Server が応答メッセージを待機する時間は 20 ミリ秒です。

```
sp_msgadmin 'show', 'queues', 'ibm_mq:/tcp/bigcrunch(3150)',
            'timeout=20, replyqueue=Q1'
```

Name	Type
Q1	LOCAL
SYSTEM.MQSC.REPLY.QUEUE	MODEL
RQ1	REMOTE
AQ1	ALIAS
...	

例 24 (MQ) キュー・マネージャ上のすべてのチャンネルを表示します。

```
sp_msgadmin 'show', 'channels', 'ibm_mq:/tcp/bigcrunch(3150)'
```

Name	Type
SNCH1	SENDER
SECH2	SERVER
RCCH3	RECEIVER
CHL5	SRVCONN
...	

例 25 (SonicMQ) 指定された終了ポイントに “sub1” を呼び出し、Q1 キューに配置されたサブスクリプションを登録します。

```
sp_msgadmin register, subscription, sub1,
            'sonicmq_jms:tcp://mysonic:7223??topic=T1,user=sonic_usr, password=sonic_pwd'
```

使用法

トランザクション内で `sp_msgadmin` を使用することはできません。

`sp_msgadmin 'register'`

- メッセージ・プロバイダに接続する際にログイン名が使用された場合、ログイン名は次の順序で解決されます。
 - a 終了ポイントで指定されている場合は、指定された明示的なログイン名とパスワード。
 - b 現在の Adaptive Server ログインの明示的なログイン・マッピング。
 - c メッセージ・プロバイダのデフォルトのログイン名とパスワードと、Adaptive Server ログインに対応する役割。
 - d 特定の役割との関連付けを持たない、メッセージ・プロバイダのデフォルトのログイン名とパスワード。
 - e 上記のいずれも該当しない場合は、null ログイン名とパスワード。
- Adaptive Server ログインとメッセージング・プロバイダ・ログインの間のログイン・マッピングは、別のマッピング・セットを使用してこれを削除して再登録した場合にのみ、変更できます。

- MQのみ – 登録されたプロバイダを使用して終了ポイントを入力した場合、msgsubscribe、msgunsubscribe、msgpublish、msgconsume を使用すると、エラーが返されます。
- sp_msgadmin の変形に共通の使用法については、[「sp_msgadmin」\(59 ページ\)](#) を参照してください。

sp_msgadmin 'remove'

- メッセージング・プロバイダを削除しても、このメッセージ・プロバイダに転送中のメッセージ(送受信中のメッセージ)は影響を受けません。
- sp_msgadmin 'remove' は、メッセージ・プロバイダに実行中の接続に影響を与えません。つまり、指定されたメッセージ・プロバイダに実行中の接続が存在する場合にメッセージ・プロバイダ、ログイン、またはデフォルトが削除された場合、その接続は影響を受けません。ただし、Sybase では、これを行わないことをおすすめします。
- *role_name* を指定した場合は、*local_login* を null として指定する必要があります。

sp_msgadmin 'config'

- sp_msgadmin 'config' は、JMS の場合にのみ使用できます。
- すべての sp_msgadmin 'config' は、sysattributes テーブルに保存されます。値を取得するには、次のコマンドを実行します。

```
1> select * from sysattributes where class = 21
```

sysattributes の詳細については、『リファレンス・マニュアル：テーブル』を参照してください。

- sp_msgadmin 'config' で使用可能なすべてのパラメータは、'jvm' を除き、動的に設定されます。

パーミッション

msgsend 関数と msgrecv 関数を実行するには、messaging_role を持っている必要があります。

次のコマンドを発行するには、messaging_role と sso_role パーミッションを持っている必要があります。

- sp_msgadmin 'default'
- sp_msgadmim 'register'
- sp_msgadmin 'remove'

次のコマンドはどのユーザも発行できます。

- sp_msgadmim 'help'
- sp_msgadmin 'list'

msgconsume

説明 EAServer JMS のみ - 別のトピックにパブリッシュされるメッセージをコンシュームするための SQL インタフェースを提供します。

構文

```
msgconsume_call ::=
    msgconsume (subscription_name, option_and_returns)
                subscription_name:= basic_character_expression
                option_and_returns ::= [option_clause] [returns_clause]
                option_clause ::= [,] option_option_string
                returns_clause ::= [,] returns_sql_type
                subscriber_name ::= basic_character_expression
                sql_type ::=
                    varchar(integer) | java.lang.String | text
                    | varbinary(integer) | image
```

パラメータ *basic_character_expression*
データ型が char、varchar、または java.lang.String の Transact-SQL クエリ式です。

option_string

「[option_string](#)」(141 ページ) で説明されている *option_string* の一般的なフォーマットです。メッセージのコンシュームに使用する特殊なオプションについては、[表 3-6](#) を参照してください。

表 3-6: msgconsume の option および option_string の値

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
timeout	-1、0 ~ (2 ³¹ - 1) の timespec	-1	デフォルトでは、msgconsume はメッセージ・バスから次のメッセージを読み込むまで、メッセージをブロックします。timeout が -1 でない場合は、timeout 間隔が経過してもメッセージが読み込まれないときに msgconsume は null を返します。値はミリ秒単位の数値です。 timeout は timespec オプションを使用します。「 timespec 」(143 ページ) を参照してください。
requeue	<i>string</i>	なし	Adaptive Server が処理できないメッセージを再キューイングする場所、キュー、またはトピックの名前です。requeue が指定されていない場合、メッセージを処理できないとエラー・メッセージが表示されます。指定された終了ポイントは、msgconsume および msgrecv と同じメッセージング・プロバイダである必要があります。

subscription_name

メッセージをコンシュームするサブスクリプションの名前です。

returns

返す句を指定します。

SQL_type

SQL 文で使用するデータ型です。

返されるデータ型を指定しない場合、デフォルトは `varchar(16384)` です。有効な SQL データ型は次のとおりです。

- `varchar(n)`
- `text`
- `java.lang.String`
- `varbinary(n)`
- `image`
- `univarchar(n)`

例 **例 1** メッセージをコンシュームする前に、クライアント・サーバでサブスクリプションを定義します。

```
sp_msgadmin 'register', 'subscription', 'subscription_1',  
  'my_jms_provider?topic=topic.sample,user=user1,password=pwd',  
  'Supplier=12345',null,'durable1', 'client1'
```

サブスクリプションをサブスクライブしてから、サブスクリプションからメッセージをコンシュームすることをおすすめします。

```
select msgsubscribe('subscription_1')  
declare @mymsg varchar(16384)  
select @mymsg = msgconsume('subscription_1')
```

例 2 変数を宣言し、指定されたサブスクリプションからメッセージを受け取ります。

```
declare @mymsg varchar (16384)  
select @mymsg = msgconsume('subscription_1',  
  option 'timeout=0')
```

メッセージを転送します。

```
select msgsend  
  (msgconsume('subscription_1'), 'my_jms_provider?queue=queue.sample')
```

メッセージを読み込み、`varbinary` として返します。

```
select msgconsume('subscription_1' returns varbinary(500))
```

使用法

- 認識できないオプション名があると、エラーになります。
- `msgconsume` は、`subscription_name` によって指定された `end_point` および `message_filter` が定義するトピックからメッセージを読み込みます。タイムアウトまたはエラーの場合、`null` 値が返されます。それ以外の場合は、読み込まれたメッセージの本文が返されます。

- Adaptive Server が処理するメッセージ・タイプは、`message`、`text`、または `bytes` のみです。処理できないメッセージがあり、`requeue` が指定されていない場合は、そのメッセージは元のキューに残されます。これ以降、読み込みを続けて同じメッセージがあった場合、同じ結果になります。これを回避するには `requeue` を指定します。`requeue` が指定されると、Adaptive Server が処理できないメッセージは指定されたキューに配置されます。

指定された終了ポイントは、`msgconsume` で使用された終了ポイントと同じメッセージング・サービス・プロバイダに存在する必要があります。

- メッセージング・プロバイダが `message`、`text`、または `bytes` 以外のタイプのメッセージを発行し、`requeue` が指定されていない場合、Adaptive Server はエラー・メッセージを返します。
- サブスクリプションがサブスクライブされていない場合、Adaptive Server は `msgconsume` の実行時に自動的にサブスクライブします。
- `msgconsume` を呼び出すと、次のような結果になります。
 - 返される値は、メッセージ・プロバイダによって返され、指定された `returns` タイプに変換された `message_body` 値です。
 - `@@msgheader` および `@@msgproperties` の値は `<msgheader>` および `<msgproperties>` ドキュメントに設定されます。これらには、`msgconsume` によって返されるメッセージのプロパティが含まれています。
`<msgheader>` および `<msgproperties>` の詳細については、「[メッセージ関連のグローバル変数](#)」(40 ページ)を参照してください。
 - `msgpropvalue` を使用して、XML ドキュメント `<msgheader>` と `<msgproperties>`、およびその他の関連する関数から特定のプロパティの値を抽出できます。「[msgpropvalue](#)」(80 ページ)を参照してください。

パーミッション

`msgconsume` を実行するには、`messaging_role` が必要です。

msgpropcount

説明	<code><msgheader></code> および <code><msgproperties></code> ドキュメントから <code>msg_doc</code> 内のプロパティまたは属性の数を抽出して返します。
構文	<pre>msgpropcount_call ::= msgpropcount(<i>msg_doc</i>) <i>msg_doc</i> ::= <i>basic_character_expression</i> <i>prop_name</i>::= <i>basic_character_expression</i></pre>
パラメータ	<p><code>msgpropcount_call</code> msgpropcount 関数を使用する要求を作成します。</p> <p><code>msg_doc</code> <code><msgheader></code> または <code><msgproperties></code> XML ドキュメントで、形式は <code>basic_character_expression</code> です。 <code>msg_doc</code> を指定しない場合、msgpropcount は <code>@@msgproperties</code> の現在の値を使用します。</p> <p><code>prop_name</code> <code>basic_character_expression</code> 形式の値またはタイプを抽出するプロパティ名です。</p>
例	<p>この例では、msgrecv からの呼び出しが、trade_name という名前の 1 つのプロパティと “Acme Maintenance” (“Quick & Safe”) という値を持つメッセージを返します。この場合、<code>@@msgproperties</code> グローバル変数の値は次のようになります。</p>

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
  <msgproperties
    trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
  </msgproperties>
```

アンパサンドと Quick & Safe という句を囲む引用符は、XML 規約に従って、`"` および `&` という XML エンティティで置き換えられます。

最後に取得されたメッセージのプロパティ数を取得します。

```
select msgpropcount (@@msgproperties)
```

msgproplist

説明	<code><msgheader></code> および <code><msgproperties></code> ドキュメントから、すべてのプロパティ属性が <code>msg_doc</code> である <code>option_string</code> 形式の文字列を抽出して返します。
構文	<pre>msgproplist_call ::= msgproplist([msg_doc] [returns varchar text]) msg_doc ::= basic_character_expression prop_name ::= basic_character_expression</pre>
パラメータ	<p><code>msgproplist_call</code> <code>msgproplist</code> 関数を使用する要求を作成します。</p> <p><code>msg_doc</code> <code><msgheader></code> または <code><msgproperties></code> XML ドキュメントで、<code>basic_character_expression</code> です。<code>msg_doc</code> が指定されていない場合は、<code>@@msgproperties</code> の現在の値が使用されます。</p> <p><code>prop_name</code> 値またはタイプを抽出するプロパティ名です。<code>basic_character_expression</code> です。</p> <p><code>returns varchar text</code> 返されるメッセージの形式を指定します。</p>
例	<p>この例では、<code>msgrecv</code> からの呼び出しが、“<code>trade_name</code>” という名前の 1 つのプロパティと “<code>Acme Maintenance</code>” (“<code>Quick & Safe</code>”) という値を持つメッセージを返します。この場合、<code>@@msgproperties</code> グローバル変数の値は次のようになります。</p>

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgproperties
  trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
</msgproperties>
```

アンパサンドと `Quick & Safe` という句を囲む引用符は、XML 規約に従って、`"` および `&` という XML エンティティで置き換えられます。

次のいずれかによって、メッセージに属すプロパティのリストが取得されます。

```
select msgproplist
select msgproplist(@@msgproperties)
```

使用法

- msgproplist 呼び出しの結果が 16K を超える場合は、結果値には “TRUNCATED” という語が含まれます。このような場合は、returns text を指定して結果がトランケートされないようにします。他の msgprop 関数を使用して、プロパティ・リストを反復処理し、プロパティの名前と値を取得する必要があります。
- 戻り値の長さを指定しないで msgproplist を実行した場合、デフォルトの戻り値 (32) を超す出力はトランケートされます。この問題を回避するには、戻り値の長さを指定します。たとえば、次の文はトランケートされます。

```
declare @properties varchar(1000)
select @properties = msgproplist(@@msgproperties returns varchar)
```

一方、次の文はトランケートされません。

```
declare @properties varchar (1000)
select @properties= msgproplist(@@msgproperties returns varchar(1000))
```


msgpropname

説明 `<msgheader>` および `<msgproperties>` ドキュメントからプロパティ名を抽出して返します。integer パラメータの値が 1 より小さいか `msg_doc` のプロパティの数より大きい場合は、結果は null 値になります。

構文

```
msgpropname_call ::= msgpropname(integer[ ,msg_doc], )
                    msg_doc ::= basic_character_expression
                    prop_name ::= basic_character_expression
```

パラメータ *integer*
値のインデックスです。

msgpropname_call
`msgpropname` 関数を使用する要求を作成します。

msg_doc
`<msgheader>` または `<msgproperties>` XML ドキュメントで、*basic_character_expression* です。*msg_doc* が指定されていない場合は、`@@msgprproperties` の現在の値が使用されます。

prop_name
値またはタイプを抽出するプロパティ名です。*basic_character_expression* です。

例 **例 1** `msgrecv` からの呼び出しが、`trade_name` という名前の 1 つのプロパティと “Acme Maintenance” (“Quick & Safe”) という値を持つメッセージを返します。この場合、`@@msgproperties` グローバル変数の値は次のようになります。

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgproperties
  trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
</msgproperties>
```

アンパサンドと Quick & Safe という句を囲む引用符は、XML 規約に従って、`"` および `&` という XML エンティティで置き換えられます。

例 2 9 番目のプロパティが存在しないため、null 値を返します。

```
select msgpropname(9, @@msgproperties)
```

msgproptype

説明	<msgheader> および <msgproperties> ドキュメントから、 <i>prop_name</i> と同じ名前を持つ <i>msg_doc</i> プロパティのメッセージ・プロバイダのプロパティ・タイプを抽出して返します。 <i>msg_doc</i> に <i>prop_name</i> と同じ名前のプロパティがない場合、結果は null 値になります。
構文	<pre>msgproptype_call ::= msgproptype(prop_name [, msg_doc] msg_doc ::= basic_character_expression prop_name ::= basic_character_expression</pre>
パラメータ	<p>msgproptype_call msgproptype 関数を使用する要求を作成します。</p> <p>msg_doc <msgheader> または <msgproperties> XML ドキュメントで、<i>basic_character_expression</i> です。<i>msg_doc</i> が指定されていない場合は、@@msgproproperties の現在の値が使用されます。</p> <p>prop_name 値またはタイプを抽出するプロパティ名です。<i>basic_character_expression</i> です。</p>
例	<p>この例では、メッセージは値が 1234 の整数である “integer_prop” および値が “cat” という文字列である “string_prop” の 2 つのプロパティと共に送信されます。</p>

```
select msgsend('msgproptype example',
              'tibco_jms:tcp://localhost:7222?queue=queue.sample'
              MESSAGE PROPERTY "integer_prop=1234,string_prop='cat'")
go
```

```
-----
ID:E4JMS-SERVER.82CC311EC:1
(1 row affected)
```

次に、メッセージが読み込まれます。

```
select msgrecv('tibco_jms:tcp://localhost:7222?queue=queue.sample')
go
```

```
-----
msgproptype example
(1 row affected)
```

@@msgproperties グローバル変数が選択され、受信したメッセージのプロパティが表示されます。

```
select @@msgproperties
go
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
  <msgproperties
    string_prop="'&apos;cat&apos;'"
    ASE_RTMS_CHARSET="1"
```

```

ASE_ORIGIN='&apos;francis_pinot_2&apos;';
ASE_SPID="15"
ASE_MSGBODY_FORMAT='&apos;string&apos;';
ASE_TIMESTAMP='&apos;2005/06/22 15:01:36.91&apos;';
ASE_MSGBODY_SCHEMA='&apos;NULL&apos;';
ASE_RTMS_VERSION='&apos;1.0&apos;';
ASE_VERSION='&apos;12.5.0.0&apos;';
integer_prop="1234">
</msgproperties>

```

(1 row affected)

最初の `msgproptype` 呼び出しは “integer_prop” プロパティのタイプを要求し、“Integer” を返します。

```

1> select msgproptype('integer_prop')
2> go

```

```

-----
Integer
(1 row affected)

```

2 番目の `msgproptype` 呼び出しは “string_prop” プロパティのタイプを要求し、“String” を返します。

```

1> select msgproptype('string_prop')
2> go

```

```

-----
String
(1 row affected)

```

使用法

- (MQ) `msgproptype` を使用して MQ メッセージ・ヘッダに含まれている次のバイナリ・フィールドのいずれかのクエリを実行すると、文字列 “Hex” が返されます。
 - MsgId
 - CorrelId
 - GroupId
 - Encoding

たとえば、次の式は、“Hex” を返します。

```
select msgproptype ('Encoding', @@msgheader)
```

msgpropvalue

説明 `<msgheader>` および `<msgproperties>` ドキュメントから `prop_name` と同じ名前の `msg_doc` プロパティの値を抽出して返します。結果は `varchar` に変換されたプロパティ値で、`msg_doc` に `prop_name` と同じ名前のプロパティがない場合は `null` 値になります。

構文

```
msgpropvalue_call ::= msgpropvalue(prop_name [ , msg_doc ]
                                msg_doc ::= basic_character_expression
                                prop_name ::= basic_character_expression
```

パラメータ

`msgpropvalue_call`
`msgpropvalue` 関数を使用する要求を作成します。

`msg_doc`
`<msgheader>` または `<msgproperties>` XML ドキュメントで、`basic_character_expression` です。`msg_doc` が指定されていない場合は、`@@msgproperties` の現在の値が使用されます。

`prop_name`
値またはタイプを抽出するプロパティ名です。`basic_character_expression` です。

例 **例 1** これらの例では、`msgrecv` からの呼び出しが、“trade_name” という名前の 1 つのプロパティと “Acme Maintenance” (“Quick & Safe”) という値を持つメッセージを返します。この場合、`@@msgproperties` グローバル変数の値は次のようになります。

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
  <msgproperties
    trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
  </msgproperties>
```

アンパサンドと `Quick & Safe` という句を囲む引用符は、XML 規約に従って、`"` および `&` という XML エンティティで置き換えられます。次のコード例では、`trade_name` というメッセージ・プロパティを取得します。

```
select msgpropvalue(@@msgproperties, 'trade_name')
-----
('Quick & Safe') Acme Maintenance
```

これは、Transact-SQL 変数またはカラムに格納された元の文字列です。

例 2 取得したメッセージに “discount” という名前のプロパティがないため、`null` 値を返します。

```
select msgpropvalue('discount', @@msgproperties)
```

例 3 8 番目のプロパティ値を取得します。

```
select msgpropvalue (msgpropname(8, @@msgproperties))
```

msgpublish

説明 JMS のみ トピックにメッセージをパブリッシュする SQL インタフェースを提供します。

構文

```

message_publish_call ::=
    msgpublish(message_body, subscription_name
        [options_and_properties])
    options_and_properties ::=
        [option_clause] [properties_clause]
        [header_clause]
        option_clause ::= [,] option option_string
        header_clause ::= [,] message header
            option_string
        properties_clause ::=
            [,] message property option_string
        message_body ::= scalar_expression |
            (select_for_xml)
  
