



# BEA WebLogic Integration™

**B2B Integration**  
メッセージング  
アプリケーション  
プログラミング  
ガイド



**BEA**



**BEA**

## 著作権

Copyright © 2002, BEA Systems, Inc. All Rights Reserved.

## 限定的権利条項

本ソフトウェアおよびマニュアルは、**BEA Systems, Inc.** 又は日本ビー・イー・エー・システムズ株式会社（以下、「**BEA**」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができません。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、**BEA Systems, Inc.** からの書面による事前の同意なしに、複写、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、**BEA** の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また **BEA** による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、**BEA** は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

## 商標または登録商標

**BEA**、**Jolt**、**Tuxedo**、および **WebLogic** は **BEA Systems, Inc.** の登録商標です。**BEA Builder**、**BEA Campaign Manager for WebLogic**、**BEA eLink**、**BEA Manager**、**BEA WebLogic Commerce Server**、**BEA WebLogic Enterprise**、**BEA WebLogic Enterprise Platform**、**BEA WebLogic Express**、**BEA WebLogic Integration**、**BEA WebLogic Personalization Server**、**BEA WebLogic Platform**、**BEA WebLogic Server**、**BEA WebLogic Workshop** および **How Business Becomes E-Business** は、**BEA Systems, Inc** の商標です。

その他の商標はすべて、関係各社が著作権を有します。

## B2B Integration メッセージング アプリケーション プログラミング ガイド

パート番号	日付	ソフトウェアのバージョン
なし	2002年6月	7.0

---

# 目次

## このマニュアルの内容

対象読者.....	ix
e-docs Web サイト.....	x
このマニュアルの印刷方法.....	x
関連情報.....	x
サポート情報.....	xi
表記規則.....	xii

## 1. ビジネス メッセージ交換用 XOCP アプリケーションの開発

はじめに.....	1-1
主要な概念.....	1-3
XOCP アプリケーション.....	1-3
XOCP アプリケーション セッション.....	1-5
Messaging API クラス ライブラリ.....	1-5
XOCP ビジネス メッセージおよびメッセージ エンベロープ.....	1-6
XOCP ビジネス メッセージの図.....	1-7
XOCP ビジネス メッセージのコンポーネント.....	1-7
メッセージ エンベロープの情報フロー.....	1-8
会話開始者と会話参加者.....	1-10
会話コーディネータ.....	1-12
グローバル会話コーディネータ.....	1-12
ローカル会話コーディネータ.....	1-13
トレーディング パートナの状態.....	1-13
セキュアなメッセージング.....	1-14
XOCP アプリケーションの主要なタスク.....	1-14
XOCP アプリケーション セッションの作成.....	1-14
会話でのロールの登録.....	1-15
トレーディング パートナとの会話への関与.....	1-16
会話の開始とビジネス メッセージの送信.....	1-17
会話への参加.....	1-17

会話からの退出.....	1-18
会話の終了.....	1-18
XOCP アプリケーション セッションのシャットダウン .....	1-18
実行時の情報フロー .....	1-19
情報フローの図.....	1-20
情報フローでのステップ .....	1-21

## 2. XOCP アプリケーションのプログラミング ステップ

手順 1: パッケージのインポート .....	2-2
手順 2: MessageListener インタフェースの実装.....	2-2
手順 3: XOCP アプリケーション セッションの作成 .....	2-3
手順 4: メッセージ リスナの作成と登録.....	2-4
手順 5: 会話の開始または会話への参加.....	2-5
手順 6: ビジネス メッセージの交換 .....	2-5
手順 7: 会話の終了 .....	2-6
参加者が会話から退出する場合 .....	2-6
開始者が会話を終了する場合 .....	2-6
手順 8: XOCP アプリケーション セッションのシャットダウン .....	2-7

## 3. XOCP ビジネス メッセージの送信

手順 1: ビジネス メッセージの作成.....	3-1
必要なパッケージのインポート .....	3-2
ペイロード部分の作成.....	3-2
XML ドキュメントの作成 .....	3-2
添付ファイルの作成 .....	3-4
XOCP ビジネス メッセージの作成とペイロード部分の追加 .....	3-4
手順 2: ビジネス メッセージの受信側の指定 (省略可能) .....	3-5
特定のトレーディング パートナの指定 .....	3-6
XPath 式によるメッセージ受信側の基準の指定 .....	3-6
標準的なトレーディング パートナ属性の指定 .....	3-7
拡張プロパティを使用した XOCP XPath 式の指定.....	3-8
手順 3: メッセージ配信のサービス品質の指定.....	3-9
自動的なサービス品質機能.....	3-10
QualityOfService クラス .....	3-10
サービス品質設定、オプション、およびデフォルト値 .....	3-11
コード例.....	3-13

メッセージ配信確認レベルの設定 .....	3-14
メッセージ タイムアウトの設定 .....	3-15
タイムアウトのアルゴリズム .....	3-15
配信再試行の回数の設定 .....	3-16
ビジネス メッセージの関連 ID の設定 .....	3-16
手順 4 : XOCP ビジネス メッセージの送信 .....	3-17
同期メッセージ配信 .....	3-17
遅延同期メッセージ配信 .....	3-18
手順 5 : ビジネス メッセージの配信状態の確認 .....	3-19
メッセージ トークン .....	3-20
配信状態のトラッキング .....	3-21
メッセージトラッキング場所 .....	3-22
メッセージトラッキング場所の図 .....	3-23
メッセージトラッキング場所の説明 .....	3-23

#### 4. XOCP ビジネス メッセージの受信

XOCP ビジネス メッセージを受信する方法 .....	4-1
XOCP ビジネス メッセージの受信 .....	4-2

## 索引



---

# このマニュアルの内容

このマニュアルでは、BEA WebLogic Integration B2B Integration の Messaging API を使用して XOCP プロトコル メッセージング アプリケーションを開発する方法について説明します。

**注意：** Messaging API および XOCP ビジネス プロトコルは、WebLogic Integration の本リリースより非推奨になりました。Messaging API および XOCP ビジネス プロトコルの代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

このマニュアルの内容は以下のとおりです。

- 第 1 章「ビジネス メッセージ交換用 XOCP アプリケーションの開発」では、WebLogic Integration の eXtensible Open Collaboration Protocol (XOCP) を使用してビジネス メッセージを交換するアプリケーションの開発に必要な手順について説明します。
- 第 2 章「XOCP アプリケーションのプログラミング ステップ」では、XOCP プロトコルを使用してビジネス メッセージを交換するアプリケーションのプログラミングに必要な手順について説明します。
- 第 3 章「XOCP ビジネス メッセージの送信」では、XOCP ビジネス メッセージを送信するための要件について説明します。
- 第 4 章「XOCP ビジネス メッセージの受信」では、XOCP ビジネス メッセージを受信するための要件について説明します。

## 対象読者

このマニュアルは、WebLogic Integration 環境を拡張する独立ソフトウェアベンダ (ISV) を対象としています。BEA WebLogic Integration のプラットフォームおよび Java プログラミングに読者が精通していることを前提として書かれています。

---

# e-docs Web サイト

BEA 製品のドキュメントは、BEA Systems, Inc. の Web サイトで入手できます。BEA のホーム ページで [製品のドキュメント] をクリックするか、または「e-docs」という製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/index.html>) を直接表示してください。

## このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルを一度に 1 ファイルずつ印刷できます。

このマニュアルの PDF 版は、Web サイトで入手できます。WebLogic Integration PDF を Adobe Acrobat Reader で開くと、マニュアルの全体（または一部分）を書籍の形式で印刷できます。PDF を表示するには、WebLogic Integration ドキュメントのホーム ページを開き、[PDF 版] ボタンをクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader がない場合は、Adobe の Web サイト (<http://www.adobe.co.jp/>) で無料で入手できます。

## 関連情報

次の WebLogic Integration マニュアルには、WebLogic Integration の提供する B2B Integration 機能を使用したメッセージング アプリケーションの記述方法を理解するのに役立つ情報が含まれています。

- *B2B Integration 管理ガイド*
- *B2B Integration 管理アプリケーション プログラミング ガイド*
- *B2B Integration ロジック プラグイン プログラミング ガイド*

---

BEA WebLogic Integration および Java の詳細については、WebLogic Integration マニュアル (<http://edocs.beasys.co.jp/e-docs/index.html>) を参照してください。

## サポート情報

BEA WebLogic Integration のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで **docsupport-jp@bea.com** までお送りください。寄せられた意見については、WebLogic Integration のドキュメントを作成および改訂する BEA の専門の担当者が直に目を通します。

電子メールのメッセージには、BEA WebLogic Integration 2.1 サービス パック 1 のドキュメントをご使用の旨をお書き添えください。

本バージョンの BEA WebLogic Integration について不明な点がある場合、または BEA WebLogic Integration のインストールおよび動作に問題がある場合は、BEA WebSupport (**[websupport.bea.com/custsupp](http://websupport.bea.com/custsupp)**) を通じて BEA カスタマサポートまでお問い合わせください。カスタマサポートへの連絡方法については、製品パッケージに同梱されているカスタマサポート カードにも記載されています。

カスタマサポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

---

# 表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
太字	用語集で定義されている用語を示す。
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
斜体	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 <i>例</i> <pre>#include &lt;iostream.h&gt; void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
太字の等幅 テキスト	コード内の重要な箇所を示す。 <i>例</i> <pre>void <b>commit</b> ( )</pre>
斜体の等幅 テキスト	コード内の変数を示す。 <i>例</i> <pre>String <i>expr</i></pre>