```

パラメータ

message_body

送信するメッセージです。メッセージ本文には、任意の文字列だけでなく、バイナリ・データ、文字データ、または SQLX データも格納できます。

subscription_name

メッセージのパブリッシュ先のサブスクリプションの名前です。

option_clause

option 名および *option_string* の一般的な形式です。141 ページの項で説明しています。

`msgsend` に対して指定できるオプションは、表 3-7 (83 ページ) に記載されています。

properties_clause

option_string または以下の表にリストされているオプションのいずれかです。表 3-7 および表 3-8 (83 ページ) に記載されたオプションは、表の特性の列に示されるとおり、メッセージ・ヘッダまたはメッセージ・プロパティにプロパティとして設定されます。オプション値がプロパティ値になります。

プロパティ名の太文字と小文字は区別されます。

scalar_expression

メッセージが SQL *scalar_expression* である場合、データ型はどれでもかまいません。

type オプションが指定されない場合、*scalar_expression* が `character` データ型に評価されると、メッセージ・タイプは `text` になります。それ以外の場合には、メッセージ・タイプは `bytes` になります。

scalar_expression のデータ型が `character` でない場合、暗黙的な変換用の通常の SQL 規則を使用して `varbinary` に変換されます。ホスト・マシンのバイト順に従って、データ型のバイナリ値がメッセージに埋め込まれます。

select_for_xml

for xml 句を指定する `select` 式です。

表 3-7: msgpublish option_string パラメータの値

オプション	値	デフォルト	コメント
schema	<ul style="list-style-type: none"> no yes "user_schema" 	no	<p>次のいずれかの値を入力します。</p> <ul style="list-style-type: none"> user_schema – ユーザ指定スキーマです。message_body を記述します。 no – 生成されメッセージの一部として送信されるスキーマがないことを示します。 yes – Adaptive Server がメッセージ用の XML スキーマを生成することを示します。yes が意味を持つのは、パラメータ select_for_xml を使用する message_body 内だけです。select_for_xml は SQL 結果セットの SQLX フォーマット表現を生成します。生成された XML スキーマは、結果セット・ドキュメントを記述する SQLX フォーマット・スキーマです。このスキーマは、ASE_MSGBODY_SCHEMA プロパティとしてメッセージに埋め込まれます。
type	text または bytes	text	送信されるメッセージ・タイプです。

- 表 3-8 に、properties_clause パラメータのオプションと値をリストします。表 3-8 に記載されていないプロパティを使用する場合、送信されるメッセージのメッセージ・プロパティにプロパティとして設定されます。

表 3-8: msgpublish properties_clause パラメータの値

オプション	値	デフォルト	特性	コメント
correlation	string	なし	ヘッダ	複数のメッセージを 1 つにまとめるためにクライアント・アプリケーションが関連 ID を設定します。Adaptive Server はアプリケーションが指定する関連 ID を設定します。
mode	<ul style="list-style-type: none"> persistent non-persistent 	persistent	ヘッダ	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> persistent – メッセージはステープル・ストレージを使用して JMS プロバイダによってバックアップされます。メッセージをコンシュームする前にメッセージング・プロバイダに障害が発生し、mode が persistent に設定されていると、メッセージは高い確率で保存されます。 non-persistent でありメッセージング・プロバイダに障害が発生 – メッセージは送信先に届く前に失われることがあります。
priority	1 ~ 9	4	ヘッダ	<p>priority の動作は、基本となるメッセージ・バスによって制御されます。この値が JMS に適用されます。</p> <p>通常の優先度は 1 ~ 4 です。優先度が特に高いのは 5 ~ 9 です。</p>

オプション	値	デフォルト	特性	コメント
replyqueue	<code>queue_name</code> が含まれる文字列	なし	ヘッダ	<code>queue_name</code> または <code>topic_name</code> の値に応じて、次のようになります。
replytopic	<code>topic_name</code> が含まれる文字列	なし	ヘッダ	<ul style="list-style-type: none"> • <code>syb_temp</code> – Adaptive Server が一時的な送信先を作成し、新規作成した送信先に関連する情報をヘッダ情報の一部として送信します。 <p>次に、Adaptive Server は一時的な送信先として <code>@@msgreplytoinfo</code> を更新します。</p> <p>一時的な送信先の種類がキューなのかトピックなのかは、<code>replyqueue</code> と <code>replytopic</code> のどちらを指定するのかによって決まります。最後に列挙されたオプションだけが使用されます。</p> <ul style="list-style-type: none"> • 既存の送信先 – Adaptive Server は新規送信先を作成せず、ユーザが指定した送信先を使用します。
ttl	0 ~ (2 ⁶³ -1)	0	ヘッダ	<p><code>ttl</code> は、メッセージング・パスでの有効期間を指します。Adaptive Server はこの影響を受けません。</p> <p>有効期間情報とは、メッセージが有効である期間をミリ秒単位で示したものです。たとえば、60 はメッセージの有効期間が 60 ミリ秒であることを示します。</p> <p>値が 0 の場合、メッセージの有効期間が切れないことを意味します。</p> <p><code>ttl</code> は <code>timespec</code> オプションを使用します。「timespec」(143 ページ) を参照してください。</p>

パーミッション

`msgpublish` を実行するには、`messaging_role` が必要です。

msgrecv

説明 さまざまなサービス終了ポイントからメッセージを受け取るための SQL インタフェースを提供します。このサービス終了ポイントは queue 型である必要があります。

`msgrecv` は指定された `service_provider` および `service_destination` からメッセージを受け取り、そのメッセージを返します。返される値は、サービス・プロバイダによって返され、指定された return タイプに変換されたメッセージ本文です。

構文

```
msgrecv_call ::=
    msgrecv (end_point options_filter_and_returns)
            options_filters_and_return ::=
            [option_clause] [filter_clause] [returns_clause]
            option_clause ::= [,] option option_string
            filter_clause ::= [,] message selector message_filter
            message_filter ::= basic_character_expression
            returns_clause ::= [,] returns sql_type
            end_point ::= basic_character_expression
            sql_type ::=
                varchar(integer) | java.lang.String | text
                | varbinary(integer) | image
            message_filter ::= basic_character_expression
```

パラメータ

basic_character_expression

データ型が `char`、`varchar`、または `java.lang.String` の SQL クエリ式です。

end_point

`basic_character_expression` であり、ランタイム値は `service_provider_uri` です。`end_point` はメッセージの送信先です。

filter_clause

`message_filter` を指定されたメッセージ・プロバイダに直接渡します。これにより、使用法が決まります。

message_filter

フィルタ・パラメータであり `basic_character_expression` です。フィルタ値はメッセージ・プロバイダに直接渡されます。使用法はメッセージ・プロバイダによって異なります。メッセージ・フィルタの説明については、以下の「使用法」の項を参照してください。

プロバイダ・クラスが `“ibm_mq”` である場合、`msgrecv` に指定される `message_filter` は無視されます。

msgrecv

指定された `service_provider` および `service_destination` からメッセージを受け取り、そのメッセージを返します。返される値は、サービス・プロバイダによって返され、指定された return タイプに変換されたメッセージ本文です。

option

MQ の場合は表 3-9 (89 ページ) に示す値、JMS の場合は表 3-10 (95 ページ) に示す値です。

注意 認識できないオプション名があると、エラーになります。

option_string

141 ページで説明されている **option_string** の一般的なフォーマットです。**msgrecv** のオプションについては、MQ の場合は表 3-9 (89 ページ)、JMS の場合は表 3-10 (95 ページ) で説明します。

returns_clause

返されるデータ型です。**returns_clause** を省略した場合のデフォルトは、**varchar(16384)** です。

varbinary タイプまたは **image** タイプの **returns_clause** を指定した場合、データはメッセージのバイト順で返されます。

sql_type

以下の有効な SQL データ型のいずれかです。

- **varchar(n)**
- **text**
- **java.lang.String**
- **varbinary(n)**
- **image**
- **univarchar(n)**

例 **例 1** (MQ) メッセージをタイムアウトが指定されたキュー Q1 から読み込みます。3 秒のタイムアウトまでに使用できるメッセージが Q1 にない場合は、null 値が返されます。

```
select msgrecv(  
    'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
    option 'timeout=3ss')
```

例 2 (MQ) **correlationId** をタイムアウトなしで指定します。**correlationId** と一致するメッセージをキューで使用できるときに呼び出しが返されます。

```
select msgrecv(  
    'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
    option 'correlationId=x67a12z99')
```

例 3 (MQ) `groupId` と `allMsgsInGroup` は指定されていますが、タイムアウトは指定されていません。この呼び出しは、指定された `groupId` のすべてのメッセージがキューで使用できるようになるまで、ブロックされます。

```
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'groupId=g7853b77,allMsgsInGroup=yes')
```

例 4 (MQ) 以下のメッセージがすでにキューにあるとします。

```
AA BB CC DD EE FF GG HH
```

最初の 3 つのメッセージ (AA ~ CC) がブラウザ・モードで読み込まれ、CC が削除されます。次に、ブラウザ・カーソルが先頭に戻り、3 つのメッセージ (AA ~ DD) がブラウザ・モードで読み込まれ、DD が削除されます。最後に、次に設定された位置で読み込みが実行され、AA が読み込まれて削除されます。この例が完了したとき、キューにはメッセージ AA、CC、および DD はありません。

```
-- Browse cursor at the beginning, this will return 'AA'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,browse=first')

-- Browse the next message, this will return 'BB'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,browse=next')

-- Browse the next message, this will return 'CC'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,browse=next')

-- Remove the message under the browse cursor, this will return 'CC'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,position=cursor')

-- Reposition browse cursor at the beginning, this will return 'AA'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,browse=first')

-- Browse the next message, this will return 'BB'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,browse=next')

-- Browse the next message, this will return 'DD'
select msgrecv(
```

```

'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
option 'inputMode=browse+Qdefault,browse=next')

-- Read the message under the cursor, this will return 'DD'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,position=cursor')

-- Read the next message in queue order, this will return 'AA'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,position=next')

```

例 5 Tibco JMS – 指定された *end_point* からメッセージを受け取ります。

```

select msgrecv
  ('tibco_jms:tcp://my_jms_host:7222?queue=queue.sample,'
  +'user=jms_user1,password=jms_user1_password')

```

例 6 SonicMQ JMS – *timeout* オプションを使用して、指定された *end_point* のキュー Q1 からメッセージを受け取ります。

```

select msgrecv
  ('sonicmq_jms:tcp://mysonic:7223?queue=Q1,user=sonic_usr,
  password=sonic_pwd',option 'timeout=1000')

```

例 7 (JMS) timeout オプションを使用し、メッセージ・セレクタを指定して、指定された *end_point* からメッセージを受け取ります。

```

declare @mymsg varchar (16384)
select @mymsg = msgrecv('my_jms_provider?queue=queue.sample',
  option 'timeout=1000'
  message selector 'correlationID = 'MSG_001''')

```

例 8 (JMS) この *msgrecv* 呼び出しは、メッセージ・プロパティ “Name” が “John Smith” である場合、*queue.sample* からのメッセージのみコンシュームします。

```

select msgrecv('my_jms_provider?queue=queue.sample',
  message selector 'Name='John Smith''')

```

例 9 (JMS) text メッセージをテーブルに挿入する方法を示します。

```

create table T1(c1 numeric(5,0)identity, m text)
insert into T1
select msgrecv('my_jms_provider?queue=queue.sample',
  returns text)

```

例 10 (JMS) この例では、メッセージを読み込み、*varbinary* として返します。

```

select msgrecv('my_jms_provider?queue=queue.sample'
  returns varbinary(500))

```

使用法

(MQ) 表 3-9 は、msgrecv プロパティで使用できる *option* の値および *option_string* の値を示します。

表 3-9: (MQ) msgrecv の option および option_string の値

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
allMsgsInGroup	<ul style="list-style-type: none"> • yes • no 	no	<p>groupid を指定しないかぎり、このオプションは無視されます。</p> <p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – グループの最初のメッセージが返される前に、グループのすべての論理メッセージがキューに存在する必要があります。 • no – グループの最初のメッセージが返される前に、グループのすべての論理メッセージがキューに存在する必要はありません。
allSegments	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – 最初のメッセージ・セグメントが返される前に、セグメント化されたメッセージのすべてのメッセージがキューに存在する必要があります。 • no – 最初のメッセージ・セグメントが返される前に、セグメント化されたメッセージのすべてのメッセージが存在する必要はありません。
browse	<ul style="list-style-type: none"> • next • next+Lock • first • first+Lock • cursor • cursor+Lock • reopen • reopen+Lock • unlock • null 	null	<p>browse プロパティに設定する値は次のとおりです。</p> <ul style="list-style-type: none"> • null – メッセージが読み込まれ、キューから削除されません。 position オプションは、読み込むメッセージを制御します。 • null 以外 – メッセージは読み込まれるが、キューから削除されません。順序はキューのデフォルトの順序 (先入れ先出し、または優先度) によって決まります。 <p>次の値を指定することもできます。</p> <ul style="list-style-type: none"> • msgId, correlationId, groupId, sequenceId, または offset を指定 – MQ は、指定した選択基準に合致する次のメッセージをブラウズし読み込みます。 • timeout を指定し、選択基準に合致するメッセージが見つからない – null 値が返されます。 • timeout を指定しない – 選択基準に合致するキューにメッセージが表示されるまで、msgrecv 操作はブロックされます。

option の値	option_string の値	デフォルト	説明
browse (続き)			<p>browse には、次の値を指定できます。</p> <ul style="list-style-type: none"> • next – 次のメッセージを返します。 • next+Lock – メッセージを返し、他のリーダがそのメッセージを削除しないようにロックします。 • first – 最初のメッセージを返します。 browse=next オプションを発行後 browse=first を指定した場合、ブラウザ・カーソルはキューをオープンした開始位置に再度配置されます。 • first+Lock – 最初のメッセージを返し、他のリーダがそのメッセージを削除しないようにロックします。 • cursor – ブラウズ・カーソルのあるメッセージを返します。 browse=first、browse=first+Lock、browse=next、または browse=next+Lock を先に実行しないと、browse=cursor を使用できません。 browse=cursor を繰り返すと同じメッセージが返されます。 • cursor+Lock – カーソルのあるメッセージを返し、他のリーダがそのメッセージを削除しないようにロックします。 • reopen – ブラウズ・カーソルを閉じ、再オープンし、先頭に配置します。優先度キューの場合、前回のオープン以降に優先度が高いメッセージが入った場合、そのメッセージはキューの先頭に表示されます。 • reopen+Lock – ブラウズ・カーソルを閉じ、再オープンし、先頭に配置し、他のリーダが最初のメッセージを削除しないようにロックします。 • unlock – カーソル下のメッセージのロックを解除して返します。

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
bufferLength	sizespec 0、または 1 ~ 値		<p>bufferLength – サイズを指定したバッファを使用してメッセージを返します。</p> <ul style="list-style-type: none"> メッセージの組み込み関数はこの長さのバッファを割り付けようとしています。バッファを割り付ける十分なメモリがないと、コマンドは失敗します。 text または image を返すように msgrecv を指定する場合、msgrecv ではメッセージ・サイズは指定されたキューが対応できる最大のメッセージであると仮定し、maxMsgLength キューのプロパティを使用します。maxMsgLength を次の値に設定した場合は、メッセージ用のメモリを増やします。 <ul style="list-style-type: none"> デフォルト値の 4MB メッセージの実際の長さよりも大きい値 <p>Adaptive Server がメッセージを読み込むのに使用するメモリを最小限にするように、maxMsgLength キューのプロパティをアプリケーションで許容される最小値に設定することをおすすめします。maxMsgLength を設定するには、MQ コマンド (MQSC) ツールを使用してキューの MAXMSGL 属性を変更します。</p> <p>デフォルト. bufferLength のデフォルト値は次のいずれかです。</p> <ul style="list-style-type: none"> キュー・マネージャおよびターゲット・キューに定義された maxMsgLength の最小値。 戻り型が text、image、または java.lang.String でない場合は、戻り型の長さ。 <p>0 はデフォルト値を使用することを示します。</p> <p>pub/sub メッセージの場合、bufferLength には MQRF ヘッダなどのメッセージ・トピックの長さが含まれている必要があります。</p>
closeAfterRecv	<ul style="list-style-type: none"> yes no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> yes – 現在の msgrecv 操作後、キューを閉じ、それ以降の msgrecv 呼び出しでは別の入力モードでキューを再オープンできるようにします。 no – 現在の msgrecv 操作後、キューをオープンしたままにします。
completeMsg	<ul style="list-style-type: none"> yes no 	yes	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> yes – セグメント化されたメッセージが1つのメッセージとして返されます。 no – セグメント化されたメッセージがある場合、各セグメントが個別のメッセージとして返されます。 <p>completeMsg は、同じ終了ポイントの msgrecv のすべての呼び出しと同じ設定である必要があります。</p>

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
correlationId	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>読み込むメッセージの相関 ID で、キューの特定のメッセージを選択するために select 文で使用されます。</p> <p>MQ では、このフィールドを binary 値をサポートできる “unsigned char” と定義しています。binary 文字列を correlationId として入力するには、値として “0x...” を使用します。値を引用符で囲まないでください。</p>
formatName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>予期されるメッセージのフォーマットの名前です。この値が指定されていて、メッセージの名前 formatName フィールドが一致しない場合、メッセージは読み込まれません。詳細については、この表の requeue オプションを参照してください。</p> <p>MQ では、この文字列は 8 バイトに制限されます。</p>
groupid	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>読み込むメッセージのグループ ID です。これは select オプションです。MQ ではこのフィールドを “unsigned char” と定義しています。つまり、binary 値をサポートできます。binary 文字列を msgId として入力するには、値として “0x...” を使用します。値を引用符で囲まないでください。</p>

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
inputMode	<ul style="list-style-type: none"> • browse • Qdefault • shared • exclusive • browse+Qdefault • browse+shared • browse+exclusive 	Qdefault	<p>inputMode の値は、次のように MQ キューをオープンします。</p> <ul style="list-style-type: none"> • browse – キューが開かれるのは参照目的のみです。破壊的読み取りを試みると、キュー・マネージャがエラーを生成します。 • Qdefault – キューは定義済みのデフォルトの入力モードで開かれます。 • shared – キューは共有入力モードでオープンされます。キューが他の MQ ハンドルによって排他モードですでにオープンされている場合、エラーが表示されます。 • exclusive – キューは排他的入力モードでオープンされず。キューが他の MQ ハンドルによって共有モードまたは排他モードですでにオープンされている場合、エラーが表示されます。 • browse+Qdefault – キューはブラウズ入力モードおよび共有入力モードでオープンされます。 • browse+shared – キューはブラウズ入力モードおよび共有入力モードでオープンされます。キューが他の MQ ハンドルによって排他モードですでにオープンされている場合、エラーが表示されます。 • browse+exclusive – キューはブラウズ入力モードおよび排他的入力モードでオープンされます。キューが他の MQ ハンドルによって共有モードまたは排他モードですでにオープンされている場合、エラーが表示されます。 <p>inputMode は msgrecv のみで有効です。 任意の終了ポイントに対し、以下のいずれかのタイミングで inputMode を指定する必要があります。</p> <ul style="list-style-type: none"> • 最初の msgrecv 操作時 • closeAfterRecv の指定後 <p>呼び出し間で inputMode の値を変更しようとする、予期しない結果を生じる可能性があります。</p>
msgld	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>読み込むメッセージのメッセージ ID です。 選択オプションとして、キューの特定のメッセージの選択に msgld を使用できます。</p> <p>MQ では、このフィールドを binary 値をサポートできる “BYTE array” と定義しています。binary 文字列を msgld として入力するには、値として “0x...” を使用します。値を引用符で囲まないでください。囲むと、引用符付き文字列として解釈されてしまいます。</p>
offset	-1、0 ~ <i>maxint</i> の <i>integer</i>		<p>読み込むメッセージのオフセットです。 -1 の場合、オフセットは指定されません。</p> <p>select オプションとして、キューの特定のメッセージの選択に offset を使用できます。</p>

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
ordering	<ul style="list-style-type: none"> logical physical 	physical	<p>ordering には次の値を指定できます。</p> <ul style="list-style-type: none"> logical – メッセージは、<code>groupId</code>、<code>sequenceId</code>、および <code>offsets</code> に従って論理順に読み込まれます。 physical – メッセージは、キューに表示される順序で読み込まれます。
position	<ul style="list-style-type: none"> next cursor 	next	<p>position は、返されるメッセージを制御します。指定する <code>inputMode</code> 値によって、1 つまたは 2 つの「読み込み」位置があります。</p> <ul style="list-style-type: none"> “Normal” – 破壊的読み取りが通常発生する、デフォルトの読み込み位置です。キューをオープンすると、「通常」の読み込み位置はキューの最初のメッセージに配置されます。 “Browse cursor” – 参照が指定されたこれまでの呼び出しによって読み込み位置が決められた場所です。キューをオープンして参照するとき、「ブラウズ・カーソル」はキューの最初のメッセージの前に配置されます。「ブラウズ・カーソル」は <code>browse+Qdefault</code>、<code>browse+shared</code>、<code>browse+exclusive</code> のみに使用されます。指定する値は次のとおりです。 next – 「通常の」読み込み位置の現在のメッセージを返します。メッセージが返されると、「通常の」読み込み位置はメッセージの前へ移動します。 cursor – 「ブラウズ・カーソル」がある現在のメッセージを返します。「ブラウズ・カーソル」がまだ配置されていない場合は、MQ キュー・マネージャがエラーを返します。メッセージが返されると、「ブラウズ・カーソル」はメッセージの前へ移動します。 <p>MQ クエリ・マネージャには、返すメッセージを決める前に次の内容が適用されます。</p> <ul style="list-style-type: none"> キューのデフォルトの順序 (優先度、先入れ先出し) 指定された選択基準 (<code>messageId</code>、<code>correlationId</code>、<code>groupId</code>、<code>sequenceId</code>、または <code>offset</code>)
requeue	<ul style="list-style-type: none"> null string 	null	<p>これは、終了ポイントの完全 URI である必要があります。次の場合、読み込まれたメッセージは指定されたキューに再キューイングされます。</p> <ul style="list-style-type: none"> <code>formatName</code> が指定されているとき、msgrecv がメッセージを読み込む場合。 読み込まれたメッセージに異なる <code>formatName</code> がある場合。 requeue が null でない場合。 <p>指定されたキューにメッセージを再キューイングできない場合、メッセージは読み込まれたキューに残され、例外が発生します。</p> <p>MQ では、requeue は 48 バイトに制限されます。</p>

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
sequenceId	-1 ~ 9,999,999 の <i>integer</i>	-1	読み込むメッセージのシーケンス ID です。 -1 の場合、シーケンス ID は指定されません。 選択オプションとして、キューの特定のメッセージの選択に <code>sequenceId</code> を使用できます。
truncationAllowed	<ul style="list-style-type: none"> • yes • no 	no	次の場合、メッセージをトランケートできます。 <ul style="list-style-type: none"> • メッセージの読み込みに使用されるバッファ (<code>bufferLength</code>、つまり返されるデータ型の長さ) の場合 • バッファがメッセージの長さよりも小さい場合 指定する値は次のとおりです。 <ul style="list-style-type: none"> • yes - トランケートできます。 • no - トランケートできません。値が <code>no</code> でメッセージがトランケートされた場合、読み込みは失敗します。
timeout	-1, 0 ~ (2 ³² - 1) の <i>timespec</i>	-1	タイムアウトを指定します。指定する値は次のとおりです。 <ul style="list-style-type: none"> • -1 - タイムアウトはありません。 • <code>timeout</code> を整数で指定 - ミリ秒単位の値を指定します。詳細については、「timespec」(143 ページ) を参照してください。

(JMS) 表 3-10 は、`msgrecv` プロパティで使用できる *option* の値および *option_string* の値を示します。

表 3-10: (JMS) `msgrecv` の *option* および *option_string* の値

<i>option</i> の値	<i>option_string</i> の値	デフォルト	説明
requeue	<i>string</i>	なし	Adaptive Server が処理できないメッセージを再キューイングする場所、キュー、またはトピックの名前です。 <code>requeue</code> を指定せず、メッセージを処理できない場合は、エラー・メッセージが表示されます。指定された終了ポイントが <code>msgconsume</code> および <code>msgrecv</code> と同じメッセージング・プロバイダに存在する必要があります。
timeout	<i>timespec</i> -1, 0 - (2 ³¹ - 1)	-1	デフォルトでは、 <code>msgrecv</code> はメッセージ・バスから次のメッセージを読み込むまで、メッセージをブロックします。 <code>timeout</code> が -1 でない場合は、 <code>timeout</code> 間隔が経過してもメッセージが読み込まれないときに <code>msgrecv</code> は <code>null</code> を返します。値はミリ秒単位の数値です。「 timespec 」(143 ページ) を参照してください。

- 認識できないオプション名があると、エラーになります。
- メッセージ・ヘッダから読み込まれるプロパティについては、「[@@msgheader](#)」(40 ページ) を参照してください。
- `msgrecv` は、指定された *service_provider* および *service_definition* からメッセージを受け取り、そのメッセージを返します。

- デフォルトでは、`msgrecv` はメッセージ・バスから次のメッセージを読み込むまでメッセージをブロックする、ブロックング・コマンドです。`timeout` が -1 でない場合は、`timeout` 間隔が経過してもメッセージが読み込まれないときに `msgrecv` は `null` を返します。値はミリ秒単位の数値です。
- Adaptive Server が処理するメッセージ・タイプは、`message`、`text`、または `bytes` のみです。処理できないメッセージがあり、`requeue` が指定されていない場合は、そのメッセージは元のキューに残されます。これ以降、読み込みを続けて同じメッセージがあった場合、同じ結果になります。これを回避するには、`requeue` を指定します。`requeue` を使用すると、Adaptive Server が処理できないメッセージは指定されたキューに配置されます。
指定された終了ポイントは、`msgrecv` で使用された終了ポイントと同じメッセージング・サービス・プロバイダに存在する必要があります。
- ホスト・マシンのバイト順に従って、データ型のバイナリ値がメッセージに埋め込まれます。
- `msgrecv` を呼び出すと、次のような結果になります。
 - 返される値は、メッセージ・プロバイダによって返され、指定された `returns` タイプに変換された `message_body` 値です。
 - `@@msgheader` および `@@msgproperties` の値は `<msgheader>` および `<msgproperties>` ドキュメントの値に設定されます。これらには、`msgrecv` によって返されるメッセージのプロパティが含まれています。
 - `msgpropvalue` を使用して、`<msgheader>` および `<msgproperties>` ドキュメントから特定のプロパティの値を抽出できます。[「msgpropvalue」 \(80 ページ\)](#) を参照してください。
 - `<msgheader>` および `<msgproperties>` の一般的なフォーマットについては、[「メッセージ関連のグローバル変数」 \(40 ページ\)](#) を参照してください。

MQ と msgrecv

これらの文は、プロバイダ・クラスが `"ibm_mq"` の場合のみ有効です。

- `msgId`、`correlationId`、`groupId`、`sequenceId`、および `offset` オプションは、メッセージ選択の一致基準として機能します。指定した場合、指定された値と一致する次のメッセージが返されます。WebSphere MQ キュー・マネージャが確認します。

- 受け取ったメッセージの `MQMD.Format` フィールドが “MQSTR” である場合、データは文字データとみなされ、`text` または `varchar` として返されます。他のフォーマット名は、`image` または `binary` としてのみ返されます。特殊なケースとして、`MQMD.Format` が “MQHRF” である場合があります。この場合は、`MQRFH.Format` フィールドを代用します。指定された戻り型でメッセージ本文を返すことができない場合は、`requeue` オプションが指定されていればメッセージは `requeue` オプションに送信されます。それ以外の場合は読み込み操作は失敗します。MQ では、`MQMD.Format` が “MQSTR” であるとき、メッセージ本文には文字データしか含まれないということは強制されません。プログラマは `image` または `varbinary` 戻り型を必ず指定する必要があります。

プロパティまたはオプションの値の引用

- `option` 値を文字列として扱うには、アポストロフィ (') で囲みます。アポストロフィを省略すると、`option` 値は他のプロパティ名として扱われ、2 つのプロパティが同じ値である場合のみ、式は `true` になります。

アプリケーションで引用符付き識別子を使用する場合は、メッセージ・セクタをアポストロフィ (') で囲む必要があります。つまり、セクタ内に文字列値がある場合は、これらの値を 2 つのアポストロフィ (") で囲む必要があります。次に例を示します。

```
set quoted_identifier on
select msgrecv ('my_jms_provider?queue=queue.sample',
  message selector 'color = 'red''')
```

アプリケーションで引用符付き識別子を使用しない場合は、メッセージ・セクタを通常の二重引用符で囲むことができます。次に例を示します。

```
set quoted_identifier off
select msgrecv('my_jms_provider?queue=queue.sample',
  message selector "color='red'")
```

次の例では、**messaging client** アプリケーションは、“red” という値を持つ “color” という名前のプロパティおよび “color” という値を持つ “red” という名前のプロパティを表すメッセージを送信します。

```
select msgsend ('Sending message with property color',
  'my_jms_provider?queue=queue.sample'
  message selector 'color=red, red=color')
```

“red” という値を持つ “color” という名前のプロパティが含まれているメッセージのみをコンシュームするクライアント アプリケーションでは、セクタ値を 2 つのアポストロフィ (") で囲む必要があります。次に例を示します。

```
select msgrecv('my_jms_provider?queue=queue.sample'
  message selector 'color='red''')
```

ただし、クライアント・アプリケーションが次の構文を使用した場合は、“red”はプロパティ名として扱われるため、メッセージを受け取りません。

```
select msgrecv('my_jms_provider?queue=queue.sample',
  message selector 'color=red')
```

次の例では、クライアントは複数のプロパティを選択しフィルタ処理するメッセージを送信します。

```
select msgsend('Sending message with properties',
  'my_jms_provider?queue=queue.sample',
  message selector 'color=red, shape=square')
```

別のクライアントが、プロパティ“color”が“red”でプロパティ“shape”が“square”であるメッセージを選択するには、次のコマンドを実行する必要があります。

```
select msgrecv('my_jms_provider?queue=queue.sample',
  message selector 'color='red' and shape='square''')
```

メッセージ・フィルタ

- フィルタ・パラメータを指定する場合、フィルタ値はメッセージ・プロバイダに直接渡されます。使用法はメッセージ・プロバイダによって異なります。
- メッセージ・フィルタに指定された比較はメッセージ・プロバイダが指定したソート順を使用します。これは、Adaptive Server が使用するソート順とは異なる場合があります。
- JMS メッセージ・プロバイダは JMS メッセージ・セレクタをフィルタとして使用します。JMS メッセージ・セレクタの規則は以下のとおりです。
 - メッセージ・セレクタの構文は、not、and、or、between、like などの条件式のサブセットです。
 - 識別子は大文字と小文字を区別します。
 - 識別子はメッセージ・ヘッダ・フィールドとプロパティ名を指定する必要があります。
- JMS のみ – *message_filter* が msgrecv に指定されている場合は無視されます。
- MQ のみ – メッセージ・オプションで相関およびメッセージ ID を指定して、特定のメッセージを選択できます。

パーミッション

msgrecv を実行するには、messaging_role が必要です。

msgsend

説明 queue 型のさまざまなサービス終了ポイントにメッセージを送信するための SQL インタフェースを提供します。

構文

```
message_send_call ::=
    msgsend(message_body, end_point [options_and_properties])
    options_and_properties ::= [option_clause]
    [properties_clause] [header_clause]
    option_clause ::= [,] option option_string
    properties_clause ::= [,] message property
    property_option_string
    header_clause ::= [,] message header
    header_option_string
    message_body ::= scalar_expression |
    (select_for_xml)
    end_point ::= basic_character_expression
```

パラメータ **message_body**
送信するメッセージです。メッセージ本文には、任意の文字列だけでなく、バイナリ・データ、文字データ、または SQLX データも格納できます。

endpoint
メッセージの追加先となるキューです。**endpoint** は **basic_character_expression** であり、ランタイム値は **service_provider_uri** です。

option
msgsend のオプションを指定できるようにします。JMS を使用している場合には、[表 3-11 \(108 ページ\)](#) に記載されたオプションを使用してください。MQ を使用している場合には、[表 3-12 \(108 ページ\)](#) に記載されたオプションを使用してください。

option_string
option_string の一般的な構文と処理を指定します。個々のオプションについては、そのオプションを参照する関数で説明しています。

```
option_string ::= basic_character_expression
option_string_value ::= option_and_value [,] option_and_value]
option_and_value ::= option_name = option_value
option_name ::= simple_identifier
option_value ::= simple_identifier
| quoted_string | integer_literal | float_literal | byte_literal
| true | false | null
```

パラメータ	説明
option_string	指定するオプションについて説明する文字列
simple_identifier	option の値を特定する文字列
quoted_string	埋め込み引用符に関する通常の SQL 表記規則を使用した文字列
integer_literal	通常の SQL 表記規則によって指定されたリテラル
float_literal	通常の SQL 表記規則によって指定されたリテラル
true	ブール・リテラル

パラメータ	説明
false	ブール・リテラル
null	null リテラル
byte_literal	0xHH 形式 (ただし各 H は 16 進数)

properties_clause

property_option_string か、MQ 用の表 3-13 (111 ページ) または JMS 用の表 3-14 (120 ページ) に記載されたオプションのいずれか。この 2 つの表に記載されたオプションは、表の特性の列に示されるとおり、メッセージ・ヘッダまたはメッセージ・プロパティにプロパティとして設定されます。オプション値がプロパティ値になります。

プロパティ名の太文字と小文字は区別されます。

Tibco JMS のみ – 表 3-14 (120 ページ) に記載されていないプロパティを使用する場合、送信されるメッセージのメッセージ・プロパティにプロパティとして設定されます。

MQ のみ – *properties_clause* の値は、rhfCommand オプションで指定される内容に応じて異なります。

- 表 3-15 (121 ページ) に記載されたプロパティが有効になるのは、rhfCommand が deletePublication である場合だけである。

パブリケーション・ストリームに deletePublication コマンド・メッセージが送信されると、MQ pub/sub ブローカはパブリケーション・ストリーム内の指定トピック用に保管されたパブリケーションのコピーを削除します。

msgsend に渡される message_body 引数は、無視されます。

- 表 3-16 (122 ページ) に記載されたプロパティが有効になるのは、rhfCommand が deregisterPublisher である場合だけである。
- MQ pub/sub ブローカのコントロール・キューに送信される deregisterPublisher コマンド・メッセージは、同ブローカにパブリッシャが指定トピックに関してそれ以降はパブリッシュしないことを伝える。
- 表 3-17 (123 ページ) に記載されたプロパティが有効になるのは、rhfCommand が deregisterSubscriber である場合だけである。

msgsend に渡される message_body 引数は、無視されます。

msgType が request である場合、応答メッセージが replyToQmgr と replyToQueue に送信されます。

- 表 3-18 (124 ページ) に記載されたプロパティが有効になるのは、`rnfCommand` が `publish` である場合だけである。

`publish` コマンド・メッセージがパブリケーション・ストリーム・キューに送信され、特定のトピックに関する情報がパブリッシュされます。パブリケーション・データが、`msgsend` に渡される `message_body` 引数として指定されます。

`msgType` が `request` である場合、応答メッセージが `replyToQmgr` と `replyToQueue` に送信されます。
- 表 3-20 (128 ページ) に記載されたプロパティが有効になるのは、`rnfCommand` が `registerSubscriber` である場合だけである。

MQ pub/sub ブローカのコントロール・キューに送信される `registerSubscriber` コマンド・メッセージは、同ブローカにパブリッシャが 1 つ以上の指定トピックに関するデータをパブリッシュしているかまたはパブリッシュできることを伝えます。パブリッシャが登録済みであり、かつ他にエラーがない場合、パブリッシャの登録はそれに合わせて変更されます。

`msgType` が `request` である場合、応答メッセージが `replyToQmgr` と `replyToQueue` に送信されます。
- 表 3-21 (131 ページ) に記載されたプロパティが有効になるのは、`rnfCommand` が `requestUpdate` である場合だけである。

MQ pub/sub ブローカのコントロール・キューに送信される `requestUpdate` コマンド・メッセージは、指定トピックに一致する保管されたパブリケーションすべてを転送するようサブスクライバが同ブローカに望んでいることを伝えます。

`msgType` が `request` である場合、応答メッセージが `replyToQmgr` と `replyToQueue` に送信されます。

scalar_expression

メッセージが SQL *scalar_expression* である場合、データ型はどれでもかまいません。

`type` オプションが指定されない場合、*scalar_expression* が `character` データ型に評価されると、メッセージ・タイプは `text` になります。それ以外の場合には、メッセージ・タイプは `bytes` になります。

scalar_expression のデータ型が `character` でない場合、暗黙的な変換用の通常の SQL 規則を使用して `varbinary` に変換されます。ホスト・マシンのバイト順に従って、データ型のバイナリ値がメッセージに埋め込まれます。

basic_character_expression

データ型が `char`、`varchar`、または `java.lang.String` の Transact-SQL クエリ式です。

(select_for_xml)

for xml 句を指定する select 式です。

select_for_xml パラメータである **message_body** で、**select_for_xml** は SQL 結果セットの SQLX フォーマット表現を生成します。

select_for_xml を指定できるのは、Adaptive Server がネイティブ XML 機能向けに設定されている場合だけです。**select_for_xml** は、msgsend 呼び出しのスカラ式としてだけ参照できます。

select_for_xml をカッコで囲む必要があります。

header_clause

MQ 用の表 3-13 (111 ページ) と JMS 用の表 3-14 (120 ページ) で指定されたヘッダ・プロパティだけを指定できるようにします。認識できないヘッダ・プロパティを入力しようとすると、エラー・メッセージが表示されます。

認識されるヘッダ・プロパティが **message property** 句と **message header** 句の両方に指定された場合、**message header** 句に指定されたヘッダ・プロパティが優先されます。

認識されない名前が **message header** パラメータに指定されると、エラー・メッセージが表示されます。

例 **例 1 SonicMQ JMS** — 指定された終了ポイントにメッセージ “hello” を送信します。

```
select msgsend('hello',
  'sonicmq_jms:tcp://mysonic:7223?queue=testq,user=xyz')
```

例 2 (JMS) 指定された終了ポイントにメッセージ “Hello Messaging World!” を送信します。

```
declare @mymsg varchar (255)
set @mymsg = 'Hello Messaging World!'
select msgsend(@mymsg,
  +'my_jms_provider?queue=queue.sample,user=jms_user1,'
  +'password=jms_user1_password')
```

例 3 Tibco JMS — 本文が SQL 結果セットの SQLX フォーマット表現であるメッセージを送信します。この結果は、指定された終了ポイントに対する SQL クエリによって返されたものです。

```
select msgsend ((select * from pubs2..publishers FOR XML),
  'tibco_jms:tcp://my_jms_host:7222?queue=queue.sample,'
  +'user=jms_user1,password=jms_user1_password')
```

例 4 (JMS) 2つのプロパティを設定し、メッセージのXMLスキーマを生成します。

```
select msgsend
((select pub_name from pubs2..publishers where pub_id = '1389' FOR XML),
 my_jms_provider?queue=queue.sample',
 message property 'priority=6, correlationID=MSG_001',
 option 'schema=yes')
```