表記法	適用
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 <i>例</i> LPT1 SIGNON OR
{ }	構文の中で複数の選択肢を示す。実際には、この括弧は入力しない。
[ ]	構文の中で任意指定の項目を示す。実際には、この括弧は入力しない。 <i>例</i> buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...
	構文の中で相互に排他的な選択肢を区切る。実際には、この記号は入力しない。
...	コマンドラインで以下のいずれかを示す。 <ul style="list-style-type: none"> <li>■ 引数を複数回繰り返すことができる</li> <li>■ 任意指定の引数が省略されている</li> <li>■ パラメータや値などの情報を追加入力できる</li> </ul> 実際には、この省略記号は入力しない。 <i>例</i> buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...
.	コード サンプルまたは構文で項目が省略されていることを示す。実際には、この省略記号は入力しない。



---

# 1 ビジネス メッセージ交換用 XOCP アプリケーションの開発

**注意：** Messaging API および XOCP ビジネス プロトコルは、WebLogic Integration の本リリースより非推奨になりました。Messaging API および XOCP ビジネス プロトコルの代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

eXtensible Open Collaboration Protocol (XOCP) は、ビジネス メッセージの交換のために WebLogic Integration によって使用されるデフォルトのビジネス プロトコルです。この付録のトピックは以下のとおりです。

- はじめに
- 主要な概念
- XOCP アプリケーションの主要なタスク
- 実行時の情報フロー

## はじめに

WebLogic Integration では、XOCP プロトコルに基づくトレーディング パートナの会話を実装するために 2 つの方法が用意されています。

- WebLogic Integration Studio を使用して作成する Business Process Management (BPM) 協調的ワークフローによる方法。このワークフローは会話でのロールを定義し、ビジネス メッセージがトレーディング パートナによってどのように処理および交換されるかを指定します。協調的ワークフローの作成の詳細については、『*B2B Integration ワークフローの作成*』を参照してください。
- WebLogic Integration Messaging API を使用して作成する XOCP アプリケーションによる方法。XOCP アプリケーションでは、トレーディング パートナ

のロールを実装し、B2B エンジンと直接会話して会話を管理し、ビジネスメッセージを適切に処理します。

このマニュアルでは、トレーディング パートナ間の会話を処理および管理する XOCP アプリケーションの作成で **Messaging API** を使用する方法を説明します。

このマニュアルにあるコード例の多くは、**Messaging API** の例から派生したものです。詳細については、『*B2B Integration サンプルの使い方*』の「**Messaging API サンプル**」を参照してください。

**注意：** 以前に XOCP アプリケーションの作成で使用されていた **C-Enabler API** (WebLogic Integration リリース 2.0 および WebLogic Collaborate 製品) は非推奨になりましたが、依然としてサポートされています。この非推奨になった **API** を使用するアプリケーションを作成する方法の詳細については、次の **URL** を参照してください。

[http://edocs.bea.com/wlintegration/v2\\_0/collaborate/devxocp/index.htm](http://edocs.bea.com/wlintegration/v2_0/collaborate/devxocp/index.htm)

XOCP アプリケーションは、WebLogic Integration リリース 2.0 の **WebLogic Collaborate C-Enabler API** で作成されたものも含めて、別個の Java 仮想マシン (JVM) で非永続モードで実行される必要があります。

## 主要な概念

この節では、XOCP アプリケーションに関連する以下の主要な概念を説明します。

- XOCP アプリケーション
- Messaging API クラス ライブラリ
- XOCP ビジネス メッセージおよびメッセージ エンベロープ
- 会話開始者と会話参加者
- 会話コーディネータ
- トレーディング パートナの状態
- セキュアなメッセージング

## XOCP アプリケーション

*XOCP* アプリケーションは、ユーザによって記述される **Java** アプリケーションです。このアプリケーションは、ハブ アンド スポーク コンフィグレーションでデプロイされる **WebLogic Integration** ノードで実行されます。このノードは、会話定義での特定のロールを実行するために *XOCP* アプリケーションクラスを使用します。ハブ アンド スポーク コンフィグレーションでは、トレーディング パートナの *XOCP* アプリケーションが、スポーク配信チャネル、または **B2B** スポークと関連付けられます。この *XOCP* アプリケーションによって、トレーディング パートナはハブ配信チャネルでコンフィグレーションされた仲介機能またはルーティングプロキシを使用して、**B2B** スポークにいる他のトレーディング パートナと通信をすることができます。

ユーザ記述 *XOCP* アプリケーションは、以下のタスクを実行します。

- *XOCP* アプリケーション セッションの作成およびシャットダウン
- 会話の開始または会話への参加
- 他のトレーディング パートナとの *XOCP* ビジネス メッセージの交換

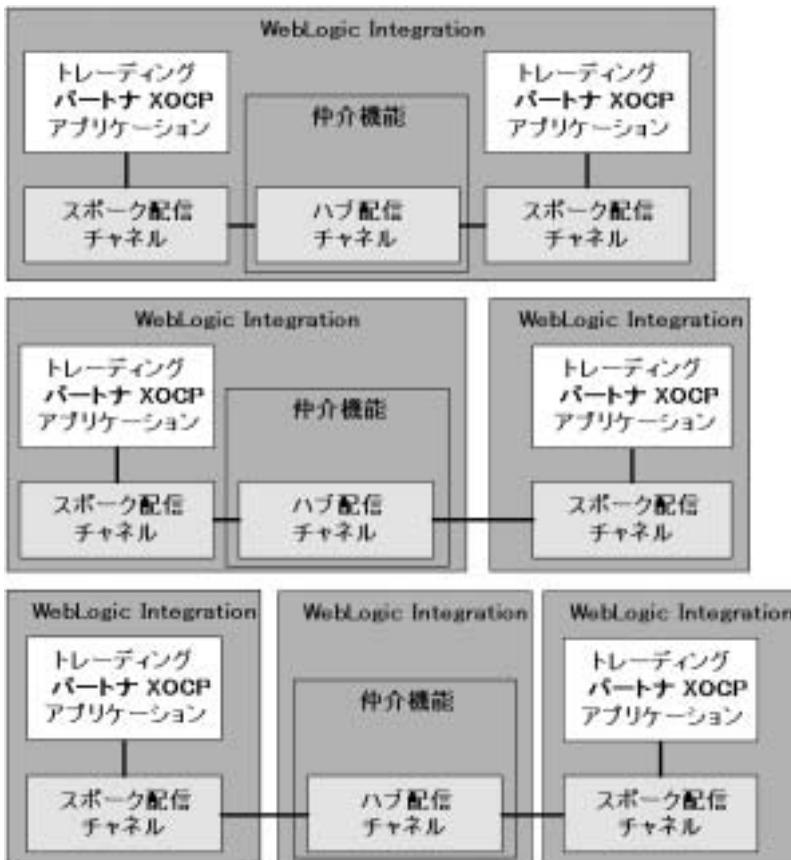
- 会話の終了または会話からの退出

**注意：** XOCP アプリケーション クラスの完全な詳細については、『*BEA WebLogic Integration Javadoc*』の

`com.bea.b2b.protocol.xocp.application` クラスを参照してください。

次の図は、配信チャネル、XOCP アプリケーション、および XOCP アプリケーションをホストする WebLogic Integration について可能な、3 つのハブ アンド スポーク コンフィグレーションを示します。

図 1-1 可能なハブ アンド スポーク コンフィグレーション



1 つの WebLogic Integration ノードは、多数の XOCP アプリケーションをホストできます。XOCP アプリケーションで使用されるハブ アンド スポーク配信チャネルのコンフィグレーションの詳細については、『*B2B Integration 管理ガイド*』の「コンフィグレーション要件」を参照してください。

## XOCP アプリケーション セッション

XOCP アプリケーション セッションによって、XOCP アプリケーションがコラボレーション アグリーメントおよび配信チャネルと関連付けられます。XOCP アプリケーション セッションは、トレーディング パートナと通信するためにアプリケーションによって作成されます。XOCP アプリケーションの範囲は配信チャネルによって区切られます。XOCP アプリケーションは、XOCP アプリケーション クラスの `getXOCPApplicationSession` メソッドを呼び出して XOCP アプリケーション セッションを作成します。

1 つの XOCP アプリケーションを複数の XOCP アプリケーション セッションと関連付けることができるので、アプリケーションは複数の会話に同時に参加できます。

## Messaging API クラス ライブラリ

Messaging API クラス ライブラリは XOCP アプリケーション クラスを含み、XOCP ビジネス メッセージを交換するための API を提供します。Messaging API に含まれるパッケージを、次の表に示します。

表 1-1 Messaging API クラス ライブラリのパッケージ

パッケージ名	説明
<code>com.bea.b2b.protocol.xocp.application</code>	XOCP アプリケーションおよび XOCP アプリケーション セッションでの処理に使用する。このパッケージは、スポーク配信チャネルでコンフィグレーションされたトレーディング パートナによって使用されるアプリケーション向けに設計されている。

表 1-1 Messaging API クラス ライブラリのパッケージ (続き)

パッケージ名	説明
<code>com.bea.b2b.protocol.xocp.conversation.local</code>	XOCP に基づく会話の処理に使用する。
<code>com.bea.b2b.protocol.messaging</code>	会話でのメッセージの処理に使用する。
<code>com.bea.b2b.protocol.xocp.messaging</code>	XOCP に基づく会話でのビジネス メッセージの処理に使用する。

これらのパッケージの詳細については、WebLogic Integration マニュアルの「*BEA WebLogic Integration Javadoc*」を参照するか、Windows システムでは、[スタート | プログラム | BEA WebLogic e-Business Platform | WebLogic Integration 2.1 | Javadocs] を選択してください。

## XOCP ビジネス メッセージおよびメッセージ エンベロープ

XOCP ビジネス メッセージは、XOCP 会話でのトレーディング パートナ間で交換される通信の基本単位です。XOCP ビジネス メッセージは、Messaging API クラス ライブラリでは `com.bea.b2b.protocol.xocp.messaging.XOCPMessage` クラスによって表されます。

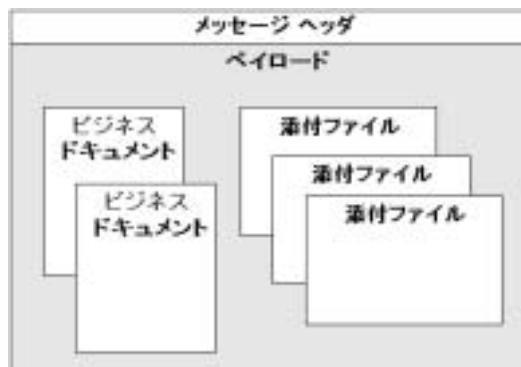
メッセージ エンベロープは、ビジネス メッセージのコンテナです。メッセージ エンベロープには、送信側の情報 (送信側 URL など) および受信側の情報 (送り先 URL など) が含まれています。メッセージ エンベロープは、Messaging API クラス ライブラリでは

`com.bea.b2b.protocol.messaging.MessageEnvelope` クラスによって表されます。ただし、メッセージ エンベロープにプログラム的にアクセスできるのは、XOCP アプリケーションではなく、ロジック プラグインのみです。詳細については、『*B2B Integration ロジック プラグイン プログラミング ガイド*』の「ビジネス メッセージのルーティングとフィルタ処理」および 1-8 ページの「メッセージ エンベロープの情報フロー」を参照してください。

## XOCP ビジネス メッセージの図

次の図は、メッセージ エンベロープと XOCP ビジネス メッセージのコンポーネントを示します。

図 1-2 XOCP ビジネス メッセージのコンポーネント



## XOCP ビジネス メッセージのコンポーネント

XOCP ビジネス メッセージは、マルチパート MIME (Multipurpose Internet Mail Extensions) メッセージです。XOCP ビジネス メッセージは、以下のコンポーネントからなります。

表 1-2 XOCP ビジネス メッセージのコンポーネント

コンポーネント	説明
メッセージ ヘッダ	メッセージ属性。送信側および受信側、会話、サービス品質などについての情報を含んでいる。
ペイロード	1 つまたは複数のビジネス ドキュメント、1 つまたは複数の添付ファイル、あるいはそれら両方の組み合わせのコンテナ。ペイロード コンポーネントは、Messaging API クラス ライブラリでは <code>com.bea.b2b.protocol.messaging.PayloadPart</code> インタフェースによって表される。

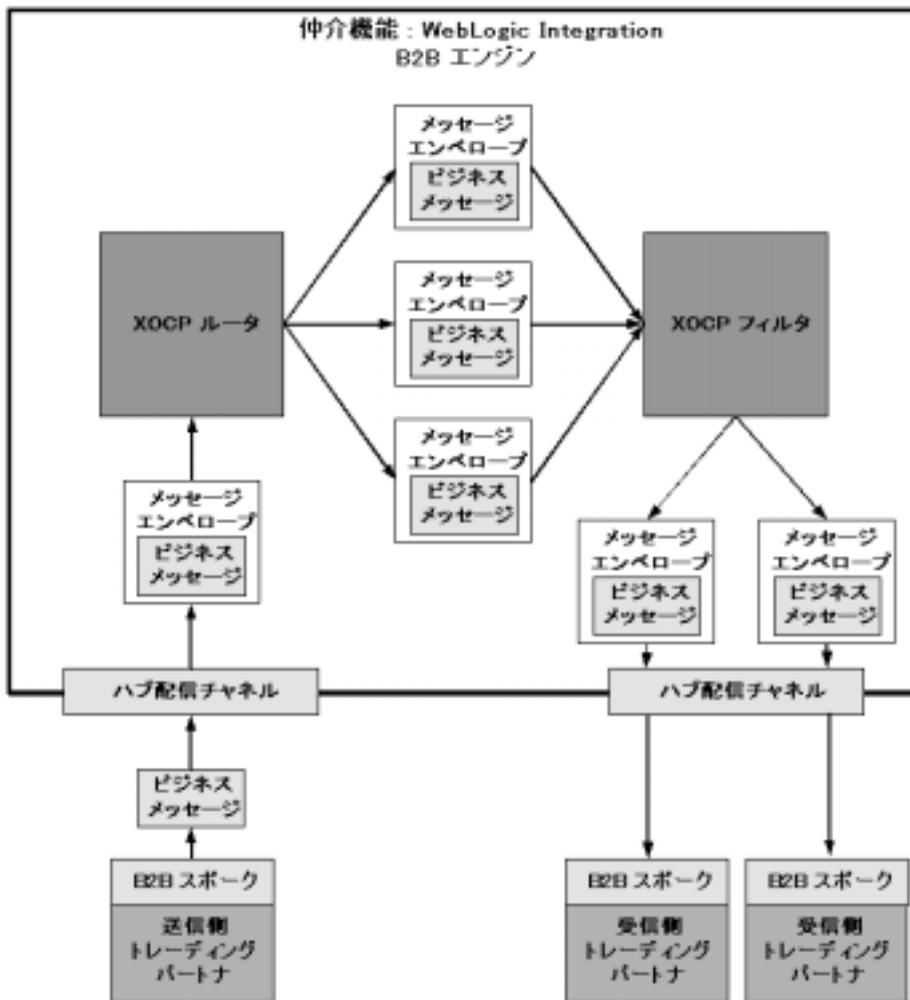
表 1-2 XOCP ビジネス メッセージのコンポーネント (続き)

コンポーネント	説明
ビジネスドキュメント	ペイロードの XML ベースの部分にある XML ファイル。Messaging API クラスライブラリでは <code>com.bea.b2b.protocol.messaging.BusinessDocument</code> クラスによって表される。
添付ファイル	ペイロードの XML ベースでない部分にある非 XML ファイル。内容はバイナリ。Messaging API クラスライブラリでは <code>com.bea.b2b.protocol.messaging.Attachment</code> クラスによって表される。

### メッセージ エンベロープの情報フロー

次の図は、メッセージ エンベロープが WebLogic Integration でどのように処理されるかの例を示します。

図 1-3 WebLogic Integration でのメッセージ エンベロープの処理



メッセージ エンベロープの処理の流れは以下のとおりです。

1. トレーディング パートナがビジネス メッセージを作成して、自身のスポーク配信チャネルから仲介機能のハブ配信チャネルへ送信します。このハブ配信チャネルは、送信側トレーディング パートナに配置された **B2B** エンジン、スタンドアロン マシン、または受信側トレーディング パートナに配置された **B2B** エンジンにコンフィグレーションできます。図 1-1 を参照してください。
2. ビジネス メッセージがハブ配信チャネルで受信されます。**B2B** エンジンには、ビジネス メッセージから特定の送信側情報および受信側情報を抽出した後、メッセージ エンベロープでビジネス メッセージをラップします。
3. **XOCP** ルータがビジネス メッセージを処理し、宛先のリストを検証および決定します。
4. ルータは、宛先のリストにある各宛先ごとに個別のメッセージ エンベロープを作成し、各メッセージ エンベロープにビジネス メッセージの論理的なコピーを挿入してから、すべてのメッセージ エンベロープを **XOCP** フィルタへ転送します。  
図 1-3 に示されている例では、ルータは 3 つの宛先に対するメッセージ エンベロープを作成します。
5. **XOCP** フィルタ内では、各受信側トレーディング パートナに適用されるフィルタが各ビジネス メッセージを評価して、受信側へ送信すべきかを判定します。フィルタは、送信すべきと判定されたメッセージを **B2B** エンジンでの次の処理ステップへ転送します。  
図 1-3 では、3 つのビジネス メッセージがフィルタで評価されます。2 つが送信可、1 つが送信不可とされています。
6. **B2B** エンジンには宛先を検証してから、受信側トレーディング パートナへメッセージ エンベロープ内のビジネス メッセージを送信します。
7. 受信側トレーディング パートナがビジネス メッセージを受信します。

## 会話開始者と会話参加者

どの **XOCP** 会話でも、トレーディング パートナのロールには 2 つの種類があります。

- **会話開始者**は、会話を作成し、最初のビジネス メッセージ（要求など）を1つまたは複数の受信側トレーディング パートナへ送信するトレーディング パートナです。通常、会話開始者は各トレーディング パートナからの応答を待ち、それ以降のビジネス メッセージを交換します。会話が完了したら、会話開始者は会話を終了します（会話がタイムアウトになっていない場合）。
- **会話参加者**は、会話開始者から最初のビジネス メッセージを受信したときに会話に追加されたトレーディング パートナです。通常、会話参加者は会話開始者へ応答を送信し、必要に応じて、それ以降のビジネス メッセージを交換することもできます。会話が完了したら、会話参加者は会話から退出するか、会話が終了するまで待機します。

リポジトリ内の各会話定義には、最低限これらのルールが両方とも含まれています。トレーディング パートナが、関連する会話定義に関連付けられた会話を開始または参加するには、会話の適切なロールにサブスクライブされている必要があります。