例 5 (JMS) メッセージ・プロパティのユーザ指定値を表示します。

```
select msgsend ('hello', 'my_jms_provider?queue=queue.sample'
 message property 'ttl=30,category=5, rate=0.57, rank='top',
 priority=6')
```

ttl と **priority** は、ヘッダ・プロパティとして内部的に設定されます。**category**、**rate**、**rank** は、ユーザ指定メッセージ・プロパティとして設定されます。

例 6 (MQ) 要求メッセージを送信します。応答は同じキュー・マネージャの指定キューに入られます。

```
select msgsend('do something',
 'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
 option 'msgType=request'
 message property 'replyToQueue=QUEUE.REPLY')
```

例 7 (MQ) 応答メッセージを送信します。相関IDと応答キューが、以前に受信した要求メッセージから既に抽出されています。

```
select @correlationId = msgpropvalue("CorrelId", @@msgheader)
select @replyQ = @@msgreplytoinfo
select msgsend('i'm done', @replyQ
 option 'msgType=report'
 message property 'correlationId=' + @correlationId)
```

例 8 (MQ) レポート・メッセージを送信します。相関ID、応答キュー、レポート・メッセージのデータ・ヘッダが、以前に受信した要求メッセージから既に抽出されています。

```
select @correlationId = msgpropvalue("CorrelId", @@msgheader)
select @replyQ = @@msgreplytoinfo
select msgsend(@reportData, @replyQ
 option 'msgType=report'
 message property 'correlationId=' + @correlationId)
```

例 9 (MQ) 4 つのデータグラム・メッセージを送信します。各メッセージは、“theGroup” という名前のグループの一部です。各メッセージのシーケンス番号は順に増えていきます。

```
begin tran
select msgsend('message 1',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=1')
select msgsend('message 2',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=2')
select msgsend('message 3',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=3')
select msgsend('message 4',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=4,lastMsgInGroup=yes')
commit
```

例 10 (MQ) データグラム・メッセージを送信します。さまざまな確認レポートが要求され、“myReplyQueue” に送信されます。

```
select msgsend('I want a confirmation',
  'ibm_mq:channel1/TCP/host1(5678)?queue=QUEUE.COMMAND',
  message property 'replyToQueue=myReplyQueue'
  + ',exceptionReport=yes,'
  + ',arrivalReport=withData'
  + ',deliveryReport=withFullData')
```

例 11 (MQ) トピック “A”、“A/B”、“A/B/C” に関するデータグラム・メッセージをパブリッシュします。トピック “A”、“A/B”、“A/B/C” に関してパブリッシュするためにパブリッシャが登録され、パブリケーションにトピック “A/B” についての情報が記載されます。デフォルトの MQ pub/sub ブローカのキューとストリーム・キューが使用されます。

```
-- First register the publisher
select msgsend(null,
  'ibm_mq:channel1/TCP/host1(5678)?queue=SYSTEM.BROKER.CONTROL.QUEUE'
  option 'msgType=datagram,rfhCommand=registerPublisher'
  message property 'topics='a:A/B:a/b/c''')

-- Now publish the publication
select msgsend('something about A/B',
  'ibm_mq:channel1/TCP/host1(5678)?queue=SYSTEM.BROKER.DEFAULT.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=A/B')
```

例 12 (MQ) 複数のメッセージを1つにまとめて送信します。ordering が logical に設定されているので、*msgInGroup*、*lastMsgInGroup*、*msgSegment*、*msgLastSegment* の各オプションだけを指定してください。グループ名が指定されていないので、キュー・マネージャが選択します。

```
begin tran
select msgsend('first logical message of the group',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes')

select msgsend('second logical message of the group',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes')

select msgsend('third logical message of the group, first segment',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes,msgSegment=yes')

select msgsend('third logical message of the group, second segment',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes,msgSegment=yes')

select msgsend('third logical message of the group, third segment',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes,msgLastSegment=yes')

select msgsend('fourth logical message of the group',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,lastMsgInGroup=yes')
commit
```

例 13 (MQ) *alter_user=yes* オプションを *msgsend* で使用して、ユーザ Joe (彼の SQL ログインは “joe” です) がマシン “host1” 上に “joe” と呼ばれるユーザ ID がなくても、そのマシン上で稼働している MQ アプリケーションとの間でメッセージを Adaptive Server 経由で送受信できるようにします。

```
select msgsend('Hello world',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=joeQM,queue=QUEUE1,alter_user=yes')
```

例 14 (MQ) *msgsend* を使用してサブスクライバをいったん登録してから、登録を解除します。サブスクライバは、トピック “A” または “A/B/*” に一致するパブリケーションすべてに関連します。一致するパブリケーションは、MQ *pub/sub* プローカによってキュー “Q2” に転送されます。

```
-- Register the subscriber
select msgsend(null,
  'ibm_mq:channel1/TCP/host1(5678)'
  + '?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEUE'
  option 'msgType=datagram,rfhCommand=registerSubscriber'
  message property 'topics='A:A/B/*',streamName=stream1,queueName=Q2')

-- Publish a message to the stream queue, let it do implicit registration
```

```
select msgsend('happy birthday',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,
    queue=stream1'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='A''')

-- Read a message forwarded to us by the MQ pub/sub
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q2'
  option 'timeout=50ss')

-- Deregister the subscriber
select msgsend(null,
  'ibm_mq:channel1/TCP/host1(5678)'
  + ?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEUE'
  option 'msgType=datagram,rfhCommand=deregisterSubscriber'
  message property 'topics='A:A/B/*',streamName=stream1,queueName=Q2')
```

例 15 (MQ) msgsend の clustQBinding=bind オプションを表示します。 ローカル “INVC” キュー・マネージャは Q1 クラスタ・キューのメンバであり、Q1 はクラスタ・キューです。

```
select msgsend(
  "M", "ibm_mq:CH1/TCP/box1(5599)?qmgr=INVC,queue=Q1,alter_user=yes",
  option "clustQBinding=bind")
```

ユーザがこの `select` 文を初めて実行すると、MQOPEN 呼び出しによってメッセージを受信するクラスタ・キュー・マネージャが選択されます。同じ SQL セッション中に発行されるそれ以降の文は、同じキュー・マネージャに自動的にルート指定されます。

例 16 (MQ) msgsend の clustQBinding=nobind オプションを表示します。 メッセージを受信するクラスタ・キュー・マネージャは、毎回選択されます。

```
select msgsend(
  "M", "ibm_mq:CH1/TCP/box1(5599)?qmgr=INVC,queue=Q1,alter_user=yes",
  option "clustQBinding=nobind")
```

例 17 (MQ) msgsend の clustQBinding = default オプションを表示します。 動作はキューの “DEFBIND” プロパティによって判断されます。値が “open” である場合、動作は `clustQBinding=bind` と同じです。それ以外の場合には、動作は `clustQBinding=nobind` と同じです。

```
select msgsend(
  "M", "ibm_mq:CH1/TCP/box1(5599)?qmgr=INVC,queue=Q1,alter_user=yes",
  option "clustQBinding=default")
```

使用法

- 送信先に `queue=queue_name` の形式がある場合、メッセージはこのキューに送信されます。
- `service_provider_class` および “user” と “password” では、大文字と小文字は区別されません。 `local_name`、 `hostname`、 `port`、 `queue_name`、 `user_name`、 `password` の各パラメータでは、大文字と小文字が区別されます。
- [表 3-3 \(48 ページ\)](#) に従って Adaptive Server 固有のメッセージ・プロパティを設定できます。
- `msgsend` でのオプション文字列の使用法は、次のとおりです。
 - 空のオプション文字列は無視される。
 - 複数のオプション文字列をカンマまたはスペースで区切ることができる (最初のオプションの前、最後のオプションの後ろ、オプションどうしの間、囲む等号の前には、任意の数のスペースを使用できる)。
 - 引用符で囲まれた文字列には、埋め込み引用符に関する SQL 表記規則を適用する。
 - 名前が同じオプションを複数指定すると、最後に指定されたオプションだけが処理されます。たとえば、次の文では 'priority' について使用されるか検証されるのは、値 7 だけです。その他の値は無視されます。

```
select msgsend( 'Hello Messaging World!',
               'my_jms_provider?queue=queue.sample',
               MESSAGE PROPERTY 'priority='high', priority=yes, priority=7')
```

- `msgsend` が実行された後は、その呼び出しの情報を使用してグローバル変数の値が設定されます。「[メッセージ関連のグローバル変数](#)」(40 ページ)を参照してください。
- 引用符付きオプションまたはプロパティ値は、二重引用符ではなく一重引用符で囲みます。

注意 `msgsend` でメッセージをトピックに送信できるのは、`topic=topic_name` を送信先として指定した場合です。ただし、予期しない動作が発生することがあるので、これはお勧めしません。

- 認識されないオプションとプロパティは無視されますが、このようなオプションとプロパティの値は、エラーとして通知されます。

注意 この動作は、Adaptive Server バージョン 12.5.3a から変更されたものであり、それ以前のバージョンとは異なります。

msgsend オプション *option_string* パラメータの値

表 3-11 に JMS で使用できる msgsend オプション・パラメータを示します。

表 3-11: msgsend で有効な JMS オプション *option_string* の型と値

データ型	値	デフォルト	説明
schema	<ul style="list-style-type: none"> • no • yes • "user_schema" 	no	<ul style="list-style-type: none"> • <i>user_schema</i> はユーザ指定スキーマであり、<i>message_body</i> を記述する。 • no は、生成されメッセージの一部として送信される schema がないことを示す。 • yes は、Adaptive Server がメッセージ用の XML スキーマを生成することを示す。yes が意味を持つのは、パラメータ <i>select_for_xml</i> を使用する <i>message_body</i> 内だけです。<i>select_for_xml</i> は SQL 結果セットの SQLX フォーマット表現を生成します。生成された XML スキーマは、結果セット・ドキュメントを記述する SQLX フォーマット・スキーマです。 <p>このスキーマは、ASE_MSGBODY_SCHEMA プロパティとしてメッセージに埋め込まれます。</p>
type	text, bytes	text	送信するメッセージのタイプ。

表 3-12 に MQ で使用できる msgsend オプション・パラメータを示します。

表 3-12: msgsend で有効な MQ オプション *option_string* の型と値

データ型	値	デフォルト	説明
msgType	<ul style="list-style-type: none"> • datagram • request • reply • report 	datagram	<p>メッセージ・タイプが次の場合、</p> <ul style="list-style-type: none"> • request – <i>replyQueue</i> プロパティも指定する必要がある。 • report – <i>reportDataHeader</i> プロパティと <i>feedback</i> プロパティも指定する必要がある。

データ型	値	デフォルト	説明
rfhCommand	<ul style="list-style-type: none"> • null • deletePublication • deregisterPublisher • deregisterSubscriber • publish • registerPublisher • registerSubscriber • requestUpdate 	null	<p>MQ pub/sub では、MQRF ヘッダが制御メッセージとなり、キューに送信され MQ pub/sub ブローカによって読み取られます。ブローカは、キューから読み取ったメッセージを処理します。</p> <p>rfhCommand が null である場合、メッセージに MQRF ヘッダは格納されません。メッセージに MQRF ヘッダが格納されるのは、rfhCommand にそれ以外の値が設定されており、かつ MQPSCCommand が次のように設定されているときです。</p> <ul style="list-style-type: none"> • deletePublication が DeletePub に設定されている。終了ポイントが、パブリッシュする側のストリーム・キューの終了ポイントである。表 3-15 (121 ページ) を参照してください。 • deregisterPublisher が DeregPub に設定されている。表 3-16 (122 ページ) を参照してください。 • deregisterSubscriber が DeleteSub に設定されている。表 3-17 (123 ページ) を参照してください。 • publish が Publish に設定されている。終了ポイントが、パブリッシュする側のストリーム・キューの終了ポイントである。表 3-18 (124 ページ) を参照してください。 • registerPublisher が RegPub に設定されている。表 3-15 (121 ページ) を参照してください。 • registerSubscriber が RegSub に設定されている。表 3-20 (128 ページ) を参照してください。 • requestUpdate が ReqUpdate に設定されている。表 3-21 (131 ページ) を参照してください。 <p>メッセージは指定された終了ポイントに送信されません。次のオプションでは、パブリッシュする側のストリーム・キューの終了ポイントを指定してください。</p> <ul style="list-style-type: none"> • publish • deletePublication <p>次のオプションでは、MQ pub/sub ブローカのコントロール・キューの終了ポイントを指定してください。</p> <ul style="list-style-type: none"> • deregisterPublisher • deregisterSubscriber • registerPublisher • registerSubscriber • requestUpdate

データ型	値	デフォルト	説明
alter_user	<ul style="list-style-type: none"> • yes • no 	null	<p>alter_user=yes オプションでは、messaging_role パーミッションを付与されたユーザが、MQ が稼働しているマシンとの間でメッセージの送受信を行うことができます。これは、そのマシン上でのオペレーティング・システム (ログイン) ID を持っていない場合でも適用されます。</p> <p>管理者がこのオプションを設定せず、ユーザが MQ が稼働しているマシン上でのログイン ID を持っていないと、MQ 認証に失敗し、メッセージング操作は機能しません。</p> <hr/> <p>注意 MQ が稼働しているマシンで Adaptive Server Enterprise も稼働していない場合には、alter_user=yes の実行後であっても、エラー・メッセージが表示されます。これを回避するには、Adaptive Server を起動したユーザのユーザ ID と同じログインを MQ マシン上で新規作成します。</p>
clustQBinding	<ul style="list-style-type: none"> • bind • nobind • default 	default	<p>clustQBinding オプションを使用すると、複数のメッセージを同じ 1 つのインスタンスに入れることができます。1 つのメッセージをクラスタ・キューに送信しない場合には、このオプションは無視されます。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • bind – WebSphere MQ は、メッセージを初めて開くとき、メッセージの送信先と自らをホストするキュー・マネージャの両方を選択します。そして、MPOPEN 呼び出しが行われたときに決定された送信先への MQPUT 呼び出しすべてを判断します。 • nobind – WebSphere MQ は、MQ に対してメッセージをキューに入れる要求が行われるたびに別の送信先をメッセージに選択します。その際に、MPOPEN 呼び出しによって取得されたクラスタ・キュー・ハンドラを使用して MQPUT が実行されるたびに送信先が選択されます。メッセージの送信先は、負荷分散に関する考慮結果 (このオプションが有効の場合) とキュー・マネージャの手が空いているかどうかに基づいて決まります。 • default – 送信先はクラスタ・キュー定義レベルで指定されたバインド・プロパティによって指定されます。この動作は、クラスタ・システムを使用している場合でも clustQBinding オプションを指定しない場合にも発生します。

msgsend *properties_clause* パラメータ値

表 3-13 に MQ で使用できる *msgsendproperties_clause* パラメータを示します。

表 3-13: msgsend で有効な MQ メッセージ・プロパティ *property_option_clause* の型と値

データ型	値	デフォルト	説明
arrivalReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>このメッセージが最終送信先に届くと、到着時確認 (COA) レポートが生成されます。</p> <p>replyToQueue を必ず指定してください。次のオプションがあります。</p> <ul style="list-style-type: none"> • yes – 受信メッセージのデータなしに COA レポートが生成される。 • withData – 受信メッセージの最初の 100 バイトのデータで COA レポートが生成される。 • withFullData – 受信メッセージの全データで COA レポートが生成される。 • no – COA レポートは生成されない。
correlationId	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>複数のメッセージを 1 つにまとめるためにクライアントが相関 ID を設定します。</p> <p>MQ では、この文字列は 24 バイトに制限されます。</p> <p>MQ ではこのフィールドを unsigned char と定義しています。つまり、binary 値をサポートできます。binary 文字列を correlationId として入力するには、値として “0x...” を使用します (値を引用符で囲まないでください)。</p> <p>rfhCommand が null でない場合、次のようになります。</p> <ul style="list-style-type: none"> • correlationId が null でない場合、新しい相関 ID は要求されません。correlationAsId が yes であり、かつ correlationId が null である場合には、これは独立した従来型 ID (相関 ID が空の ID) となります。 • deletePublication、deregisterPublisher、publish、registerPublisher の rfhCommands では、指定される相関 ID はパブリッシャの従来型 ID となります。

データ型	値	デフォルト	説明
deliveryReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>このメッセージが最終送信先から届くと、到着時確認 (COA) レポートが生成されます。</p> <p>replyToQueue を必ず指定してください。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – 受信メッセージのデータなしに COA レポートが生成される。 • withData – 受信メッセージの最初の 100 バイトのデータで COA レポートが生成される。 • withFullData – 受信メッセージの全データで COA レポートが生成される。 • no – COA レポートは生成されない。
exceptionReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>このメッセージの有効期限が切れるか send に失敗すると、例外レポートが生成されます。</p> <p>replyToQueue を必ず指定してください。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – 受信メッセージのデータなしに例外レポートが生成されます。 • withData – 受信メッセージの最初の 100 バイトのデータで例外レポートが生成されます。 • withFullData – 受信メッセージの全データで例外レポートが生成されます。 • no – 例外レポートは生成されません。
expirationReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>この send に失敗すると、例外レポートが生成されます。</p> <p>replyToQueue を必ず指定してください。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – 受信メッセージのデータなしに例外レポートが生成されます。 • withData – 受信メッセージの最初の 100 バイトのデータで例外レポートが生成されます。 • withFullData – 受信メッセージの全データで例外レポートが生成されます。 • no – 例外レポートは生成されません。

データ型	値	デフォルト	説明
expiry	-1 ~ 214748364799 の timespec	-1、有効期限なし	<p>キュー・マネージャでのメッセージの有効期間。 timespec が整数の場合、単位はミリ秒です。 有効な値は次のとおりです。</p> <ul style="list-style-type: none"> 0 – メッセージは期限切れにならない。 -1 – キューのデフォルト値を使用する。 <hr/> <p>注意 expiry は 0.1 秒単位なので、この数は 0.1 秒の位に四捨五入されてから MQ に渡される。</p> <hr/> <p>「timespec」(143 ページ) を参照してください。</p>
feedback	<i>integer</i> 範囲は MQFB_APPL_FIRST (65536) ~ MQFB_APPL_LAST (999999999)	0	<p>レポート・メッセージの場合、feedback はレポート・メッセージの特性を示すコードです。 MQ は次の項目ごとに 1 つの feedback コードを定義します。</p> <ul style="list-style-type: none"> システム・レポート・メッセージ アプリケーション・レポート・メッセージ
formatName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>メッセージ・フォーマットについての情報を渡すためのアプリケーション指定プロパティです。 このプロパティでは、送信側アプリケーションがメッセージ・データを記述するフォーマット名を設定できます。 受信側アプリケーションは、@@msgheader 内の formatName をチェックして、メッセージ・データの処理方法を決定できます。 “MQ” で始まる名前は予約されています。 MQ では、この文字列は 8 バイトに制限されます。</p>
groupId	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>ユーザ定義グループ。 MQ では、この文字列は 24 バイトに制限されます。 MQ ではこのフィールドを unsigned char と定義しています。つまり、binary 値をサポートできます。binary 文字列を groupId として入力するには、値として “0x...” を使用します。値を引用符で囲まないでください。囲むと、引用符付き文字列として解釈されてしまいます。 groupId を指定せずにいずれかのグループ化プロパティを指定した場合、キュー・マネージャがグループ名を生成します。 ordering が logical に設定された場合には、無視されます。 1 つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>

データ型	値	デフォルト	説明
lastMsgInGroup	<ul style="list-style-type: none"> • yes • no 	no	<p>値が yes である場合、メッセージをグループの最終論理メッセージとしてマーク付けします。</p> <p>単一の論理メッセージだけで1つのグループを構成するには、lastMsgInGroup を yes に設定します。</p> <p>1つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>
mode	<ul style="list-style-type: none"> • persistent • non-persistent • default 	default	<p>mode の設定に応じて、次のようになります。</p> <ul style="list-style-type: none"> • persistent – メッセージはステープル・ストレージを使用してメッセージング・プロバイダによってバックアップされる。メッセージを消費できるようになる前にメッセージング・プロバイダに障害が発生すると、メッセージは高い確率で保存される。 • non-persistent でありメッセージング・プロバイダに障害が発生 – メッセージは送信先に届く前に失われることがあります。 • default – キューのデフォルト値を使用する。
msgld	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>指定されると、WebSphere MQ は既存のメッセージ ID を msgld に指定された値で置換します。</p> <p>MQ では、この文字列は 24 バイトに制限されます。MQ ではこのフィールドを “unsigned char” と定義しています。つまり、binary 値をサポートできます。</p> <p>binary 文字列を msgld として入力するには、値として “0x...” を使用します。値を引用符で囲まないでください。</p>
msgInGroup	<ul style="list-style-type: none"> • yes • no 	no	<p>値が yes である場合、このメッセージはメッセージ・グループの論理メッセージです。</p> <p>1つのグループに属する複数のメッセージの場合、論理メッセージすべてに対してこのプロパティを yes に設定してください。ただし、最終論理メッセージを除きます。最終論理メッセージでは、lastMsgInGroup を yes に設定する必要があります。</p> <p>1つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>
msgLastSegment	<ul style="list-style-type: none"> • yes • no 	no	<p>値が yes である場合、このメッセージはセグメント化されたメッセージの最終セグメントです。1つのセグメント・メッセージだけで1つのローカル・メッセージを構成するには、メッセージの msgLastSegment を yes に設定します。</p> <p>値が yes であつ ordering が physical に設定されている場合、offset プロパティも設定します。</p> <p>1つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>

データ型	値	デフォルト	説明
msgSegment	<ul style="list-style-type: none"> • yes • no 	no	<p>値が yes である場合、このメッセージはセグメント化されたメッセージのセグメントです。単一セグメントの一部である複数のメッセージの場合、セグメントすべてに対してこのプロパティを yes に設定してください。ただし、最終セグメントを除きます。最終セグメントでは、msgLastSegment を yes に設定する必要があります。</p> <p>値が yes であかつ ordering が physical に設定されている場合、offset プロパティも設定します。</p> <p>1つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>
negativeActionReport	<ul style="list-style-type: none"> • yes • no 	no	<p>replyToQueue を必ず指定してください。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – 取得側アプリケーションがこのメッセージを読み取り、ネガティブに対処すると、ネガティブ・アクション通知 (NAN) レポートが生成される。 • no – NAN レポートは生成されない。
offset	-1、0 ~ maxint の <i>integer</i>	-1	<p>メッセージがセグメント化されたメッセージのセグメントである場合、offset を論理メッセージ内の現在のメッセージのバイト・オフセットに設定します。</p> <p>-1 は offset が指定されないことを示します。</p> <p>msgSegment か msgLastSegment も指定されない限り、offset は無視されます。</p> <p>msgpublish によって無視されます。</p> <p>ordering が logical に設定された場合には、無視されます。</p> <p>1つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>
onNoDelivery	<ul style="list-style-type: none"> • deadLetter • discard 	deadLetter	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • deadLetter – 配信できないメッセージは、配信不能キューに入れられる。 • discard – メッセージはキュー・マネージャによって廃棄される。

データ型	値	デフォルト	説明
ordering	<ul style="list-style-type: none"> logical physical 	physical	<p>このプロパティの設定に応じて、次のようになります。</p> <ul style="list-style-type: none"> physical – アプリケーションはグループ (またはセグメント化されたメッセージ) の一部であるメッセージを任意の順序で送信できる。キュー・マネージャは、セグメントの紛失またはシーケンス識別子のギャップを検出すると、エラーを返します。 logical – アプリケーションは <code>msgInGroup</code>、<code>lastMsgInGroup</code>、<code>msgSegment</code>、<code>lastMsgSegment</code> の各オプションを適切に設定するだけでよい。グループ名、シーケンス識別子、セグメント・オフセットはキュー・マネージャが自動的に設定します。
positiveActionReport	<ul style="list-style-type: none"> yes no 	no	<p><code>replyToQueue</code> を必ず指定してください。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> yes – 取得側アプリケーションがこのメッセージを読み取り、ポジティブに対処すると、ポジティブ・アクション通知 (PAN) レポートが生成される。 no – PAN レポートは生成されない。
priority	<p><i>integer</i></p> <ul style="list-style-type: none"> -1, 0 (キュー・マネージャ) 設定された <code>max priority</code> 	-1	<p>メッセージの優先度を制御します。指定する値は次のとおりです。</p> <ul style="list-style-type: none"> -1 – キューのデフォルト優先度を使用する。 指定された <code>priority</code> が、キュー・マネージャに指定された <code>max priority</code> より大きい – キュー・マネージャに指定された最大の優先度を使用する。これは MQ によって実装されます。
replyCorrelationId	<ul style="list-style-type: none"> msgId correlationId 	msgId	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> msgId – レポート・メッセージの相関 ID は、受信メッセージのメッセージ ID を使用する。 correlationId – レポート・メッセージの相関 ID は、受信メッセージの相関 ID を使用する。
replyMsgId	<ul style="list-style-type: none"> new original 	new	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> new – 生成されたレポート・メッセージに新しいメッセージ ID が格納される。 original – レポート・メッセージが受信メッセージと同じメッセージ ID を使用する。

データ型	値	デフォルト	説明
replyToInputMode	<ul style="list-style-type: none"> • browse • Qdefault • shared • exclusive • browse+Qdefault • browse+shared • browse+exclusive 	Qdefault	<p>replyToQueue が開いているモード。</p> <p>replyToQueue を指定すると、それ以降の入力に対してキューが自動的に開かれます。このモードは、replyToQueue が開いているモードを指定します。</p> <p>replyToQueue を指定しないと、このプロパティは無視されます。</p> <p>モードには次の意味があります。</p> <ul style="list-style-type: none"> • browse – キューが開かれるのは参照目的のみ。破壊的読み取りを試みると、キュー・マネージャがエラー・メッセージを発行します。 • Qdefault – キューはデフォルトの入力モードで開かれる。 • shared – キューは共有入力モードで開かれる。キューが他の MQ ハンドルによって排他モードで既に開かれている場合、エラー・メッセージが表示されます。 • exclusive – キューは排他的入力モードで開かれる。キューが他の MQ ハンドルによって共有モードまたは排他モードで既に開かれている場合、エラー・メッセージが表示されます。 • browse+Qdefault – キューは参照目的とデフォルトの入力モードで開かれる。 • browse+shared – キューは参照目的と共有入力モードで開かれる。キューが他の MQ ハンドルによって排他モードで既に開かれている場合、エラー・メッセージが表示されます。 • browse+exclusive – キューは参照目的と排他的入力モードで開かれる。キューが他の MQ ハンドルによって共有モードまたは排他モードで既に開かれている場合、エラー・メッセージが表示されます。
replyToModel	<ul style="list-style-type: none"> • null • string 	null	<p>replyToQueue が動的キューの場合に応答キューの作成元となるモデル・キューの名前です。</p> <p>replyToQueue を指定しないと、このプロパティは無視されます。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
replyToQmgr	<ul style="list-style-type: none"> • null • string 	null	<p>replyToQueue が存在するキュー・マネージャが将来使用するために予約されています。現時点では、replyToQueue は常に接続先のキュー・マネージャ上に存在します。</p>

データ型	値	デフォルト	説明
replyToQueue	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>要求メッセージへの応答を返すとアプリケーションが期待するキュー。</p> <hr/> <p>注意 送信されるメッセージ・タイプは、request だけでなくかまいません。MQ でこのことは強制されないからです。</p> <hr/> <p>指定されたキュー名が“*”で終わる場合、システム生成の動的キュー名が、指定されたプレフィクス付きで生成されます。</p> <p>replyToModel と動的キュー名が指定された場合、replyToModel 用に指定されたモデル・キューから動的キューが作成されます。</p> <p>send 操作後に @@msgreplytoinfo セッション変数を介して、システム生成の動的キュー名を取得できます。</p> <hr/> <p>注意 動的キュー名を指定した場合には、動的キューを作成するために、現在の Adaptive Server ログインにキュー・マネージャでの“crt”権限が存在する必要があります。</p> <hr/> <p>動的キュー名を指定した場合、受信側アプリケーションが作成された動的キューを削除しないときには、管理者が手動で削除してください。</p> <p>rfhCommand が null でない場合、replyToQueue を指定して MQ pub/sub プローカから応答を取得できます。</p>

データ型	値	デフォルト	説明
rfhCommand	<ul style="list-style-type: none"> • null • deletePublication • deregisterPublisher • deregisterSubscriber • publish • registerPublisher • registerSubscriber • requestUpdate 	null	<p>MQ pub/sub では、MQRF ヘッダが制御メッセージとなり、キューに送信され MQ pub/sub ブローカによって読み取られます。ブローカは、キューから読み取ったメッセージを処理します。</p> <p>rfhCommand が null である場合、メッセージに MQRF ヘッダは格納されません。メッセージに MQRF ヘッダが格納されるのは、rfhCommand にそれ以外の値が設定されており、かつ MQPSCCommand が次のように設定されているときです。</p> <ul style="list-style-type: none"> • deletePublication が DeletePub に設定されている。終了ポイントが、パブリッシュする側のストリーム・キューの終了ポイントである。表 3-15 (121 ページ) を参照。 • deregisterPublisher が DeregPub に設定されている。表 3-16 (122 ページ) を参照。 • deregisterSubscriber が DeleteSub に設定されている。表 3-17 (123 ページ) を参照。 • publish が Publish に設定されている。終了ポイントが、パブリッシュする側のストリーム・キューの終了ポイントである。表 3-18 (124 ページ) を参照。 • registerPublisher が RegPub に設定されている。表 3-15 (121 ページ) を参照。 • registerSubscriber が RegSub に設定されている。表 3-15 (121 ページ) を参照。 • requestUpdate が ReqUpdate に設定されている。表 3-15 (121 ページ) を参照。 <p>メッセージは指定された終了ポイントに送信されません。次のオプションでは、パブリッシュする側のストリーム・キューの終了ポイントを指定してください。</p> <ul style="list-style-type: none"> • publish • deletePublication <p>次のオプションでは、MQ pub/sub ブローカのコントロール・キューの終了ポイントを指定してください。</p> <ul style="list-style-type: none"> • deregisterPublisher • deregisterSubscriber • registerPublisher • registerSubscriber • requestUpdate

データ型	値	デフォルト	説明
sequenceId	-1 ~ 9,999,999 の <i>integer</i>	-1	<p>グループの一部である論理メッセージを順序付けるために使用されます。</p> <p>-1 は sequenceId が指定されないことを示します。</p> <p>msgInGroup か lastMsgInGroup が共に指定されない限り、sequenceId は無視されます。</p> <p>msgpublish によって無視されます。</p> <p>ordering が logical に設定された場合には、無視されます。</p> <p>1つのグループのメッセージは、すべて同じトランザクションで送信してください。</p>

表 3-14 に JMS で使用できる msgsendproperties_clause パラメータを示します。

表 3-14: msgsend で有効な JMS メッセージ・プロパティ properties_option_string の型と値

オプション	値	デフォルトと特性	説明
ttl	0 ~ (2 ⁶³ -1)	0 ヘッダ	<p>ttl は、メッセージング・バスでの有効期間を指します。Adaptive Server はこの影響を受けません。</p> <p>有効期間情報とは、メッセージが有効である期間をミリ秒単位で示したものです。たとえば、60 はメッセージの有効期間が 60 ミリ秒であることを示します。</p> <p>値が 0 の場合、メッセージの有効期間が切れないことを意味します。</p> <p>ttl は timespec オプションを使用します。 「timespec」(143 ページ)を参照してください。</p>
priority	1 ~ 9	4 ヘッダ	<p>priority の動作は、基本となるメッセージ・バスによって制御されます。この値が Tibco JMS に適用されます。</p> <p>通常の優先度は 1 ~ 4 です。優先度が特に高いのは 5 ~ 9 です。</p>
correlation	<i>string</i>	なし ヘッダ	<p>複数のメッセージを 1 つにまとめるためにクライアント・アプリケーションが相関 ID を設定します。Adaptive Server が、アプリケーションが指定する相関 ID を設定します。</p>
mode	<ul style="list-style-type: none"> • persistent • non-persistent 	<p>persistent</p> <p>ヘッダ</p>	<p>モードは次のとおりです。</p> <ul style="list-style-type: none"> • persistent – メッセージはステابل・ストレージを使用して JMS プロバイダによってバックアップされます。メッセージが消費される前にメッセージング・プロバイダに障害が発生すると、メッセージは高い確率で保存されます。 • non-persistent でありメッセージング・プロバイダに障害が発生 – メッセージは送信先に届く前に失われることがあります。

オプション	値	デフォルトと特性	説明
replyqueue	<i>queue_name</i> が含まれる文字列	なし ヘッダ	<i>queue_name</i> または <i>topic_name</i> の値に応じて、次のようになります。
replytopic	<i>topic_name</i> が含まれる文字列	なし ヘッダ	<ul style="list-style-type: none"> • <i>syb_temp</i> – Adaptive Server が一時的な送信先を作成し、新規作成した送信先に関連する情報をヘッダ情報の一部として送信します。 次に、Adaptive Server は <i>@@msgreplytoinfo</i> を新規作成した一時的な送信先として更新します。 一時的な送信先の種類がキューなのかトピックなのかは、<i>replyqueue</i> と <i>replytopic</i> のどちらを指定するのにかによって決まります。最後に列挙されたオプションだけが使用されます。 • 既存の送信先 – Adaptive Server は送信先を作成せず、その代わりにユーザが指定した送信先を使用する。

msgsend プロパティと *rfhCommand*

MQ の場合、表 3-15 に記載されたプロパティが有効になるのは、*rfhCommand* が *deletePublication* である場合だけです。

表 3-15: *rfhCommand* が *deletePublications* に設定された場合の *msgsend* プロパティ

プロパティ	値	デフォルト	説明
<i>local</i>	<ul style="list-style-type: none"> • <i>yes</i> • <i>no</i> 	<i>no</i>	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • <i>yes</i> – このブローカでローカルにパブリッシュされた保管済みパブリケーションだけが削除される。 • <i>no</i> – グローバルに保管されたパブリケーションが、ネットワーク内のすべてのブローカから削除される。
<i>streamName</i>	<ul style="list-style-type: none"> • <i>null</i> • <i>string</i> 	<i>null</i>	<p>指定されたトピックのパブリケーション・ストリームの名前。</p> <p>指定されない場合、この MQRFH コマンド・メッセージの送信先ストリーム・キューがデフォルトになる。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
<i>topics</i>	<i>string</i>	なし	<p>「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。</p> <p>このトピックに一致する保管されたメッセージが削除される。</p> <p>1 つ以上のトピックを指定する必要がある。</p> <p>必須プロパティであるため、省略されるとエラーを生成する。</p>

MQ の場合、表 3-15 に記載されたプロパティが有効になるのは、rhfCommand が deregisterPublisher である場合だけです。

表 3-16: rhfCommand が deregisterPublisher に設定された場合の msgsend プロパティ

プロパティ	値	デフォルト	説明
deregAll	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – このパブリッシャに登録されたトピックの登録がすべて解除され、topics プロパティが無視される。 • no – 登録済みトピックの登録は解除されない。 <p>topics が指定されると、Adaptive Server はエラーを返します。</p>
streamName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • null 以外 – パブリケーション・ストリームの名前。 • null – SYSTEM.BROKER.DEFAULT.STREAM と想定。 <p>MQ では、この文字列は 48 バイトに制限されます。</p>
topics	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。</p> <p>このパブリッシャが登録を解除するトピック。</p> <p>次の場合、Adaptive Server はエラーを返します。</p> <ul style="list-style-type: none"> • deregAll プロパティが yes に設定されている。 • topics が null でない。
qmgrName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>パブリッシャのキュー・マネージャ名であり、パブリッシャの従来型 ID を確立するために使用される。パブリッシャ登録時に指定した値と同じ値を指定してください。</p> <p>null の場合、デフォルトで replyToQmgr になります。</p>
queueName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>パブリッシャのキュー名であり、パブリッシャの従来型 ID を確立するために使用される。パブリッシャ登録時に指定した値と同じ値を指定してください。</p> <p>null の場合、デフォルトで replyToQueue になります。</p>
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – correlationId がパブリッシャの従来型 ID の一部として使用される。correlationId の指定は必須だが、0x00 としては登録できない。 • no – correlationId がパブリッシャの従来型 ID の一部として使用されない。 • generate – システム生成の correlationId がパブリッシャの従来型 ID の一部として使用される。

MQ の場合、表 3-17 に記載されたプロパティが有効になるのは、`rhfCommand` が `deregisterSubscriber` である場合だけです。

表 3-17: `rhfCommand` が `deregisterSubscriber` に設定された場合の `msgsend` プロパティ

プロパティ	値	デフォルト	説明
<code>deregAll</code>	<ul style="list-style-type: none"> • <code>yes</code> • <code>no</code> 	<code>no</code>	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • <code>yes</code> – このサブスクライバのトピックの登録がすべて解除される。<code>topics</code> プロパティは無視される。 • <code>no</code> – サブスクライバ・トピックの登録は解除されない。 <p><code>topics</code> が <code>null</code> でない場合、Adaptive Server はエラーを返します。</p>
<code>streamName</code>	<ul style="list-style-type: none"> • <code>null</code> • <code>string</code> 	<code>null</code>	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • <code>null</code> 以外 – パブリケーション・ストリームの名前。 • <code>null</code> – <code>SYSTEM.BROKER.DEFAULT.STREAM</code> と想定。 <p>MQ では、この文字列は 48 バイトに制限されます。</p>
<code>topics</code>	<ul style="list-style-type: none"> • <code>null</code> • <code>string</code> 	<code>null</code>	<p>「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。</p> <p>このサブスクライバが登録を解除するトピック。次の場合、Adaptive Server はエラーを返します。</p> <ul style="list-style-type: none"> • <code>deregAll</code> が <code>Yes</code>。 • <code>topics</code> が <code>null</code> でない。
<code>qmgrName</code>	<ul style="list-style-type: none"> • <code>null</code> • <code>string</code> 	<code>null</code>	<p>サブスクライバのキュー・マネージャ名であり、サブスクライバの従来型 ID を確立するために使用される。サブスクライバ登録時に指定した値と同じ値を指定してください。</p> <p><code>null</code> の場合、デフォルトで <code>replyToQmgr</code> になります。</p>
<code>queueName</code>	<ul style="list-style-type: none"> • <code>null</code> • <code>string</code> 	<code>null</code>	<p>サブスクライバのキュー名であり、サブスクライバの従来型 ID を確立するために使用される。サブスクライバ登録時に指定した値と同じ値を指定してください。</p> <p><code>null</code> の場合、デフォルトで <code>replyToQueue</code> になります。</p>

プロパティ	値	デフォルト	説明
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – correlationId がパブリッシャの従来型 ID の一部として使用される。correlationId の指定は必須だが、0x00 としては登録できない。 • no – correlationId がパブリッシャの従来型 ID の一部として使用されない。 • generate – システム生成の correlationId がパブリッシャの従来型 ID の一部として使用される。