通常、会話の開始者は、トレーディング パートナが登録されているロールによって決定されます。たとえば、GetQuote 会話では、バイヤのロールに登録されているトレーディング パートナが GetQuote 会話を開始するのが通例です。セラーのロールに登録されているトレーディング パートナは、GetQuote 会話では会話参加者になるのが通例です。

次の図は、会話開始者および会話参加者が実行するタスクの一部を示します。

図 1-4 会話開始者と会話参加者



## 会話コーディネータ

WebLogic Integration では、実行時に会話を管理する 2 種類の会話コーディネータがサポートされています。グローバル会話コーディネータは B2B 仲介機能でアクティブな会話を管理し、ローカル会話コーディネータは B2B スポークと関連付けられて、グローバル会話コーディネータによるアクティブな会話の管理をローカルに補助します。

次の図は、グローバル会話コーディネータおよびローカル会話コーディネータが WebLogic Integration アーキテクチャのどこで動作するかを示します。

図 1-5 グローバル会話コーディネータおよびローカル会話コーディネータ



## グローバル会話コーディネータ

グローバル会話コーディネータは仲介機能と関連付けられたサービスで、ハブ配信チャネルでコンフィグレーションされます。グローバル会話コーディネータは XOCP のルールに従って会話のライフサイクルを管理し、複数の組織境界にまたがる長期の恒久会話をサポートします。グローバル会話コーディネータは、アクティブな会話のリストを保持します。

グローバル会話コーディネータは、以下のサービスを実行します。

- 会話でのトレーディング パートナの追加と削除
- XOCP 会話終了プロトコルの実行
- 会話に関するステータス情報の保持
- ビジネスプロトコルの実行に対する対話コンテキストの提供

## ローカル会話コーディネータ

ローカル会話コーディネータは、B2B スポークと関連付けられているサービスです。ローカル会話コーディネータは、ローカルトレーディング パートナ（スポーク配信チャネルでコンフィグレーションされたもの）が参加している会話を管理し、アクティブな会話のリストを保持します。各 XOCP アプリケーションセッションには、個別のローカル会話コーディネータがあります。

ローカル会話コーディネータは、以下のタスクを実行します。

- 会話での最初のビジネス メッセージを仲介機能から受信したときに、会話をローカルに追加する
- 会話終了のシステム メッセージを仲介機能から受信したときに、会話をローカルに削除する

## トレーディング パートナの状態

次の表は、トレーディング パートナが XOCP アプリケーション セッションおよび会話参加に関連するタスクを実行する際に、トレーディング パートナに関連付けられる状態を説明します。

表 1-3 トレーディング パートナの状態

状態	説明
REGISTERED	接続されたトレーディング パートナは会話のロールに登録されて、会話を開始したり、会話に参加したりする準備が整っている。
ACTIVE	登録されているトレーディング パートナは少なくとも 1 つの会話に参加している（つまり、ビジネス メッセージを送信または受信している）。
DROPPEDOUT	トレーディング パートナは会話から退出している。

トレーディング パートナの状態の一部は WebLogic Integration B2B Console に表示されます。

# セキュアなメッセージング

トレーディング パートナ間の通信のセキュリティは Secure Sockets Layer (SSL) を使用して確保されています。トレーディング パートナ間でビジネス メッセージを交換できるようにする前に、WebLogic Integration ノードはトレーディング パートナの証明書を利用して各トレーディング パートナのアイデンティティを認証しなければなりません。トレーディング パートナのアイデンティティが認証されれば、ビジネス メッセージはトレーディング パートナの間で安全に交換されます。WebLogic Integration のセキュリティの詳細については、『*B2B Integration セキュリティの実装*』を参照してください。

# XOCP アプリケーションの主要なタスク

この節では、XOCP アプリケーションで実行される主要なタスクについて説明します。

- XOCP アプリケーション セッションの作成
- 会話でのロールの登録
- トレーディング パートナとの会話への関与
- XOCP アプリケーション セッションのシャットダウン

# XOCP アプリケーション セッションの作成

ビジネス メッセージを交換する前に、XOCP アプリケーションはトレーディング パートナおよび関連する配信チャネルのための XOCP アプリケーション セッションを作成する必要があります。

トレーディング パートナの XOCP アプリケーションが XOCP アプリケーション セッションを作成するには、次の条件が満たされる必要があります。

- 配信チャネルおよびトレーディング パートナに関するコンフィグレーション情報が、コラボレーション アグリーメントと関連付けられているハブおよびスポーク配信チャネルの両方の WebLogic Integration リポジトリで定義され

ている必要があります。詳細な情報については、『*B2B Integration 管理ガイド*』の「ワークフローリソースのコンフィグレーション」を参照してください。

- トレーディング パートナは配信チャネルへのアクセスを認証されている必要があります。

**注意：** スポーク配信チャネルと関連付けられた XOCP アプリケーションをホストするマシンが、ハブ配信チャネルと接続した後でクラッシュした場合でも、XOCP アプリケーションは通常の起動時にハブ配信チャネルと再接続できます。以前の XOCP アプリケーション セッションは破棄されて、新しい XOCP アプリケーション セッションに新しいリソースが割り当てられます。ただし、スポーク配信チャネルと関連付けられたマシンがダウンしている間、仲介機能はビジネス メッセージを配信できません。配信されなかったビジネス メッセージは、再試行の回数を超過した場合、またはビジネス メッセージや会話がタイムアウトした場合に破棄されます。

トレーディング パートナが他のトレーディング パートナとのビジネス メッセージの交換を止めるときには、XOCP アプリケーションが XOCP アプリケーション セッションをシャットダウンします。詳細については 1-18 ページの「XOCP アプリケーション セッションのシャットダウン」を参照してください。

## 会話でのロールの登録

XOCP アプリケーション セッションが作成されたら、トレーディング パートナは、コラボレーション アグリーメントにバインドされる会話型のメッセージリスナを登録する必要があります。メッセージリスナは、トレーディング パートナが会話にどのように参加するかを定義する会話型に応じて登録される必要があります。

ロールの登録では、ハブ配信チャネルと関連付けられているリポジトリの以下の情報が必要です。

- **会話型**—会話定義のサブセット。トレーディング パートナがサブスクライブされている会話定義でのトレーディング パートナのロールに基づいて、トレーディング パートナの会話を定義します。
- **メッセージ定義**—順序付けされたメッセージ パートからなります。1 つのメッセージ パートは、コンテンツ タイプ (XML またはバイナリ) の他に、

ドキュメント定義を含むことができます。あるパートのコンテンツタイプが XML であれば、そのパートにはドキュメント定義が必要です。バイナリタイプであれば、他に必要な情報はありません。

これらの概念については、『*B2B Integration 入門*』の「概要」を参照してください。

コラボレーション アグリーメントの会話型に応じたメッセージリスナを登録する前に、トレーディング パートナは最初に登録するための認証を受ける必要があります。認証は、仲介機能の管理者によってコンフィグレーションされ、トレーディング パートナが会話定義のどのロールにサブスクライブしているかに基づいています。

XOCP アプリケーション セッションでコラボレーション アグリーメントの特定の会話型へのメッセージリスナの登録が試みられると、スポーク配信チャンネルは仲介機能へ XOCP システム メッセージ（会話への登録）を送信します。仲介機能は、関連付けられた配信チャンネルで要求される会話型についてトレーディング パートナのロールを検証します。登録が有効であれば、以後トレーディング パートナは登録された会話型と関連付けられている会話について開始および参加ができます。この時点で、トレーディング パートナは REGISTERED 状態になり、会話について開始または参加をする準備が整います。

## トレーディング パートナとの会話への関与

会話のロールに登録されたら、トレーディング パートナは自身のロールに従って会話に関与できます。会話の開始および参加は、仲介機能それ自体の上で発生します。ただし、XOCP アプリケーション セッションは、自身が関係している会話の状態に関する情報の一部を保守します。

会話開始者 XOCP アプリケーションおよび会話参加者 XOCP アプリケーションは、非常に似ています。ただし、会話開始者 XOCP アプリケーションは会話を終了できますが、会話参加者 XOCP アプリケーションはできません。会話参加者 XOCP アプリケーションは、会話から退出できるだけです。

## 会話の開始とビジネス メッセージの送信

会話を開始するには、まず会話開始者 XOCP アプリケーションが会話を作成します。会話開始者 XOCP アプリケーションはタイムアウト値を指定することもできます。このタイムアウト値の時間が経過すると、会話は自動的に終了されます。この値は、リポジトリにある関連する会話定義で指定されているタイムアウト値をオーバーライドします。

**B2B** スポーク上のローカル会話コーディネータは XOCP システム メッセージ（特定のコラボレーション アグリーメントの会話を作成）を仲介機能へ送信します。仲介機能にあるグローバル会話コーディネータは、適切な配信チャネルを使用して会話を作成し、会話開始者としてトレーディング パートナを追加します。会話が作成されたら、会話開始者 XOCP アプリケーションはビジネス メッセージを作成および送信します。詳細については 3-1 ページの「XOCP ビジネス メッセージの送信」を参照してください。

## 会話への参加

仲介機能にあるグローバル会話コーディネータは、会話に関して仲介機能が受信するすべてのビジネス メッセージを処理します。仲介機能が最初のビジネス メッセージを受信者トレーディング パートナ（複数可）へ配信した後で、グローバル会話コーディネータは配信先となったトレーディング パートナを会話へ追加します。トレーディング パートナが会話に追加されたら、そのトレーディング パートナは ACTIVE 状態になり、会話の中でビジネス メッセージの送信および受信ができます。