MQ の場合、[表 3-18](#) に記載されたプロパティが有効になるのは、rhfCommand が publish である場合だけです。

表 3-18: rfhCommand が publish に設定された場合の msgsend プロパティ

プロパティ	値	デフォルト	説明
topics	string	なし	<ul style="list-style-type: none"> • 「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。 • ワイルドカードは使用できない。 • このパブリケーションが情報を持っているトピック。 • 必須プロパティであるため、省略されるとエラーを生成する。
anon	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – パブリッシャの ID は MQ pub/sub ブローカによって明かされない。noReg が yes の場合、無視される。 • no – パブリッシャの ID は MQ pub/sub ブローカによって明かされる。
local	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – MQ pub/sub ブローカはこのパブリケーションを、local を指定して登録したサブスクライバだけに送信する。noReg が yes の場合、無視される。 • no – MQ pub/sub ブローカはこのパブリケーションをすべてのサブスクライバに送信する。

プロパティ	値	デフォルト	説明
directReq	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – パブリッシャはパブリケーション情報を求める他のアプリケーションからの直接要求を受け入れる。noReq が yes の場合、無視される。 <p>anon プロパティも yes に設定されている場合には、このオプションを yes に設定しないでください。設定すると、MQ pub/sub ブローカがエラーを返します。</p> <ul style="list-style-type: none"> • no – パブリッシャはパブリケーション情報を求める他のアプリケーションからの直接要求を受け入れようとしません。
noReg	<ul style="list-style-type: none"> • yes • no 	no	<p>パブリッシャが MQ pub/sub ブローカにこのストリームとトピックのパブリッシャとしてまだ登録されておらず、かつ NoReg の値が次の場合には、以下ようになります。</p> <ul style="list-style-type: none"> • yes – MQ pub/sub ブローカは暗黙的な登録を行わない。anon、local、directReq の各プロパティは無視される。 • no – MQ pub/sub ブローカは暗黙的な登録を行う。その際に、anon、local、directReq によって設定された値を使用する。 <p>パブリッシャが登録済みであり、かつanon、local、または directReq が yes に設定されている場合、既存の登録はこれらのプロパティに従って変更されます。</p>
otherSubsOnly	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – パブリッシャにこのパブリケーションに関するサブスクリプションがある場合、MQ pub/sub ブローカはこのパブリケーションをこのパブリッシャに送信する。 • no – パブリッシャにこのパブリケーションに関するサブスクリプションがあっても、MQ pub/sub ブローカはこのパブリケーションをこのパブリッシャに送信しない。
publishSequenceId	-1、0 ~ (2 ³² - 1) の number	-1	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • -1 以外 – パブリケーションのシーケンス番号。パブリケーションごとに番号が上がっていくが、MQ pub/sub ブローカは検証しない。 • -1 – シーケンス番号は設定されない。

プロパティ	値	デフォルト	説明
publishTimeStamp	<ul style="list-style-type: none"> • null • <i>integer</i> 	null	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • null 以外 – YYYYMMDDHHMSSth 形式のパブリケーションのタイムスタンプ (万国標準時を使用)。形式は検証されない。 • null – パブリケーションのタイムスタンプは設定されない。
qmgrName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>パブリッシャの従来型 ID を判断するために使用されるキュー・マネージャ。サブスクライバが直接要求をこのパブリッシャに送信できる場所でもある。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
queueName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>パブリッシャの従来型 ID を判断するために使用されるキュー。サブスクライバが直接要求をこのパブリッシャに送信できる場所でもある。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
retainPub	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – パブリッシャにこのパブリケーションに関するサブスクリプションがあっても、MQ pub/sub ブローカはこのパブリケーションをこのパブリッシャに送信しない。 • no – パブリッシャにこのパブリケーションに関するサブスクリプションがある場合、MQ pub/sub ブローカはこのパブリケーションをこのパブリッシャに送信する。
stringData	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>null 以外 – パブリッシャが定義するオプションの情報であり、パブリケーションの MQRF ヘッダに格納される。</p> <hr/> <p>注意 MQ pub/sub では MQRF ヘッダに複数の <i>stringData</i> タグを使用できますが、ASE Active Messaging では 1 つしか使用できません。</p> <hr/>
integerData	-1, 0 ~ (2 ³² - 1) の <i>number</i>	-1	<p>-1 以外の場合、パブリッシャが定義するオプションの情報であり、パブリケーションの MQRF ヘッダに格納される。</p> <hr/> <p>注意 MQ pub/sub では MQRF ヘッダに複数の <i>integerData</i> タグを使用できますが、Active Messaging では 1 つしか使用できません。</p> <hr/>

プロパティ	値	デフォルト	説明
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – correlationId がパブリッシャの従来型 ID の一部として使用される。correlationId の指定は必須だが、0x00 としては登録できない。 • no – correlationId がパブリッシャの従来型 ID の一部として使用されない。 • generate – システム生成の correlationId がパブリッシャの従来型 ID の一部として使用される。

MQ の場合、表 3-19 に記載されたプロパティが有効になるのは、rhfCommand が registerPublisher である場合だけです。

表 3-19: rhfCommand が registerPublisher に設定された場合の MQ msgsend プロパティ

プロパティ	値	デフォルト	説明
anon	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – MQ pub/sub ブローカはパブリッシャの ID を明かさない。 • no – MQ pub/sub ブローカはパブリッシャの ID を明かす。
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – correlationId がパブリッシャの従来型 ID の一部として使用される。correlationId の指定は必須だが、0x00 としては登録できない。 • no – correlationId がパブリッシャの従来型 ID の一部として使用されない。 • generate – システム生成の correlationId がパブリッシャの従来型 ID の一部として使用される。
directReq	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – パブリッシャはパブリケーション情報を求める他のアプリケーションからの直接要求を喜んで受け入れる。 • anon プロパティも yes に設定されている場合には、このオプションを yes に設定しないでください。設定すると、MQ pub/sub ブローカがエラーを返します。 • no – パブリッシャはパブリケーション情報を求める他のアプリケーションからの直接要求を喜んで受け入れない。

プロパティ	値	デフォルト	説明
local	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – MQ pub/sub ブローカはこのパブリケーションを、Local を指定して登録したサブスクライバだけに送信する。 • no – MQ pub/sub ブローカはこのパブリケーションをすべてのサブスクライバに送信する。
qmgrName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>パブリッシャの従来型 ID を判断するために使用されるキュー・マネージャ。サブスクライバが直接要求をこのパブリッシャに送信できる場所でもある。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
queueName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>パブリッシャの従来型 ID を判断するために使用されるキュー。サブスクライバが直接要求をこのパブリッシャに送信できる場所でもある。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
streamName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • null 以外 – パブリッシャがパブリケーションをパブリッシュするストリーム。 • null – デフォルトは SYSTEM.BROKER.DEFAULT.STREAM。 <p>MQ では、この文字列は 48 バイトに制限されます。</p>
topics	<i>string</i>	なし	<p>「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。</p> <p>ワイルドカードは使用できない。</p> <p>パブリッシャが情報を提供するトピック。</p> <p>必須プロパティであるため、省略されるとエラーを生成する。</p>

MQ の場合、表 3-20 に記載されたプロパティが有効になるのは、rhfCommand が registerSubscriber である場合だけです。

表 3-20: rhfCommand が registerSubscriber に設定された場合の MQ msgsend プロパティ

プロパティ	値	デフォルト	説明
topics	<i>string</i>	なし	<p>「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。</p> <p>サブスクライバがパブリケーションの受信を希望するトピック。</p> <p>必須プロパティであるため、省略されるとエラーを生成する。</p>

プロパティ	値	デフォルト	説明
anon	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – MQ pub/sub ブローカはサブスクライバの ID を明かさなない。 • no – MQ pub/sub ブローカはサブスクライバの ID を明かす。
local	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – サブスクリプションはネットワーク内の他のブローカに配布されない。local を指定してこのノードからパブリッシュされたパブリケーションだけが、このサブスクライバに送信される。 • no – サブスクリプションは RFH コマンドで指定されない。
newPubsOnly	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – ブローカはこのパブリケーションをこのサブスクライバだけに送信し、登録時に存在する保管されたパブリケーションは送信されない。 • no – パブリケーションは RFH コマンドで指定されない。
pubOnReqOnly	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – ブローカは新しいパブリケーションだけをこのサブスクライバに送信する。登録時に保管されているパブリケーションは送信されない。 • no – パブリケーションは RFH コマンドで指定されない。
inclStreamName	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – ブローカは、MQRF ヘッダ内のパブリケーション・ストリーム名をサブスクライバに転送される各メッセージに追加する。 • no – パブリケーションは RFH コマンドで指定されない。
informIfRet	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – ブローカはサブスクライバにパブリケーションが保管されるかどうかを伝える。保管は、サブスクライバに送信されるメッセージの MQRF ヘッダに MQPSPubsOptsIsRetainedPub を設定することにより行われる。 • no – パブリケーションは RFH コマンドで指定されない。

プロパティ	値	デフォルト	説明
dupsOk	<ul style="list-style-type: none"> • yes • no 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – ブローカは複製したパブリケーションをサブスクライバに時々送信できる。 • no – パブリケーションは RFH コマンドで指定されない。
pubsPersistence	<ul style="list-style-type: none"> • non-persistent • persistent • asPublication • asQueue 	asQueue	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • non-persistent – パブリケーションは非継続的なメッセージとしてサブスクライバ・キューに入れられる。 • persistent – パブリケーションは継続的なメッセージとしてサブスクライバ・キューに入れられる。 • asPublication – パブリケーションは元のパブリケーションと同じ永続性を備えた状態でサブスクライバ・キューに入れられる。 • asQueue – パブリケーションはサブスクライバ・キューのデフォルトの永続性を備えた状態でサブスクライバ・キューに入れられる。
streamName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • null 以外 – パブリッシャがパブリケーションをパブリッシュするストリーム。 • null – サブスクリプションは従来型 ID によって特定される。
qmgrName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>サブスクライバの従来型 ID を判断するために使用されるキュー・マネージャ。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
queueName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>サブスクライバの従来型 ID を判断するために使用されるキュー。</p> <p>MQ では、この文字列は 48 バイトに制限されます。</p>
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	<p>指定する値は次のとおりです。</p> <ul style="list-style-type: none"> • yes – correlationId がサブスクライバの従来型 ID の一部として使用される。correlationId の指定は必須だが、0x00 としては登録できない。 • no – correlationId がサブスクライバの従来型 ID の一部として使用されない。 • generate – システム生成の correlationId がサブスクライバの従来型 ID の一部として使用される。

表 3-21 に記載されたプロパティが有効になるのは、`rhfCommand` が `requestUpdate` である場合だけです。

表 3-21: `rhfCommand` が `requestUpdate` に設定された場合の MQ `msgsend` プロパティ

プロパティ	値	デフォルト	説明
<code>topics</code>	<code>string</code>	なし	「トピックの構文」(18 ページ) で詳細に説明されている形式を使用する。 サブスクライバが要求しているトピック。提供できるのは1つのトピックだけ。 必須プロパティであるため、省略されるとエラーを生成する。
<code>streamName</code>	<ul style="list-style-type: none"> <code>null</code> <code>string</code> 	<code>null</code>	指定する値は次のとおりです。 <ul style="list-style-type: none"> <code>null</code> 以外 – パブリッシャがパブリケーションをパブリッシュするストリーム。 <code>null</code> – デフォルトは <code>SYSTEM.BROKER.DEFAULT.STREAM</code>。
<code>qmgrName</code>	<ul style="list-style-type: none"> <code>null</code> <code>string</code> 	<code>null</code>	サブスクライバの従来型 ID を確立するために使用されるキュー・マネージャ名。サブスクライバ登録時に指定した値と同じ値を指定してください。 MQ では、この文字列は 48 バイトに制限されます。
<code>queueName</code>	<ul style="list-style-type: none"> <code>null</code> <code>string</code> 	<code>null</code>	サブスクライバの従来型 ID を確立するために使用されるキュー。サブスクライバ登録時に指定した値と同じ値を指定してください。 MQ では、この文字列は 48 バイトに制限されます。
<code>correlationAsId</code>	<ul style="list-style-type: none"> <code>yes</code> <code>no</code> <code>generate</code> 	<code>no</code>	指定する値は次のとおりです。 <ul style="list-style-type: none"> <code>yes</code> – <code>correlationId</code> がサブスクライバの従来型 ID の一部として使用される。<code>correlationId</code> の指定は必須だが、<code>0x00</code> としては登録できない。 <code>no</code> – <code>correlationId</code> がサブスクライバの従来型 ID の一部として使用されない。 <code>generate</code> – システム生成の <code>correlationId</code> がサブスクライバの従来型 ID の一部として使用される。

- `message property` を使用する場合、認識されないオプションは無視されません。`msgsend` または `msgpublish` 関数に `message header` を使用した場合、認識されないオプションを指定しようとするとエラーが表示されます。
- `msgsend` 呼び出しの結果は、`varchar` 文字列です。メッセージが正常に送信された場合、戻り値はメッセージ ID となります。メッセージが送信されなかった場合、戻り値は `null` です。

- これらの制限事項は、`service_provider_uri` のランタイム形式に適用されます。

```
service_provider_uri ::=  
  provider_name ?destination [,user=username, password=password]  
  provider_name ::= local_name | full_name  
  local_name ::= identifier  
  full_name ::= service_provider_class:service_provider_url
```

- `local_name` は、プロバイダ識別子。sp_msgadmin 'register', 'provider' の呼び出しに以前登録された識別子であり、その呼び出しで指定された `full_name` の簡略表現である。
- 現在サポートされている唯一の `service_provider_class` は JMS である。
- `service_provider_url` の形式は “tcp://hostname:port”。ホスト名には名前または IP アドレスを使用できる。
- `service_provider_url` にはスペースを使用できない。

MQ

- msgsend が返すステータスは、メッセージを指定キューに送信するときの完了ステータスであり、MQ pub/sub ブローカのステータスではありません。MQ pub/sub ブローカのステータスを取得するには、replyToQueue を指定してから、request メッセージを送信するか negativeActionReport を要求します。MQ pub/sub ブローカは response または report MQRFH メッセージを replyToQueue に送信します。どちらの場合でも、replyToQueue からの response または report メッセージを明示的に読み取り、受信メッセージの MQPSCmpCode、MQPSReason、MQPSReasonText の各プロパティをチェックします。
- msgSegment または msgLastSegment を指定する場合に、アプリケーションがメッセージ (Adaptive Server アプリケーション以外のアプリケーションには MQGMO_COMPLETE_MSG、Adaptive Server アプリケーションには completeMsg=yes を指定) を読み取るときには、その論理メッセージを構成するメッセージをすべて作業単位で送信する必要があります。したがって、1 つのトランザクションにグループ化される必要があるメッセージをすべてまとめて送信します。

パーミッション

msgsend を実行するには、messaging_role が必要です。

msgsubscribe

説明 JMS のみ – 現在の Adaptive Server セッション用のトピックのサブスクリプションを作成するための SQL インタフェースを提供します。

構文

```
msg_subscribe ::= msgsubscribe
                (subscription_name)
                subscription_name ::= basic_character_expression
```

パラメータ

subscription_name
サブスクリプションを作成しているサブスクリプションの名前です。
basic_character_expression です。

例 “subscription_1” として登録されたトピックにパブリッシュされたメッセージの保持を開始するよう JMS メッセージング プロバイダに指示します。

```
select msgsubscribe ('subscription_1')
```

使用法

- **msgsubscribe** または **msgunsubscribe** を使用してサブスクリプションを指定する前に、サブスクリプションを **sp_msgadmin** に登録してください。次に示す例では、持続的サブスクリプション “subscription_1” を登録します。

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
            'my_jms_provider?topic=topic.sample,user=user1,password=pwd',
            'Supplier=12345', null, 'durable1', 'client1'
```

- **msgsubscribe** がいったん呼び出されると、指定トピックに関してパブリッシュされたセレクトの条件を満たすメッセージはすべて、現在の Adaptive Server セッションの間保持されます。メッセージを読み取るために **msgconsume** が呼び出されると、保持されなくなります。メッセージを消費する準備が整う前に到着したメッセージを保持しない場合には、**msgsubscribe** を呼び出さないでください。先に **msgsubscribe** を呼び出さずに **msgconsume** を呼び出す場合、**msgconsume** が呼び出されると、サブスクリプションが開始されます。
- **msgsubscribe** はクライアントがメッセージを受信するためのサブスクリプションを開始します。このメッセージは、*subscription_name* によって指定された終了ポイントとフィルタによって定義されます。成功の場合は 0、失敗の場合は 1 を返します。
- 次の例では、アプリケーション・ロジックがメッセージを読み取る準備が整う前に使用される **msgsubscribe** を示します。このメッセージにより、JMS クライアントはメッセージを保持するよう強制されます。アプリケーションは次の項目のサブスクリプションを作成します。

```
select msgsubscribe ('subscription_1')
```

クライアントは、同じ1つのメッセージを複数回消費し、メッセージングに関係しない他のアプリケーション・ロジックを使用します。すると、メッセージを読み取る準備が整い、`msgsubscribe` が呼び出されてから到着したメッセージをすべて受信します。

```
select msgconsume('subscription_1')
select msgconsume('subscription_1')
```

クライアント・アプリケーションは、このサブスクリプションを終了し、サブスクリプションを解除します。

```
select msgunsubscribe('subscription_1')
```

msgunsubscribe

説明 JMS のみ – 現在の Adaptive Server セッション用のトピックのサブスクリプションを解除するための SQL インタフェースを提供します。

構文

```
msg_unsubscribe ::= msgunsubscribe
                    (subscription_name [with {remove | retain}])
                    subscription_name ::= basic_character_expression
```

パラメータ

subscription_name
サブスクリプションを作成しているサブスクリプションの名前です。
basic_character_expression です。

with {remove | retain}
JMS メッセージ・プロバイダから持続的サブスクリプションを削除するか保持します。

例 “subscription_1” として登録されたトピックにパブリッシュされたメッセージの保持を停止するよう JMS メッセージング プロバイダに指示します。

```
select msgunsubscribe('subscription_1')
```

使用法

- **msgsubscribe** または **msgunsubscribe** を使用してサブスクリプションを指定する前に、サブスクリプションを **sp_msgadmin** に登録してください。次に示す例では、持続的サブスクリプション “subscription_1” を登録します。

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
            'my_jms_provider?topic=topic.sample,user=user1,password=pwd',
            'Supplier=12345', null, 'durable1', 'client1'
```

- **msgunsubscribe** は **subscription_name** によって指定された終了ポイントとフィルタに対する現在の Adaptive Server セッションの現在のサブスクリプションを停止します。成功の場合は 0、失敗の場合は 1 を返します。
- **with retain** が指定された場合、別のサブスクリプションが接続できるようにするため、JMS メッセージング・プロバイダへの接続は終了します。その際に、サブスクリプションで指定された同じサブスクライバ **client_id** が使用されます。持続的サブスクライバは、Adaptive Server と JMS メッセージ・プロバイダ内で定義されたままになります。**with remove** が指定されると、持続的サブスクライバ定義は JMS メッセージ・プロバイダから削除されます。デフォルト値は **with retain** です。

ユーザが Adaptive Server からログ・アウトすると、その Adaptive Server セッションのサブスクリプションはすべて解除されます。効果は、**with retain** オプションを使用して **msgunsubscribe** を実行するときと同じです。

`with remove` を使用して持続的サブスクリプションを解除すると、サブスクライバ定義が JMS メッセージ・プロバイダから削除され、サブスクライバ定義によって保持されていたメッセージがすべて読み取れなくなります。

```
<login>
select msgsubscribe('subscription_1')
select msgconsume('subscription_1')
...
select msgconsume('subscription_1')
select msgunsubscribe('subscription_1' WITH REMOVE)
<logout>

----Messages published to the topic registered as subscription_1 are no
----longer held by the JMS provider

<login>
select msgsubscribe('subscription_1')
select msgconsume('subscription_1')
...
select msgconsume('subscription_1')
select msgunsubscribe('subscription_1' WITH REMOVE)
```

別のシナリオでは、別のセッションがメッセージを消費できるようにするために、SQL セッションがサブスクリプションを解放します。次に示す例では、セッション 1 がサブスクリプションを解放し、セッション 2 がそこから消費できるようになる様子を示します。

表 3-22: SQL セッション

セッション 1	セッション 2
<pre>select msgunsubscribe ('subscription_1' WITH RETAIN) selectmsgconsume ('subscription_1') ... selectmsgconsume ('subscription_1') select msgunsubscribe ('subscription_1' WITH RETAIN)</pre>	<pre>select msgsubscribe('subscription_1') select msgconsume('subscription_1') ... select msgconsume('subscription_1') select msgunsubscribe('subscription_1' WITH RETAIN)</pre>

- 次の例では、アプリケーション・ロジックがメッセージを読み取る準備が整う前に使用される `msgsubscribe` を示します。このメッセージにより、JMS クライアントはメッセージを保持するよう強制されます。アプリケーションは次の項目のサブスクリプションを作成します。

```
select msgsubscribe ('subscription_1')
```

クライアントは、同じ1つのメッセージを複数回消費し、メッセージングに関係しない他のアプリケーション・ロジックを使用します。すると、メッセージを読み取る準備が整い、`msgsubscribe` が呼び出されてから到着したメッセージをすべて受信します。

```
select msgconsume('subscription_1')
select msgconsume('subscription_1')
```

クライアント・アプリケーションは、このサブスクリプションを終了し、サブスクリプションを解除します。

```
select msgunsubscribe('subscription_1')
```

endpoint

説明

(MQ) WebSphere MQ 用の *endpoint* の一般的構文と処理を指定します。個々のオプションについては、*endpoint* 引数を受け入れる関数とストアド・プロセス・プロシージャで説明してあります。

注意 JMS 終了ポイントは、Adaptive Server からは見えないので、正しさと有効性は検査されません。その代わりに、JMS プロバイダに直接送信されます。

構文

```

service_provider_uri ::= provider_name?qmgr=qmgr_name,destination
                        provider_name ::= local_name | full_name
                        local_name ::= identifier
                        full_name ::= service_provider_class:service_provider_url
                        service_provider_class ::= ibm_mq
                        service_provider_url ::= [channel/]tcp/hostname(port)
                        channel ::= channel_name[(channel_security)]
                        channel_name ::= identifier
                        channel_security ::= ssl:SSLCIPH=channel_ciph
                        channel_ciph ::= identifier
                        hostname ::= identifier
                        port ::= integer

qmgr_name ::= identifier
destination ::= [remote_qmgr,]queue=queue_name
remote_qmgr ::= remote_qmgr=remote_qmgr_name
remote_qmgr_name ::= identifier
queue_name ::= identifier

```

パラメータ

local_name

登録されたパブリッシャかサブスクライバの名前。

qmgr_name

MQ キュー・マネージャの名前。MQ では、キュー・マネージャ名の長さは 48 文字 (バイト) に制限されます。

ibm_mq

サービス・プロバイダ・クラスを定義します。大文字でも小文字でもかまいません。

channel_name

Adaptive Server 15.0.2 ESD #1 とそれ以降ではオプションであり、MQ サーバ接続チャネルの名前です。MQ では、チャネル名の長さは 20 文字 (バイト) に制限されます。*channel_name* が定義されない場合、Active Messaging はサーバ接続チャネル “SYSTEM.DEF.SRVCONN” を使用してキュー・マネージャに接続します。

channel_security

チャネルのセキュリティ・プロパティ。*channel_security* が指定されない場合、Adaptive Server はセキュリティ・プロトコルなしで WebSphere MQ と通信します。*channel_security* の有効な値は、*ssl* です。

channel_ciph

channel_security と併用され、サーバ接続チャンネルの SSLCIPH プロパティ値を指定します。WebSphere MQ クライアントにとって有効な **CipherSpec** 値でなければなりません。channel_ciph の有効な値は、次のとおりです。

表 3-23: channel_ciph の有効な CipherSpec 名

CipherSpec 名	ハッシュ・アルゴリズム	暗号化アルゴリズム	暗号化ビット数
NULL_MD5 ¹	MD5	なし	0
NULL_SHA ¹	SHA	なし	0
RC4_MD5_EXPORT ¹	MD5	RC4	40
RC4_MD5_US ²	MD5	RC4	128
RC4_SHA_US ²	SHA	RC4	128
RC2_MD5_EXPORT ¹	MD5	RC2	40
DES_SHA_EXPORT ¹	SHA	DES	56
RC4_56_SHA_EXPORT1024 ^{3,4,5}	SHA	RC4	56
DES_SHA_EXPORT1024 ^{3,4,5,6}	SHA	DES	56
TRIPLE_DES_SHA_US ⁴	SHA	3DES	168
TLS_RSA_WITH_AES_128_CBC_SHA ⁷	SHA	AES	128
TSL_RSA_WITH_AES_256_CBC_SHA ⁷	SHA	AES	256
AES_SHA_US ⁸	SHA	AES	128

1 OS/400 では、AC2 か AC3 がインストールされていると使用できる。

2 OS/400 では、AC3 がインストールされている場合のみ使用できる。

3 z/OS には使用できない。

4 OS/400 には使用できない。

5 1024 ビットのハンドシェイク・キー・サイズを指定する。

6 Windows には使用できない。

7 Intel プラットフォームの AIX、HP-UX、Linux のみに使用できる。

8 OS/400、AC3 のみに使用できる。

tcp

転送プロトコルであり、大文字でも小文字でもかまいません。SSL を介して MQ と通信するには、**tcp** を指定します。

hostname

MQ リスナが実行されているマシンのホスト名。

port

MQ リスナが受信に使用しているポート番号。

注意 **hostname(port)** の合計の長さは、264 バイトを超えることはできません。

queue_name

MQ キューの名前。MQ では、キュー名の長さは 48 文字 (バイト) に制限されます。

remote_qmgr_name

ターゲット・キュー定義を格納するリモート MQ キューの名前。MQ では、キュー・マネージャ名の長さは 48 文字 (バイト) に制限されます。使用法は次のとおりです。

- **msgsend** – このオプションが省略された場合、キュー・オブジェクトの位置を特定するためにローカル・キュー・マネージャが使用される。クラスタ・キューの作業負荷を分散させるには、このオプションを省略してください。
- **msgreceive** – Adaptive Server はこのオプションを無視する。

JMS サポートとは異なり、終了ポイントでユーザ名とパスワードを指定することはできません。MQ は関連する OS ログインの権限をチェックします。[「MQ のセキュリティ」\(35 ページ\)](#) を参照してください。

例

例 1 メッセージをキュー・マネージャに送信します。この場合、通信は SSL 対応 CHI チャネルで行われ、暗号スイートは NULL_MD5 です。

```
select msgsend('e',
  'ibm_mq:CH1(ssl:sslciph=NULL_MD5)/tcp/linuxxml1:1105?qmgr=MASTER_QM1,
  queue=Q2')
```

例 2 メッセージ “hello world 1” をローカル・キューに送信します。このキューは、MQ がインストールされるとキュー・マネージャ上で使用できるようになります。

```
select msgsend('hello world 1',
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,
  queue=SYSTEM.DEFAULT.LOCAL.QUEUE')
```

例 3 メッセージ “hello world 2” をキューに送信します。

```
select msgsend('hello world 2',
  'ibm_mq:channel2/tcp/host2(5678)?qmgr=QM2,
  queue=SYSTEM.DEFAULT.QUEUE')
```

例 4 メッセージ “hello world 3” をキューに送信します。

```
select msgsend('hello world 3',
  'ibm_mq:channel2/tcp/host2(5678)?qmgr=QM2,
  remote_qmgr=QM3,queue=QM3.Q')
```


option_string

説明	<i>option_string</i> の一般的な構文と処理を指定します。個々のオプションについては、そのオプションを参照する関数で説明しています。
構文	<pre> option_string ::= basic_character_expression option_string_value ::= option_and_value [[,] option_and_value] option_and_value ::= option_name = option_value option_name ::= simple_identifier option_value ::= simple_identifier quoted_string integer_literal float_literal byte_literal true false null </pre>
パラメータ	<p><i>option_string</i> 指定するオプションについて説明する文字列。</p> <p><i>simple_identifier</i> <i>option</i> の値を特定する文字列。</p> <p><i>quoted_string</i> 埋め込み引用符に関する通常の SQL 表記規則を使用した文字列。</p> <p><i>integer_literal</i> 通常の SQL 表記規則によって指定されたりテラル。</p> <p><i>float_literal</i> 通常の SQL 表記規則によって指定されたりテラル。</p> <p><i>true</i> ブール・リテラル。</p> <p><i>false</i> ブール・リテラル。</p> <p><i>null</i> null リテラル。</p> <p><i>byte_literal</i> 0xHH 形式 (ただし各 H は 16 進数)。</p>
使用法	<i>option_string</i> の使用法については、 「msgsend」(99 ページ) を参照してください。

sisespec

説明 (MQ のみ) *size* を受け入れるプロパティ値とメッセージ・オプションは、サイズの指定として次の構文を受け入れます。

構文 `sisespec ::= integer_number [sisespec_units]`
`sisespec_units ::= { M | K }`

パラメータ *integer_number*
サイズ。
K または k
キロバイト数。

M または m
メガバイト数。

sisespec_units
メガバイト (M)、キロバイト (K)、またはバイトを単位とするサイズ指定。
sisespec_units を指定しない場合、デフォルトはバイト単位です。

例 **例 1** 100MB のサイズ指定を示します。

```
-- Specify buffer length to be 100 megabytes
select msgrecv('ibm_mq:channel1/tcp/host1(5678)?'
  + 'qmgr=QM1,queue=SYSTEM.DEFAULT.LOCAL.QUEUE'
  option 'bufferLength=100M')
```

例 2 300KB のサイズ指定を示します。

```
-- Specify buffer length to be 300 kilobytes
select msgrecv(
  'ibm_mq:channel2/tcp/host2(5678)?qmgr=QM2,remote_qmgr=QM3,queue=QM3.Q'
  option 'bufferLength=300K')
```

例 3 (MQ) 1MB のサイズ指定を示します。

```
-- bufferLength specified as 1 megabyte
select msgrecv(
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=DEFAULT.QUEUE'
  option 'bufferLength=1M')
```

例 4 (MQ) 10KB のサイズ指定を示します。

```
-- bufferLength specified as 10K
select msgrecv(
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=DEFAULT.QUEUE'
  option 'bufferLength=10K')
```

timespec

説明 `timespec` 関数を使用する時間間隔を受け入れるプロパティ値とメッセージ・オプションは、MQ と JMS の両方の時間指定として次の構文を受け入れます。

構文

```
'timeout=timespec'
      timespec ::= integer_number [ timespec_units ]
      timespec_units ::= { dd | hh | mi | ss | ms }
```

パラメータ

dd
日数。

hh
時間数。

mi
分数。

ss
秒数。

ms
ミリ秒数。

timespec_units
ミリ秒数。*timespec_units* を指定しない場合、デフォルトはミリ秒単位です。

例 **例 1** 100 日の時間指定を示します。

```
-- timeout specified as 100 days
select msgrecv('ibm_mq:channel2/tcp/host2(5678)?'
  + 'qmgr=QM2,remote_qmgr=QM3,queue=QM3.Q'
  option 'timeout=100dd')
```

例 2 300 分の時間指定を示します。

```
-- timeout specified as 300 minutes
select msgrecv('ibm_mq:channel1/tcp/host1(5678)?'
  + 'qmgr=QM1,queue=SYSTEM.DEFAULT.LOCAL.QUEUE'
  option 'timeout=300mi')
```

例 3 1,024 ミリ秒の時間指定を示します。

```
-- timeout specified as 1,024 milliseconds
select msgrecv(
  'ibm_mq:channel2/tcp/host2(5678)?'
  + 'qmgr=QM2,queue=SYSTEM.DEFAULT.LOCAL.QUEUE'
  option 'timeout=1024ms')
```

例 4 (MQ) 30 秒の時間指定を示します。

```
-- timeout specified as 30 seconds
select msgrecv(
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=DEFAULT.QUEUE'
  option 'timespec=30ss')
```

例 5 (JMS) 30 分の時間指定を示します。

```
-- timeout specified as 30 minutes
select msgrecv(
  'tibco)_jms:tcp://localhost:7222?queue=queue.sample'
  option 'timeout=30mi')
```

参照

[msgconsume](#), [msgpublish](#), [msgrecv](#), [msgsend](#)

この章では、Active Messaging オプションと共に配布されたメッセージング機能の例を示すサンプル・コードについて説明します。

トピック名	ページ
Sybase ディレクトリ	145
SQL を使用したコード・サンプルの使用	146
Java/JDBC を使用したコード・サンプルの使用	146

Sybase ディレクトリ

SYBASE ディレクトリには、次の 3 つのサブディレクトリが含まれています。

- *functionstring* – デフォルトの SQL テンプレートをメッセージング・システムへの呼び出しに変換するための Replication Server 関数文字列を生成するスクリプト。
- *sql* – Active Messaging を使用したサンプルが付いた SQL スクリプト。
- *jdbc* – Active Messaging を使用した JDBC のサンプル。

コード・サンプルは、`$SYBASE/$SYBASE_ASE/samples/messaging` ディレクトリにあります。

各サブディレクトリには、*README* ファイルが入っています。このファイルでは、各コード・サンプルの目的を説明し、コードを実行する手順と必要なインストール手順を示しています。

Windows とその他のプラットフォームでは、オペレーティング・システム・ファイルの名前は同じではありません。たとえば、Windows プラットフォームにある *queue_listener.bat* というファイルは、UNIX/Linux プラットフォームでは単に *queue_listener* になります。

SQL を使用したコード・サンプルの使用

`$$SYBASE/$SYBASE_ASE/samples/messaging/sql`にあるコード・サンプルは、カスタマイズしたメッセージをメッセージング・システムにパブリッシュするように SQL (ストアド・プロシージャ、トリガなど) を記述または変更する方法について説明しています。

これらのサンプルでは、SQL コードを使用して、メッセージ・バスからメッセージを使用する方法についても説明しています。ここでは、メッセージングのパーティシパントとして、またメッセージ・バスを使用するアプリケーションとして Adaptive Server を使用しています。

Java/JDBC を使用したコード・サンプルの使用

`$$SYBASE/$SYBASE_ASE/samples/messaging/jdbc`にあるコード・サンプルは、カスタマイズしたメッセージをメッセージング・システムにパブリッシュするように Java コードを記述または変更する方法について説明しています。

これらのサンプルでは、メッセージ・バスからメッセージを使用する Java コードについても説明しています。ここでは、メッセージングのパーティシパントとして、またメッセージ・バスを使用するアプリケーションとして Adaptive Server を使用しています。

用語解説

この用語解説では、このマニュアル全体で使用される JMS および MQ に関連する用語の定義を説明します。

CipherSpec	認証の完了後に SSL メッセージに適用される WebSphere MQ の暗号化アルゴリズムとハッシュ関数の組み合わせ。
cluster	論理的に関連付けられているキュー・マネージャのネットワーク。
JMS	Java Message Service。
MQ	WebSphere MQ。IBM が提供しているメッセージ指向のミドルウェア。
MQ パブリッシュ/サブスクライブ (MQ publish/subscribe)	WebSphere MQ のパブリッシュ関数とサブスクライブ関数
MQI	WebSphere MQ メッセージ・キュー・インタフェース・プログラミング API。
MQM	キューを管理する WebSphere MQ メッセージ・キュー・マネージャ・プロセス。
RF ヘッダー (RF header)	MQ パブリッシュ/サブスクライブで使用される WebSphere MQ ルールとフォーマット・ヘッダー。MQ パブリッシュ/サブスクライブ・ブローカまたはストリーム・キューに送信されたすべてのメッセージには、RF ヘッダーが必要です。RF ヘッダーには、MQ パブリッシュ/サブスクライブ・ブローカに渡す制御情報が含まれています。MQ パブリッシュ/サブスクライブ・メッセージには、メッセージ・ペイロードに RF ヘッダーが含まれており、その後アプリケーション・データが続きます。
RFH	WebSphere MQ ルールとフォーマット・ヘッダー。そのメッセージのルールとフォーマット情報を提供するメッセージ・ヘッダーの部分です。
SSL	保護されていないネットワークを介して安全にデータを転送するための業界標準プロトコル。
暗号化 (encryption)	プレーン・テキストのメッセージを暗号テキストに変換するプロセス。
暗号スイート (cipher suite)	SSL 接続で使用される一連の暗号化アルゴリズム。
暗号法 (cryptography)	判読できるテキスト (プレーン・テキスト) と判読できない形式 (暗号テキスト) を変換するプロセス。
キー・レポジトリ (key repository)	デジタル証明書およびそれに関連付けられたプライベート・キーの格納場所。

キュー (queue)	JMS でポイント間のメッセージングに使用されるドメイン。 WebSphere MQ で送信済みメッセージを格納するオブジェクト。
共有キー (shared key)	対称アルゴリズムで使用される同一のシークレット・キー。
クラスタ・キュー・マネージャ (cluster queue manager)	クラスタを構成している WebSphere MQ キュー・マネージャ。
クラスタ・キュー (cluster queue)	クラスタ・キュー・マネージャによってホストされ、クラスタ内の他のキュー・マネージャが使用できる WebSphere MQ キュー。
サービス・プロバイダ (service provider)	Tibco JMS メッセージ・プロバイダ。たとえば、Tibco JMS はサービス・プロバイダです。このマニュアルではメッセージング・プロバイダと呼んでいます。
サブスクライバ (subscriber)	WebSphere MQ におけるパブリッシュ/サブスクライブ・トポロジ内の受信者。
サブスクリプション (subscription)	1 対多のメッセージングをパブリッシュまたは使用するための Tibco JMS ドメイン。
持続的サブスクリプション (durable subscription)	クライアント接続が切断されている間にメッセージを保持する JMS サブスクリプション。
ストリーム (stream)	WebSphere MQ での関連する MQ トピックのグループ化。
対称型アルゴリズム (symmetric algorithms)	両当事者が同一のシークレット・キーを使用する必要がある暗号化アルゴリズム。
チャンネル (channel)	論理通信リンクである WebSphere MQ オブジェクト。
デジタル証明書 (digital certificate)	なりすまし攻撃に対して保護します。デジタル証明書はその所有者にパブリック・キーをバインドします。その所有者が個人、キュー・マネージャ、その他のエンティティのいずれの場合でも有効です。
トピック (topic)	Tibco JMS ではキューとほぼ同じですが、1 対多のメッセージングに使用されます。 WebSphere MQ では、パブリケーションのサブジェクトです。WebSphere MQ のパブリッシュ/サブスクライブ・トピックと JMS のトピックは異なります。JMS では、トピックはパブリッシュ/サブスクライブの終了ポイントです。それに対して WebSphere MQ のパブリッシュ/サブスクライブ・トピックと JMS のトピックは異なります。
パブリケーション (publication)	WebSphere MQ でパブリッシャによって送信される情報。
パブリック・キー (public key)	非対称アルゴリズムにおいて公開できるシークレット・キー。
パブリッシャ (publisher)	WebSphere MQ におけるパブリッシュ/サブスクライブ・モデル内の送信者。