対象のスポーク配信チャネル上の XOCP アプリケーション セッションが会話で最初のビジネス メッセージを受信すると、必要な処理（会話をローカルリストに登録するなど）を済ませてから、メッセージリスナで `onMessage` コールバックを呼び出します。詳細については、4-1 ページの「XOCP ビジネス メッセージの受信」を参照してください。

登録されたトレーディング パートナが会話に追加されたら、そのトレーディング パートナは ACTIVE 状態になり、会話の中でビジネス メッセージの送信および受信ができます。

### 会話からの退出

会話への参加が完了したら、会話参加者トレーディング パートナは会話から退出することができます。トレーディング パートナが会話から退出すると、会話コーディネータによって、参加トレーディング パートナのリストから削除されます。その会話での今後のビジネス メッセージは、退出したトレーディング パートナには送信されません。トレーディング パートナが退出した後、そのトレーディング パートナは会話が続く間 DROPPEDOUT 状態で残されます。

### 会話の終了

会話が終了するのは、会話開始者であるトレーディング パートナが明示的に会話を終了するか、会話がタイムアウトした場合です。どちらかが最初に発生した時点で会話は終了します。会話を開始したトレーディング パートナは、ビジネス プロセスの適切な時点で会話を終了させなければなりません。

**注意：** 会話の開始者のみが会話を終了できます。

会話が終了すると、会話コーディネータは参加者トレーディング パートナのすべてに XOCP システム メッセージ（終了メッセージ）を送信します。このメッセージは、B2B スポーク上の XOCP アプリケーション セッションの登録済みメッセージ リスナの間で、コールバック onTerminate として伝播されます。

## XOCP アプリケーション セッションのシャットダウン

トレーディング パートナが会話でのすべてのアクティビティを終了すると、XOCP アプリケーションによって XOCP アプリケーション セッションがシャットダウンされます。XOCP アプリケーションが XOCP アプリケーション セッションをシャットダウンすると、そのスポークと関連付けられている B2B エンジン、セッションと関連付けられているすべてのコラボレーション アグリーメントの登録を仲介機能から解除します。これにより、仲介機能は関連付けられている会話型の登録を解除します。これに応じて会話コーディネータは、XOCP アプリケーション セッションで当該のトレーディング パートナによって開始されたすべての会話を自動的に終了し、参加しているすべての会話からトレーディング パートナを削除します。

トレーディング パートナが **XOCP** アプリケーション セッションをシャットダウンすると、次のような結果になります。

- 仲介機能は、指定された配信チャネルにいるトレーディング パートナへの以後のメッセージ送信をすべて停止します。
- 当該のトレーディング パートナによって開始されたすべての会話が終了されます。
- トレーディング パートナは、参加しているすべての会話から退出します。
- トレーディング パートナは、シャットダウンされる **XOCP** アプリケーション セッションのために仲介機能で割り当てられていたリソースを戻します。

## 実行時の情報フロー

実行時には、すべての **XOCP** アプリケーションが特定のタスクを同じように実行します。配信チャネルへ接続し、メッセージ リスナを登録してから、アプリケーションセッションのシャットダウンを同じ方法で行います。ただし、個々の会話の間に会話開始者および会話参加者が実行する一連のタスクは、お互いに明確な相違点がありながらも関連しています。

## 情報フローの図

次の図は、会話開始者と会話参加者の間の、実行時の情報フローを示します。

図 1-6 会話開始者と会話参加者の間の情報フロー



この図は、1つの会話と、ビジネス メッセージの最小限の交換（要求と応答）に関する簡略化された例です。実際には、メッセージ リスナを登録してから XOCP アプリケーション セッションを終了するまでに、トレーディング パート

ナは複数の会話に参加することもあります。さらに、1つの会話の中で、複数のトレーディング パートナが1つの要求および応答にとどまらず、多くのビジネス メッセージを交換する場合があります。

## 情報フローでのステップ

実行時には、複数のトレーディング パートナ間の情報のフロー（仲介機能を通じて通信する **XOCP** アプリケーションによるもの）は、以下の順序で進みます。

1. 特定の配信チャネルを持つ各トレーディング パートナが、**XOCP** アプリケーションセッションを作成します。
2. 各トレーディング パートナ **XOCP** アプリケーションは、作成された **XOCP** アプリケーションセッションにメッセージリスナを登録し、次にメッセージリスナは（ローカル会話コーディネータの補助を得て）、仲介機能によって保持されている指定されたコラボレーション アグリーメントにある、会話中のロールにトレーディング パートナを登録します。
3. 各トレーディング パートナ **XOCP** アプリケーションは、コラボレーション アグリーメント **ID** を取得します（未知であった場合）。
4. 会話は、会話開始者 **XOCP** アプリケーションによって作成された時点で開始されます。
5. グローバル会話コーディネータは、そのグローバル会話リストに会話のインスタンスを追加し、トレーディング パートナを開始者としてマークします。
6. 会話開始者のローカル会話コーディネータは、そのローカル会話リストに会話のインスタンスを追加します。
7. 会話開始者の **XOCP** アプリケーションが、ビジネス メッセージ（要求など）を作成して送信します。
8. 会話開始者の **XOCP** アプリケーションセッションは、ビジネス メッセージを仲介機能のハブ配信チャネルへ配信します。
9. 仲介機能はビジネス メッセージを会話参加者のスポーク配信チャネルへ配信します。
10. 仲介機能のグローバル会話コーディネータは、グローバル会話リストにある会話インスタンスのエントリに参加トレーディング パートナを追加することで、会話に参加トレーディング パートナを追加します。

11. 仲介機能のローカル会話コーディネータは、ビジネス メッセージを受信するほか、ローカル会話リストに会話インスタンスを追加することで、ローカルな会話にトレーディング パートナを追加します。
12. 会話参加者 XOCP アプリケーションにある `onMessage` の実装が呼び出されて、ビジネス メッセージを処理します。
13. 会話参加者の XOCP アプリケーションが、ビジネス メッセージ（応答など）を作成して会話開始者へ返信します。
14. 会話参加者と関連付けられている XOCP アプリケーション セッションが、仲介機能へビジネス メッセージを配信します。
15. 仲介機能はビジネス メッセージを受信して、会話開始者へ配信します。
16. 会話開始者がビジネス メッセージを受信します。
17. 会話開始者 XOCP アプリケーションにある `onMessage` の実装が呼び出されて、ビジネス メッセージを処理します。
18. 会話を終了するには、まず会話開始者 XOCP アプリケーションが会話を終了します。  
**注意：** 会話のタイムアウトが経過した場合は、会話が自動的に終了します。
19. 会話開始者のローカル会話コーディネータは、仲介機能にあるグローバル会話コーディネータへ終了の通知を配信します。
20. 仲介機能にあるグローバル会話コーディネータは、グローバル会話リストから会話参加者を削除してから、会話参加者と関連付けられているローカル会話コーディネータへ終了の通知を配信します。
21. 会話参加者と関連付けられているローカル会話コーディネータは、終了の通知を受信してローカル会話リストから会話を削除します。
22. 会話参加者 XOCP アプリケーションにある `onTerminate` の実装が呼び出されます。
23. 仲介機能にあるグローバル会話コーディネータは、会話を終了としてマークしてから、会話開始者に会話終了の確認を送信して通知します。
24. 会話開始者が会話終了の通知を受信します。
25. 会話開始者のローカル会話コーディネータは、終了の通知を受信してローカル会話リストから会話を削除します。

26. 会話開始者 XOCP アプリケーションにある `onTerminate` の実装が呼び出されます。
27. 各トレーディング パートナ XOCP アプリケーションによって、それぞれの XOCP アプリケーションセッションがシャットダウンされます。

以上のステップの詳細については、1-14 ページの「XOCP アプリケーションの主要なタスク」を参照してください。



---

## 2 XOCP アプリケーションのプログラミング ステップ

以下の節では、開発者が通常 XOCP アプリケーションに用意するプロシージャの各ステップを説明します。

- 手順 1: パッケージのインポート
- 手順 2: `MessageListener` インタフェースの実装
- 手順 3: XOCP アプリケーション セッションの作成
- 手順 4: メッセージ リスナの作成と登録
- 手順 5: 会話の開始または会話への参加
- 手順 6: ビジネス メッセージの交換
- 手順 7: 会話の終了
- 手順 8: XOCP アプリケーション セッションのシャットダウン

各節には、`Messaging API` サンプルからコード例が引用されています。これらの詳細については、『*B2B Integration サンプルの使い方*』を参照してください。

**注意:** XOCP アプリケーションを実行する前に、管理者は XOCP アプリケーションと関連付けられている会話に参加するトレーディング パートナが使用するコラボレーション アグリーメントを指定する必要があります。詳細な情報については、『*B2B Integration 管理ガイド*』の「ワークフロー リソースのコンフィグレーション」を参照してください。WebLogic Integration の以前のバージョンで作成された XOCP アプリケーションの下位互換性の詳細については、『*WebLogic Integration リリース 7.0 への移行*』を参照してください。

### 手順 1 : パッケージのインポート

XOCP アプリケーションは、Messaging API クラス ライブラリから必要なパッケージをインポートします。これらのパッケージの詳細については、1-5 ページの「Messaging API クラス ライブラリ」を参照してください。

次のリスト例は、インポートする必要があるパッケージの種類を示します。

#### コード リスト 2-1 パッケージのインポート

---

```
import java.util.Properties;
import com.bea.b2b.protocol.xocp.application.*;
import com.bea.b2b.protocol.xocp.messaging.*;
import com.bea.b2b.protocol.conversation.ConversationType;
import com.bea.b2b.protocol.messaging.PayloadPart;
import com.bea.b2b.protocol.xocp.conversation.local.Conversation;
import com.bea.eci.logging.*;
```

---

### 手順 2 : MessageListener インタフェースの実装

メッセージを受信するために、XOCP アプリケーションは以下のインタフェースを実装する必要があります。

```
com.bea.b2b.protocol.xocp.messaging.XOCPMessageListener
```

このインタフェースによって提供される `onMessage` および `onTerminate` メソッドは、受信するビジネス メッセージの処理、および会話終了の通知の処理にそれぞれ使用されます。`onMessage` メソッドは、B2B スポークがビジネス メッセージを受信したときに呼び出されます。`onTerminate` メソッドは、B2B スポークが会話終了の通知を受信したときに呼び出されます。

メッセージ リスナは、トレーディング パートナが会話中にビジネス メッセージを受信するために必要です。XOCP アプリケーション セッションは、コラボレーション アグリーメントあたり 1 つのメッセージ リスナをサポートします。

---

## コード リスト 2-2 メッセージ リスナ インタフェースの実装

---

```
public class Partner1MessageListener
    implements XOCPLMessageListener{

    public void onMessage(XOCPLMessage rmsg){

        counter ++;

        QualityOfService qos = rmsg.getQoS();

        // 応答を受信、処理の開始時を待機

        synchronized(waiter){

            debug("onMessage in waiter counter = " + counter);

            PayloadPart[] payload = rmsg.getPayloadParts();
            // シングル パート ドキュメントを使用
            if (payload != null && payload.length > 0){
                BusinessDocument bd = (BusinessDocument)payload[0];
                waiter.reply = bd.getDocument();
            }
            waiter.done = true;
            waiter.notify();
        }

        public void onTerminate(Conversation conv, int result)
    }
}
```

---

XOCPLMessageListener インタフェースの詳細については、**WebLogic Integration** マニュアルの『*BEA WebLogic Integration Javadoc*』を参照するか、Windows システムでは、[ スタート | プログラム | **BEA WebLogic e-Business Platform | WebLogic Integration 2.1 | Javadocs**] を選択してください。

## 手順 3 : XOCPL アプリケーション セッションの作成

会話について開始または参加をするために、トレーディング パートナはローカル B2B スポーク配信チャネルと関連付けられる XOCPL アプリケーション セッションを作成します。各 XOCPL アプリケーションセッションでは、関連付けられたトレーディング パートナが、会話中の他のトレーディング パートナとメッセージを交換できます。

## 2 XOCP アプリケーションのプログラミング ステップ

---

新しい XOCP アプリケーションを作成したり、既存のものを取得したりするには、`com.bea.b2b.protocol.xocp.application.XOCPApplication` クラスを使用します。次のリストは、`MdmApp1` という XOCP アプリケーションを、トレーディング パートナ名 `Partner 1`、配信チャネル名 `Partner1-Channel0` として取得します。

### コード リスト 2-3 XOCP アプリケーション セッションの取得

---

```
XOCPApplication app = XOCPApplication.getXOCPApplication("MdmApp1");
XOCPApplicationSession es = app.getXOCPApplicationSession("Partner1",
"Partner1-Channel0");
```

---

## 手順 4 : メッセージ リスナの作成と登録

会話に参加するには、XOCP アプリケーションはメッセージ リスナを登録する必要があります。メッセージ リスナはアプリケーションによって実装されます。開発者は、必要に応じてメッセージ リスナを使用します。

メッセージ リスナを登録するために、XOCP アプリケーションは `XOCPApplicationSession` インスタンスの `registerMessageListener` メソッドを呼び出します。呼び出しでは、コラボレーション アグリーメント ID、トレーディング パートナの会話ロール、およびメッセージ リスナ オブジェクトをパラメータとして渡します。

次のリスト例は、`verifierConversation` 会話の要求者ロール（通常は会話開始者）のメッセージ リスナを登録する方法を示します。要求されるコラボレーション アグリーメント ID およびロールがトレーディング パートナおよび仲介機能のリポジトリにそれぞれ指定されている必要がある点に注意してください。

### コード リスト 2-4 メッセージ リスナの登録

---

```
Partner1MessageListener ml = new Partner1MessageListener();

Properties prop = new Properties();
prop.setProperty("BusinessProcessName", "verifierConversation");
prop.setProperty("BusinessProcessVersion", "1.0");
prop.setProperty("otherTradingPartner", "Hub");
```

---

```
prop.setProperty("toRole", "replier");
String caId = es.getCAId(prop);

String myRole = "requestor";

es.registerMessageListener(caId, myRole, ml);
```

---

## 手順 5 : 会話の開始または会話への参加

会話開始者アプリケーションは、明示的に会話を開始します。会話を開始するために、開始者トレーディング パートナは

`com.bea.b2b.protocol.xocp.application.XOCPApplicationSession` インスタンスの `createConversation` メソッドを呼び出します。呼び出しのパラメータとしては、コラボレーション アグリーメント ID、トレーディング パートナのロール、そして省略可能な値として会話のタイムアウト値があります。タイムアウト値は秒単位で指定されます。コンフィグレーションされたタイムアウトが、トレーディング パートナおよび仲介機能リポジトリの両方の会話定義でそれぞれゼロである場合、デフォルト値はゼロ、つまりタイムアウトなしです。トレーディング パートナは会話定義の開始者ロールに登録されている必要があります。

次のリスト例は、会話が開始される方法を示します。

### コード リスト 2-5 会話の開始

---

```
long timeout = 0;
Conversation c = es.createConversation(caId, myRole, timeout);
```

---

## 手順 6 : ビジネス メッセージの交換

会話開始者アプリケーションが会話を作成した後は、会話中の他のトレーディング パートナとビジネス メッセージの交換を開始できます。

最初は、会話開始者アプリケーションがビジネス メッセージ（要求など）を作成して、会話中の 1 つまたは複数のトレーディング パートナへ送信します。トレーディング パートナがビジネス メッセージを受信すると、会話参加者アプリケーションがビジネス メッセージを処理して、通常は、ビジネス メッセージへの応答を作成して送信します。トレーディング パートナは会話中に複数のビジネス メッセージを送信および受信できます。ビジネス メッセージの交換の詳細については、3-1 ページの「XOCP ビジネス メッセージの送信」および 4-1 ページの「XOCP ビジネス メッセージの受信」を参照してください。

## 手順 7 : 会話の終了

会話の中でトレーディング パートナがビジネス メッセージの交換を終了したら、その会話を終了させることができます。トレーディング パートナが会話への関わりをどのようにして終了するかは、会話の中でのロールによって異なります。

### 参加者が会話から退出する場合

参加者であるトレーディング パートナは会話から退出することができます。会話から退出するには、参加者 XOCP アプリケーションが Conversation インスタンスの `leave` メソッドを呼び出します。呼び出しでは `false` が渡されます。参加者が参加していないと、仲介機能にはメッセージが保持されません。

**注意：** このリリースでは `false` 引数のみがサポートされています。

次のリスト例は、参加者が会話から退出する方法を示します。

#### コード リスト 2-6 会話からの退出

---

```
c.leave(false);
```

---

### 開始者が会話を終了する場合

会話開始者は明示的に会話を終了したり、会話がタイムアウトするまで待機したりできます。会話開始者は会話の作成時にタイムアウト値を指定できます。ゼロを指定すると、トレーディング パートナまたは仲介機能のリポジトリで会話用

---

に定義されているタイムアウト値が使用されます。会話が終了すると、会話開始者およびすべての参加者トレーディング パートナが会話から削除され、すべての未配信ビジネス メッセージが破棄され、関連付けられているシステム リソースが解放されます。

会話を明示的に終了するには、開始者 XOCP アプリケーションが自身の `Conversation` インタフェースの実装で `terminate` メソッドを呼び出します。次にリスト例を示します。

---

#### コード リスト 2-7 会話の終了

---

```
c.terminate(Conversation.SUCCESS);
```

---

## 手順 8 : XOCP アプリケーション セッションのシャットダウン

XOCP アプリケーション セッションをシャットダウンして会話から退出するために、アプリケーションはその `XOCPApplicationSession` インタフェースの実装で `shutDown` メソッドを使用します。次のリスト例は、XOCP アプリケーション セッションをシャットダウンする方法を示します。

---

#### コード リスト 2-8 XOCP アプリケーション セッションのシャットダウン

---

```
es.shutDown();
```

---

XOCP アプリケーションによって XOCP アプリケーション セッションがシャットダウンされると、トレーディング パートナは会話を自動的かつ永続的に退出します。



---

## 3 XOCP ビジネス メッセージの送信

以下の節では、XOCP アプリケーションが会話の中で1つまたは複数のトレーディング パートナに対してビジネス メッセージをどのようにして送信するかを説明します。

- 手順 1: ビジネス メッセージの作成
- 手順 2: ビジネス メッセージの受信側の指定 (省略可能)
- 手順 3: メッセージ配信のサービス品質の指定
- 手順 4: XOCP ビジネス メッセージの送信
- 手順 5: ビジネス メッセージの配信状態の確認

XOCP ビジネス メッセージを送信するために、XOCP アプリケーションはビジネスドキュメントを構築し、ビジネス メッセージを作成し、メッセージルーティング基準やサービス品質といった配信オプションを指定してから、ビジネスメッセージを処理のために仲介機能へ送信します。また、XOCP アプリケーションは、ビジネス メッセージの配信ステータスを、正常に配信されたかどうかも含めて確認します。XOCP ビジネス メッセージの概要については、1-6 ページの「XOCP ビジネス メッセージおよびメッセージエンベロープ」を参照してください。