非持続的サブスクリプション (nondurable subscription)	クライアントが接続している間のみメッセージを保持する JMS サブスクリプション。
非対称型アルゴリズム (asymmetric algorithms)	暗号化と復号化にそれぞれ異なるキーを使用する暗号化アルゴリズム。一方のキーは秘密にする必要がありますが、もう一方のキーは公開できます。
復号化 (decryption)	暗号テキストのメッセージをプレーン・テキスト形式に変換するプロセス。
フル・レポジトリ (full repository)	クラスタ内のすべてのキュー・マネージャに関する情報の完全な集合をホストする WebSphere MQ キュー・マネージャ。
部分レポジトリ (partial repository)	WebSphere MQ では、クラスタ内のキュー・マネージャを指します。フル・レポジトリ内の情報について問い合わせ、独自のサブセットを構成します。
ブローカ (broker)	パブリッシュ/サブスクライブ・モデル内でサブスクリプション解析を実行する WebSphere MQ プロセス。
プライベート・キー (private key)	非対称アルゴリズムにおいて秘密にする必要があるシークレット・キー。
ペイロード (payload)	WebSphere MQ メッセージの本文。
メッセージング・クライアント (messaging client)	メッセージを生成または使用する JMS アプリケーション。
リモート・キュー・マネージャ (remote queue manager)	WebSphere MQ でアプリケーションが接続しているキュー・マネージャとは別のキュー・マネージャ。
ローカル・キュー・マネージャ (local queue manager)	アプリケーションが接続している WebSphere MQ キュー・マネージャ。

索引

記号

- \$\$SYBASE ディレクトリ内の *functionstring* サブディレクトリ 145
- \$\$SYBASE ディレクトリ内の *jdbc* サブディレクトリ 145
- \$\$SYBASE ディレクトリ内の *sql* サブディレクトリ 145
- @@msgcorrelation メッセージング・グローバル変数 40
- @@msgheader メッセージング・グローバル変数 40
- @@msgid メッセージング・グローバル変数 43
- @@msgproperties メッセージング・グローバル変数 43
- @@msgreplymgr メッセージング・グローバル変数 44
- @@msgreplytoinfo メッセージング・グローバル変数 44
- @@msgschema メッセージング・グローバル変数 45
- @@msgstatus メッセージング・グローバル変数 45
- @@msgstatusinfo メッセージング・グローバル変数 45
- @@msgtimestamp メッセージング・グローバル変数 46
- @@。「グローバル変数」参照

A

- Adaptive Server 固有のメッセージ・プロパティ 48
- Adaptive Server ホスト・マシンへの MQ クライアントのインストール 36
- ASE メッセージ・タイプ 96
- ASE_MSBODY_SCHEMA メッセージ・プロパティ 48
- ASE_MSGBODY メッセージ・プロパティ 48
- ASE_ORIGIN メッセージ・プロパティ 48
- ASE_RTMS_CHARSET メッセージ・プロパティ 48
- ASE_RTMS_VERSION メッセージ・プロパティ 48
- ASE_SPID メッセージ・プロパティ 48
- ASE_TIMESTAMP メッセージ・プロパティ 49
- ASE_VERSION メッセージ・プロパティ 49
- ASE_VERSION_FORMATS メッセージ・プロパティ 49

B

- byte メッセージ・タイプ 96

C

- CipherSpec、定義 147

E

- endpoint 構文セグメント 138–140
 - 構文 138
 - 説明 138
 - パラメータ 138
 - 例 140

J

- Java/JDBC、コード・サンプルの使用 146
- JMS
 - msgrecv *option_string* の値 95
 - msgrecv *property_option_clause* 値 120
 - msgsend *option_string* 値 108
 - キューの説明 3
 - キュー、読み込んだメッセージ 8
 - 定義 147
 - メッセージ・セクタ 5
 - メッセージ・バス 2
 - メッセージ・プロパティ 4
- JMS トピックからのメッセージのコンシューム 8
- JMS トピックからのメッセージのパブリッシュ 8
- JMS トピック、メッセージのパブリッシュとコンシューム 8

M

message

メッセージ・タイプ 96

message header キーワード 49

message property キーワード 49

message selector キーワード 49

MQ

Adaptive Server ホスト・マシンへのクライアントの
インストール 36

MQ クライアントの共有ライブラリ 36

msgrecv option および option_string の値 89

msgrecv 関数、使用法 96

msgsend option_string 値 108

msgsend property_option_clause 値 111

RF ヘッダー 5

rfhCommand が deletePublications に設定された
場合の msgsend プロパティ 121

エイリアス・キュー・オブジェクトの説明 11

概要 11

クエリ 37

クラスター・キュー・オブジェクトの説明 12

権限 37

セキュリティ 35

定義 147

トピックの構文 18

パブリッシャとサブスクライバの ID 19

パブリッシュ/サブスクライブの例 20

パブリッシュ/サブスクライブ・プロセスの例 16

パブリッシュ/サブスクライブ・メッセージング・
モデル 3

パブリッシュとサブスクライブの説明 15

ブローカ・コマンド・キュー 4

メッセージ・キュー・インタフェース (MQI) 11

メッセージ・タイプ 13

メッセージのグループ分け 13

メッセージ・プロパティ 5

メッセージ・ヘッダの説明 13

メッセージ本文の説明 13

メッセージング・モデル 3

モデル・キュー・オブジェクトの説明 11

リモート・キュー・オブジェクトの説明 11

ローカル・キュー・オブジェクトの説明 11

MQ キュー・マネージャ、接続 35

MQ クライアント共有ライブラリとディレクトリ 36

MQ クライアントの共有ライブラリ 36

MQ でのブローカ・コマンド・キュー 4

MQ のトピックの構文 18

MQ のパブリッシャとサブスクライバの ID 19

MQ パブリケーション/サブスクリプション
プロセス 17

MQ パブリッシュ/サブスクライブ、定義 147

MQ メッセージ・セレクタ 5

MQI (メッセージ・キュー・インタフェース) 11

説明 11

定義 147

MQM、定義 147

msgconsume 関数 8, 71–73

構文 71

サポートされていないメッセージのデータ型 73

サポートされているメッセージ・タイプ 73

使用法 72

説明 71

パラメータ 71

呼び出し、結果 73

例 72

msgheader XML ドキュメント 47–48

msgpropcount 関数 9, 74

msgproperties XML ドキュメント 47–48

説明 47

msgproplist 関数 9, 75–76

msgpropname 関数 9, 77

msgproptype 関数 9, 78–79

msgpropvalue 関数 9, 80

msgpublish option_string 値 83

msgpublish properties_clause 値 83

msgpublish 関数 8, 81–84

msgrecv 関数 85–98

MQ の使用法 96

構文 85

使用法 95

パーミッション 98

パラメータ 85

msgsend 関数 99–132

JMS option_string 値 108

JMS property_option_clause 値 120

MQ option_string 値 108

MQ property_option_clause 値 111

rfhCommand が deletePublications に設定された
場合の msgsend プロパティ 121

構文 99

使用法 107–132

説明 99

トランザクションの動作 33

パーミッション 132

パラメータ 99–102

例 102–106

msgsubscribe 関数 8, 133
msgunsubscribe 関数 8, 135

O

option_string 構文セグメント 141

R

requeue、誤ったメッセージ・タイプの使用 96
RF ヘッダー 5
定義 147
RFH、定義 147
RFH、「RF ヘッダー」参照
rtrim 関数 46

S

set transactional messaging コマンド 34
sizespec 構文セグメント 142
sp_configure 'enable real time messaging' スタアド・
プロシージャ 51-52
sp_engine スタアド・プロシージャ 55-58
構文 55
使用法 57
説明 55
パーミッション 58
パラメータ 55
例 56
sp_msgadmin スタアド・プロシージャ 8, 59-70
構文 59
使用法 69
説明 59
パーミッション 70
パラメータ 64
例 68
sp_msgadmin と MQSeries 50
SQL
関数、説明 50
コード・サンプルの使用 146
トランザクションのコマンド 33
メッセージ・プロパティの関数 9
SSL、定義 147

T

text メッセージ・タイプ 96
TIBCO EMS
メッセージ・バス 2
timespec 構文セグメント 143
構文 143
説明 143
パラメータ 143
例 143
transactional messaging full キーワード 49
transactional messaging none キーワード 49
transactional messaging simple キーワード 49
Transact-SQL、メッセージの送信 7

U

URL
Apache log4j ロギング・サービス 60
IBM Publication Center 38
Sun の Java 61

W

with remove キーワード 49
with retain キーワード 49

X

XML ドキュメント
msgheader 47-48
msgproperties 47-48
使用法 48
説明 47
例 47

あ

暗号化、定義 147
暗号スイート、定義 147
暗号法、定義 147

索引

え

エイリアス・キュー・オブジェクト (MQ) 11

お

オプション文字列 50

か

関数

msgconsume 71-73

msgpropcount 74

msgproplist 75-76

msgpropname 77

msgproptype 78-79

msgpropvalue 80

msgpublish 81-84

msgrecv 85-98

msgsend 99-132

msgsubscribe 133

msgunsubscribe 135

関数のリスト 50

後続ブランクを削除するための rtrim 46

説明 50

リスト 50

き

キー・レポジトリ、定義 147

キーワード 49

message header 49

message property 49

message selector 49

transactional messaging full 49

transactional messaging none 49

transactional messaging simple 49

with remove 49

with retain 49

キュー

1 対 1 のメッセージング 148

定義 148

メッセージの送受信 7

キューとトピックの作成 50

キューとトピック、作成、削除、アクセス 50

キュー・マネージャの説明 11

キュー・マネージャ、接続 35

共有キー、定義 148

く

組み込み。個々の関数名を含む関数を参照

クラスタ・キュー・オブジェクト (MQ) 12

クラスタ・キュー・マネージャ、定義 148

クラスタ・キュー、定義 148

クラスタ、定義 147

グローバル変数

@@msgcorrelation 40

@@msgcreplyqmgr 44

@@msgheader 40

@@msgid 43

@@msgmsgschema 45

@@msgproperties 43

@@msgreplytoinfo 44

@@msgstatus 45

@@msgstatusinfo 45

@@msgtimestamp 46

使用法 46

設定 40

例 46

こ

後続ブランク、rtrim を使用した削除 46

構文

endpoint 構文セグメント 138

msgconsume 関数 71

msgpropcount 関数 74

msgrecv 関数 85

msgsubscribe 関数 133

msgunsubscribe 関数 135

option_string 構文セグメント 141

sizespec 構文セグメント 142

sp_configure 'enable real time messaging' スタアド・
プロシージャ 51

sp_engine スタアド・プロシージャ 55

sp_msgadmin スタアド・プロシージャ 59

timespec 構文セグメント 143

構文セグメント

endpoint 138-140

option_string 141

sizespec 142

timespec 143

リスト 51

コード・サンプル

Java/JDBC を使用 146

SQL を使用 146

国際化 32-33

さ

- サービス・プロバイダ 148
- サブスクリバ、定義 148
- サブスクリプション
 - 持続的 3
 - 定義 148
 - 非持続的 3
- サンプル 145
 - Sybase ディレクトリ 145
 - 概要 145

し

- 持続的サブスクリプション 3
 - 定義 148
- 使用法
 - msgconsume 関数 72
 - msgrecv 関数 95
 - msgsend 関数 107–132
 - msgsend 構文 99
 - msgsubscribe 関数 133
 - msgunsubscribe 関数 135
 - option_string 構文セグメント 141
 - sp_configure 'enable real time messaging' ストアド・プロシージャ 52
 - sp_engine ストアド・プロシージャ 57
 - sp_msgadmin ストアド・プロシージャ 69
 - XML ドキュメント 48
 - グローバル変数 46

す

- ストアド・プロシージャ
 - sp_configure 'enable real time messaging' 51–52
 - sp_engine 55–58
 - sp_msadmin 'show' を使用するための WebSphere MQ の準備 38
 - sp_msgadmin 59–70
 - リスト 50
- ストリーム、定義 148

せ

- セキュリティと MQ 35
- 説明
 - endpoint 構文セグメント 138
 - MQ 11
 - MQ エイリアス・キュー・オブジェクト 11
 - MQ クラスタ・キュー・オブジェクト 12
 - MQ メッセージ・ヘッダ 13
 - MQ メッセージ本文 13
 - MQ モデル・キュー・オブジェクト 11
 - MQ リモート・キュー・オブジェクト 11
 - MQ ローカル・キュー・オブジェクト 11
 - msgconsume 関数 71
 - msgheader XML ドキュメント 47
 - msgproperties XML ドキュメント 47
 - msgsend 関数 99
 - msgsubscribe 関数 133
 - msgunsubscribe 関数 135
 - option_string 構文セグメント 141
 - sizespec 構文セグメント 142
 - sp_configure 'enable real time messaging' ストアド・プロシージャ 51
 - sp_engine ストアド・プロシージャ 55
 - sp_msgadmin ストアド・プロシージャ 59
 - timespec 構文セグメント 143
 - XML ドキュメント 47
 - キュー 148
 - キュー・マネージャ 11
 - 持続的サブスクリプション 3
 - チャンネル 148
 - 非持続的サブスクリプション 3
 - ブローカ 149
 - メッセージ指向のミドルウェア 1
 - メッセージングの概念 1

た

- 対称型アルゴリズム、定義 148

ち

- チャンネル、定義 148

索引

て

- ディレクトリ
 - functionstring* 145
 - jdbc* 145
 - sql* 145
- データ型の *binary* 値 96
- データ型、*binary* 値 96
- テキスト・メッセージング 28–29
- デジタル証明書、定義 148

と

- トピック
 - 定義 148
 - メッセージのパブリッシュとコンシューム 8
- トランザクション
 - SQL コマンド 33
 - コミット 33
 - データベース、メッセージの影響 33
- トランザクション動作、*set transactional messaging* で制御 34

は

- パーミッション
 - msgrecv* 関数 98
 - msgsend* 関数 132
 - sp_engine* ストアド・プロシージャ 58
 - sp_msgadmin* ストアド・プロシージャ 70
- バイト順 96
- パブリケーション、定義 148
- パブリック・キー、定義 148
- パブリッシャ、定義 148
- パブリッシュ/サブスクライブ
 - JMS メッセージング・モデル 3
 - MQ メッセージング・モデル 3
 - 説明 15
 - メッセージング・モデル 3
 - 例 20
- パラメータ
 - endpoint* 構文セグメント 138
 - msgconsume* 関数 71
 - msgpropcount* 関数 74
 - msgrecv* 関数 85
 - msgsend* 関数 99–102
 - msgsubscribe* 関数 133

- msgunsubscribe* 関数 135
- option_string* 構文セグメント 141
- sizespec* 構文セグメント 142
- sp_configure 'enable real time messaging'* ストアド・プロシージャ 51
- sp_engine* ストアド・プロシージャ 55
- sp_msgadmin* ストアド・プロシージャ 64
- timespec* 構文セグメント 143

ひ

- 非持続的サブスクリプション 3, 149
- 非対称型アルゴリズム、定義 149
- 非同期メッセージング 2
- 表
 - @@msgheader* グローバル変数のフィールドと説明 40
 - Adaptive Server 固有のメッセージ・プロパティ 48
 - channel_ciph* の有効な CipherSpec 名 139
 - JMS *@@msgheader* フィールドと説明 42
 - JMS の *msgrecv option_string* の値 95
 - JMS 用の *msgrecv property_option_clause* 値 120
 - JMS 用の *msgsend option_string* 値 108
 - MQ クライアント共有ライブラリとディレクトリ 36
 - MQ で *rfhCommand* が *deletePublications* に設定された場合の *msgsend* プロパティ 121
 - MQ の *msgrecv option* および *option_string* の値 89
 - MQ の原則/グループおよびその権限 37
 - MQ 用の *msgsend option_string* 値 108
 - MQ 用の *msgsend property_option_clause* 値 111
 - msgconsume* の *option* と *option_string* のパラメータ値 71
 - msgpublish option_string* 値 83
 - msgpublish properties_clause* 値 83
 - rfhCommand* が *deregisterPublisher* に設定された場合の *msgsend* プロパティ 122
 - rfhCommand* が *deregisterSubscriber* に設定された場合の *msgsend* プロパティ 123
 - rfhCommand* が *publish* に設定された場合の *send* プロパティ 124
 - rfhCommand* が *registerPublisher* に設定された場合の MQ *msgsend* プロパティ 127
 - rfhCommand* が *registerSubscriber* に設定された場合の MQ *msgsend* プロパティ 128

rfhCommand が requestUpdate に設定された場合の
 MQ msgsend プロパティ 131
 RTDS のダブル・キーワードとトリプル・
 キーワード 49
 有効な sp_msgadmin 'show' オプション、
 option_string のタイプと値 66

ふ

ファイル名、プラットフォームによって異なる 145
 復号化、定義 149
 部分レポジトリ、定義 149
 プライベート・キー、定義 149
 フル・レポジトリ、定義 149
 プレビュー、例 9
 プロローガ、定義 149
 プロバイダ、メッセージング 2

へ

ペイロード、定義 149

ほ

ポイントツーポイント・キュー
 JMS 3
 MQ 3
 ポイントツーポイント・メッセージング・モデル 3
 JMS 3
 MQ Series 3

め

メッセージ
 JMS キューから読み込み 8
 JMS での受信者 3
 JMS での送信者 3
 JMS のセレクトア 4
 JMS のプロパティ 4
 MQ のグループ分け 13
 MQ のセレクトア 5
 MQ のタイプ 13
 MQ のプロパティ 5
 Transact-SQL アプリケーションによる送信 7

インタフェース、プレビュー 9
 キューとの間の送受信 7
 トピックとのパブリッシュとコンシューム 8
 バス、TIBCO 2
 フォーマット 4
 プロパティ、使用 9
 ヘッダー 4
 本文 4
 メッセージ関連のグローバル変数 40
 メッセージ・キュー・インタフェース (MQI) の説明 11
 メッセージ・セレクトア 5
 メッセージ・タイプ
 msgconsume でサポートされている 73
 テキスト 7
 バイナリ 7
 メッセージの受信 7
 メッセージの送信 7
 メッセージ・フォーマット 4
 メッセージ・プロパティ 8
 Adaptive Server 固有 48
 ASE_MSBODY_SCHEMA 48
 ASE_MSGBODY 48
 ASE_ORIGIN 48
 ASE_RTMS_CHARSET 48
 ASE_RTMS_VERSION 48
 ASE_SPID 48
 ASE_TIMESTAMP 49
 ASE_VERSION 49
 ASE_VERSION_FORMATS 49
 JMS 4
 MQ 5
 「msgsend 関数」参照 8
 メッセージ・ヘッダー 4
 メッセージング
 概念 1
 モデル 3
 メッセージング・インタフェースのプレビュー 9
 メッセージング・クライアント、定義 149
 メッセージング・グローバル変数
 @@msgcorrelation 40
 @@msgheader 40
 @@msgid 43
 @@msgmsgschema 45
 @@msgproperties 43
 @@msgreplymgr 44
 @@msgreplytoinfo 44
 @@msgstatus 45
 @@msgstatusinfo 45
 @@msgtimestamp 46
 char データ型 46

索引

メッセージング・システム、非同期 2
メッセージング処理の実行、説明 2
メッセージングの概念 1
メッセージング・プロバイダ 2
 キューとトピックの作成、削除、アクセス 50
メッセージング・モデル
 JMS 3
 MQ 3
 MQ Series 定義 3
 MQ パブリッシュ/サブスクライブ 3
 パブリッシュ/サブスクライブ 3
 ポイントツーポイント 3

も

モデル・キュー・オブジェクト (MQ) 11
モデル、メッセージング 3

り

リモート・キュー・オブジェクト (MQ) 11
リモート・キュー・マネージャ、定義 149

る

ルールとフォーマット・ヘッダー。「RF ヘッダー」
 参照

れ

例

endpoint 構文セグメント 140
MQ パブリッシュ/サブスクライブ・プロセス 16
msgconsume 関数 72
msgheader XML ドキュメント 47
msgsend 関数 102-106
msgsubscribe 関数 133
msgunsubscribe 関数 135
sp_configure 'enable real time messaging' ストアド・
 プロシージャ 52
sp_engine ストアド・プロシージャ 56
sp_msgadmin ストアド・プロシージャ 68
timespec 構文セグメント 143
XML ドキュメント 47
グローバル変数 46
メッセージング・インタフェース 9

ろ

ローカル・キュー・オブジェクト (MQ) 11
ローカル・キュー・マネージャ、定義 149