### 手順 1: ビジネス メッセージの作成

ビジネス メッセージを作成するために、XOCP アプリケーションは最初にメッセージ ペイロードを作成します。メッセージ ペイロードは、送信する任意のビジネスドキュメントおよび添付ファイルからなります。

ペイロードの作成には以下の3つのステップがあります。

1. 必要なパッケージのインポート

#### 2. ペイロード部分の作成

#### 3. XOCP ビジネス メッセージの作成とペイロード部分の追加

この節では、この3つのステップについて説明します。XOCP ビジネス メッセージのコンポーネントの概要については、1-6 ページの「XOCP ビジネス メッセージおよびメッセージ エンベロープ」を参照してください。

## 必要なパッケージのインポート

ビジネス メッセージを作成するために、XOCP アプリケーションは必要なパッケージをインポートします。次のリストに、必要なパッケージを示します。

#### コード リスト 3-1 ビジネス メッセージの作成に必要なパッケージのインポート

---

```
class java.io.FileInputStream;
import org.apache.xerces.dom.*;
import com.bea.b2b.protocol.xocp.application.*;
import com.bea.b2b.protocol.xocp.messaging.*;
import com.bea.b2b.protocol.messaging.Attachment;
import com.bea.eci.logging.*;
```

---

## ペイロード部分の作成

次に XOCP アプリケーションはメッセージ ペイロードを作成します。ペイロードには、ビジネスドキュメントまたは添付ファイル、あるいはその両方が含まれます。

## XML ドキュメントの作成

ビジネスドキュメントには、1つまたは複数のビジネスドキュメントが入ります。ビジネスドキュメントは、ビジネス メッセージ中の XML ベースのペイロード部分です。ビジネスドキュメントは、`com.bea.b2b.protocol.messaging.BusinessDocument` クラスのインスタンスです。

`BusinessDocument` オブジェクトには XML ドキュメントが格納されます。XML ドキュメントは、World Wide Web Consortium ([www.w3.org](http://www.w3.org)) によって公開されている `org.w3c.dom` パッケージにある `org.w3c.dom.Document` クラスのインスタンスです。また、XOCP アプリケーションはサードパーティによって実装された同様のパッケージも使用できます。たとえば、Apache XML Project ([www.apache.org](http://www.apache.org)) で提供されている `org.apache.xerces.dom` パッケージなどです。このパッケージは、Messaging API のサンプルで XML ドキュメントの作成および処理に使用されています。

**注意:** 各 XML ドキュメントで指定される文書型パラメータは、リポジトリで会話定義と関連付けられているメッセージ定義のコンテンツ タイプ部分にマップする必要があります。

次のコードは Messaging API アプリケーションの `MdmTp1Servlet` から引用したもので、要求を XML ドキュメントの形式で作成します。

### コード リスト 3-2 XML ドキュメントの作成

---

```
// 要求ドキュメントを作成
DOMImplementationImpl domi = new DOMImplementationImpl();
DocumentType dType =
    domi.createDocumentType("request", null, "request.dtd");
org.w3c.dom.Document rql = new DocumentImpl(dType);
Element root1 = rql.createElement("request");
String sendStr1 = "FIRST MESSAGE"; // 他のパートナによって処理される
                                     // 実際の文字列データ
root1.appendChild(rql.createTextNode(sendStr1));
rql.appendChild(root1);
```

---

XML ドキュメントを作成した後で、XOCP アプリケーションは `BusinessDocument` オブジェクトを作成します。このとき、パラメータとして XML ドキュメント (`payload[0]`) をコンストラクタに渡します。次のリストにコードを示します。

### コード リスト 3-3 BusinessDocument オブジェクトの作成

---

```
BusinessDocument bd = (BusinessDocument)payload[0];
```

---

## 添付ファイルの作成

ビジネスドキュメントには、1つまたは複数の添付ファイルを入れることができます。添付ファイルは、バイナリのコンテンツを含むビジネス メッセージの非XML ベースのペイロード部分です。添付ファイルは、`com.bea.b2b.protocol.messaging.Attachment` クラスのインスタンスです。詳細については、『*BEA WebLogic Integration Javadoc*』を参照してください。

次のリスト例は、添付ファイルを作成する方法を示します。

### コード リスト 3-4 添付ファイルの作成

---

```
FileInputStream fis = new FileInputStream("somefile");
Attachment att = new Attachment (fis);
```

---

## XOCP ビジネス メッセージの作成とペイロード部分の追加

メッセージ ペイロードを作成した後で、XOCP アプリケーションは XOCP ビジネス メッセージを作成してペイロード部分を追加します。

`com.bea.b2b.protocol.xocp.messaging.XOCPMessage` クラスが XOCP ビジネス メッセージを表します。詳細については、『*BEA WebLogic Integration Javadoc*』を参照してください。

ビジネス メッセージを作成するために、XOCP アプリケーションは以下の処理を行います。

1. `XOCPMessage` クラスのインスタンスを作成します。
2. `XOCPMessage` メッセージ オブジェクトにある以下のメソッドのいずれかを呼び出して、ビジネス メッセージにペイロード部分を追加します。
  - `addPayloadPart` メソッド。ビジネス メッセージに1つのビジネスドキュメントまたは添付ファイルを追加します。
  - `addPayloadParts` メソッド。ビジネス メッセージに複数のビジネスドキュメントまたは添付ファイルを追加します。

次のリストでは、XOCP ビジネス メッセージが作成されて、ペイロード部分が追加されます。

### コード リスト 3-5 ビジネス メッセージの作成とペイロード部分の追加

---

```
XOCPMessage msg1 = new XOCPMessage("");
msg1.addPayloadPart(new BusinessDocument(rq1));
```

---

**注意：** XOCP アプリケーションは XOCPMessage の内容（そのペイロード部分は除く）を複製してから仲介機能へ送信します。このため、アプリケーションによって XOCPMessage 上で send または sendAndWait メソッドが呼び出された後は、ペイロード部分を変更しないでください。

## 手順 2 : ビジネス メッセージの受信側の指定 (省略可能)

ビジネス メッセージを作成した後で、XOCP アプリケーションはメッセージの宛先となるトレーディング パートナを指定することができます。XOCP アプリケーションは、たとえば会話開始者から受信した要求に応答するときなどに、特定のトレーディング パートナへビジネス メッセージを送信できます（ポイントツーポイント交換）。あるいは、特定のビジネス基準（XOCP XPath 式で表される）を満たす場合に、XOCP アプリケーションは複数のトレーディング パートナへ（マルチキャストを使用して）ビジネス メッセージを送信できます。たとえば、特定の製品を扱う複数のセラーに対してバイヤが入札要求を送信するとき、アプリケーションはマルチキャストを使用してメッセージを送信できます。

いずれの場合も、送信の対象となるトレーディング パートナのセットは、会話定義の中で適切なロールにサブスクライブされているものに限定されます。加えて、仲介機能のリポジトリで定義されているルータおよびフィルタ式も、特定のトレーディング パートナへのメッセージ配信に影響する場合があります。詳細については、『*B2B Integration 管理ガイド*』の「高度なコンフィグレーション タスク」を参照。

## 特定のトレーディング パートナの指定

XOCP ビジネス メッセージが単一の、既知のトレーディング パートナへ送信される場合、XOCP アプリケーションは `XOCPMessage` オブジェクトの `setRecipient` メソッドを呼び出すことができます。このとき、パラメータとしてはトレーディング パートナ名が渡されます。指定されたトレーディング パートナは、仲介機能のリポジトリで定義されている必要があります。

次のリスト例は、`ChipMaker` という名前のトレーディング パートナがビジネス メッセージの受信側として指定されるための方法を示します。

### コード リスト 3-6 単一のトレーディング パートナの指定

---

```
String tradingPartnerName = "ChipMaker";
XOCPMessage msg = new XOCPMessage();
msg.setRecipient(tradingPartnerName);
```

---

ビジネス メッセージに対して `setRecipient` を使用することで、メッセージの配信が早く処理されます。これは、仲介機能が、たとえばトレーディング パートナの評価や仲介機能の XPath 式の評価といった、通常のルータ処理を実行せずに済むためです。ただし、ビジネス メッセージは依然として仲介機能の適切なフィルタ処理の対象です。詳細については、『*B2B Integration 管理ガイド*』の「高度なコンフィグレーション タスク」を参照。

## XPath 式によるメッセージ受信側の基準の指定

XOCP アプリケーションは、ビジネス メッセージを受信するトレーディング パートナのセットの基準を XPath 式で指定できます。XPath 式は、XML ドキュメントの各部を指定するために使用されます。詳細については、『*B2B Integration 管理ガイド*』の「高度なコンフィグレーション タスク」を参照。

XPath 式は、仲介機能のリポジトリのドキュメント フォーマットに固有である必要があります。また、トレーディング パートナの各要素のノード固有のセットを定義する必要もあります。XPath 式は、以下の属性に基づいて受信側トレーディング パートナを選択します。これらの属性は、仲介機能のリポジトリに定義されています。

- 標準的な属性。たとえば、トレーディング パートナ名や郵便番号
- 拡張プロパティ。仲介機能の管理者によって定義されたカスタム属性、要素、およびテキスト

XPath 式は、ビジネス メッセージのメッセージ ヘッダの一部として XOCF アプリケーションから仲介機能へ渡されます。仲介機能は、この XPath 式を仲介機能のリポジトリで定義されている他の XPath 式とともに使用して、ビジネス メッセージのメッセージ受信側のセットを決定します。

仲介機能リポジトリに適切なトレーディング パートナおよび仲介機能 XPath 式が定義されている場合は、その仲介機能をホストしている B2B エンジンによって、ビジネス メッセージが受信された後にこれらの式が評価されます。式のコンフィグレーションによっては、これらの XPath 式は XOCF アプリケーションで指定される XOCF XPath 式をオーバーライドしたり、追加をしたりします。詳細については、『*B2B Integration 管理ガイド*』の「高度なコンフィグレーション タスク」を参照。

XOCF ビジネス メッセージの XOCF XPath 式を指定するために、XOCF アプリケーションは XOCFMessage オブジェクトの `setExpression` メソッドを呼び出します。このとき、XPath 式がパラメータとして渡されます。

**注意：** Apache Xalan バージョン 1.0.1 では、文字列リテラルの区切り文字として二重引用符ではなく一重引用符がサポートされています。

ビジネス メッセージが配信される前に、適切なルータおよびフィルタ処理が仲介機能で実行されます。

## 標準的なトレーディング パートナ属性の指定

次のリストは、指定された名前のトレーディング パートナを選択する XOCF XPath 式を示します。

#### コード リスト 3-7 トレーディング パートナ名を指定する XOCP XPath 式

---

```
msg1.setExpression("//trading-partner[@name='Partner2\']");
```

---

次のリストは、San という文字列を含むアドレスのトレーディング パートナを選択する XOCP XPath 式を示します。

#### コード リスト 3-8 トレーディング パートナのアドレスを指定する XOCP XPath 式

---

```
msg.setExpression("//trading-partner[contains(address,\ 'San\')]");
```

---

## 拡張プロパティを使用した XOCP XPath 式の指定

拡張プロパティはユーザ定義の要素、属性、およびテキストで、仲介機能のリポジトリにあるトレーディング パートナと関連付けることができます。これらのプロパティは、リポジトリにある標準の定義済み属性にアプリケーション拡張を提供します。拡張プロパティのセットは、XML ドキュメント内のサブツリーとして取得できるようにリポジトリ内でモデル化されます。拡張プロパティは、**WebLogic Integration B2B Console** の [トレーディング パートナ] タブでコンフィグレーションされます。詳細については、『*B2B Integration Administration Console* オンラインヘルプ』の「B2B Integration のコンフィグレーション」の「高度なトレーディング パートナのコンフィグレーション オプションの使用」を参照してください。

XOCP XPath 式は、こうした拡張プロパティを参照してビジネス メッセージのルーティングに役立てることができます。たとえば、セラーが仲介機能のリポジトリに自身の扱える最大の数量を示せるように、仲介機能の管理者によって **Maximum Order Quantity** という拡張プロパティが追加されるとします。このプロパティが定義されれば、大量の注文をするバイヤは注文を処理できるセラーのみにビジネス メッセージを送信する XOCP XPath 式を指定することができます。

次のコードは、指定されたセラーについて拡張プロパティが設定されているリポジトリから生成された XML ドキュメントを示します。

コード リスト 3-9 リポジトリから生成された XML ドキュメントに設定されている拡張プロパティ

---

```
<hub context="message-router">
  . . .
  <trading-partner name="ABC Seller"
    email="orderprocessing@somedomain.com"
    phone="999-999-9999">
    <address>123 Main St., San Jose, CA 95131</address>
    <extended-property-set name="Capacity">
    <max-order-quantity>1000</max-order-quantity>
    </extended-property-set>
  </trading-partner>
  . . .
</hub>
```

---

次のリストは、500 単位よりも多くの注文に対応できるトレーディング パートナを選択する XOCPath 式を示します。

コード リスト 3-10 注文の量を指定する XOCPath 式

---

```
msg.setExpression("//trading-partner[extended-property-set/(@max-order-qty > \'500\')]")
```

---

セラーは 1000 単位までの注文に対応できるので、このセラーがビジネス メッセージの受信側として選択されます。

## 手順 3 : メッセージ配信のサービス品質の指定

B2B エンジンのメッセージング サービスによって、XOCPath アプリケーションはサービス品質 (QoS)、つまり信頼性のレベルを定義して、受信側トレーディング パートナへビジネス メッセージを配信する際に適用することができます。サービス品質の設定は、ビジネス メッセージのメッセージ ヘッダに格納されます。メッセージング サービスは、ネットワークでリンクまたはノードの障害が

発生した場合も信頼性のあるメッセージ配信をサポートします。メッセージングサービスは信頼性のあるメッセージをサポートするために他の機能を提供します。たとえばメッセージのロギングとトラッキング、メッセージの相関、配信の再試行、メッセージのタイムアウト、およびメッセージ配信方法の選択などです。

## 自動的なサービス品質機能

B2B エンジンのメッセージング サービスは、XOCP アプリケーションからの入力が必要としない、以下の自動的なサービス品質機能を提供します。

- B2B エンジンは、重複したメッセージ配信を防止します。
- B2B エンジンは、すべてのビジネス メッセージについて、それが仲介機能または XOCP アプリケーション ノードに届いた時点でタイムスタンプを付与します。タイムスタンプは、パフォーマンスの測定やアプリケーションのデバッグに有用です。

## QualityOfService クラス

ビジネス メッセージのサービス品質設定は、`com.bea.b2b.protocol.xocp.messaging.QualityOfService` クラスによって表されます。QualityOfService クラスは、B2B エンジンのメッセージング サービスが特定のメッセージを配信する際に必要とする信頼性のレベルを定義します。また、B2B エンジンのメッセージング サービスに対して、XOCP アプリケーションで想定されているビジネス メッセージ配信方法を示します。

XOCP アプリケーションはこのクラスのインスタンスを作成してから、このインスタンス上のメソッドを呼び出してさまざまなサービス品質設定を指定します。次に XOCP アプリケーションはメッセージ インスタンスの `setQoS` メソッドを呼び出し、このとき QualityOfService オブジェクトをパラメータとして渡すことで、この設定をメッセージと関連付けます。XOCP アプリケーションでサービス品質設定が指定されないと、B2B エンジンのメッセージング サービスはデフォルト値を使用します。

## サービス品質設定、オプション、およびデフォルト値

次の表は、使用できるサービス品質設定、オプション、およびデフォルト値を示します。

表 3-1 サービス品質設定、オプション、およびデフォルト値

サービス品質設定 / 説明	オプション	デフォルト値
CONFIRMED_DELIVERY_TO_APPLICATION  ◆ 受信側アプリケーションまでのアプリケーション配信の確認を提供。  ◆ 各送り先への配信の詳細なステータスを提供。受信のタイムスタンプ、ルータで選択されたトレーディングパートナーのリスト、受信側トレーディングパートナーの最終リストなど。  ◆ 仲介機能の管理者およびメッセージを送信するトレーディングパートナーの管理者のために、詳細なメッセージトラッキング情報（すべての考え得る位置について）を提供。	該当なし	該当なし
CONFIRMED_DELIVERY_TO_DESTINATION(S)  ◆ 各送り先への配信の詳細なステータスを提供。受信のタイムスタンプ、ルータで選択されたトレーディングパートナーのリスト、受信側トレーディングパートナーの最終リストなど。  ◆ 仲介機能の管理者およびメッセージを送信するトレーディングパートナーの管理者のために、詳細なメッセージトラッキング情報（すべての考え得る位置について）を提供。	適用不可	適用不可

### 3 XOCP ビジネス メッセージの送信

表 3-1 サービス品質設定、オプション、およびデフォルト値 (続き)

サービス品質設定 / 説明	オプション	デフォルト値
CONFIRMED_ROUTING  ◆ ビジネス メッセージの受信側として選択されたトレーディング パートナについての、仲介機能にある XOCP ルータからの情報を提供。  ◆ メッセージを送信するトレーディング パートナの管理者に対してメッセージのトラッキングを提供 (ビジネス メッセージが仲介機能の XOCP ルータに到達するまで)。	適用不可	適用不可
CONFIRMED_DELIVERY_TO_HUB (デフォルト)  ◆ メッセージが仲介機能に達したことを検証。  ◆ メッセージを送信するトレーディング パートナにはメッセージトラッキングなし。	適用不可	適用不可
TIMEOUT	送信後のタイムアウト値 (ミリ秒単位)	無視される
RETRY_ATTEMPTS	0-n	仲介機能でのコンフィグレーションに従う
CORRELATION_ID	アプリケーション定義のフィールド	無視される

次の表は、サービス品質設定がメッセージのトラッキングおよび配信の確認応答に与える影響を示します。

表 3-2 サービス品質設定によるメッセージのトラッキングおよび確認応答への影響

サービス品質設定	メッセージのトラッキング (あり/なし)	確認応答 (あり/なし)
CONFIRMED_DELIVERY_TO_APPLICATIONS	あり	あり
CONFIRMED_DELIVERY_TO_DESTINATION(S)	あり	あり

表 3-2 サービス品質設定によるメッセージのトラッキングおよび確認応答への影響

サービス品質設定	メッセージのトラッキング (あり/なし)	確認応答 (あり/なし)
CONFIRMED_DELIVERY_TO_ROUTER	あり	なし
CONFIRMED_DELIVERY_TO_HUB	なし	なし

CONFIRMED\_DELIVERY\_TO\_DESTINATION(S) の設定が使用される場合は、詳細なメッセージトラッキングが使用でき、応答確認を使用すればメッセージが確実に送り先まで配信されたことを確認できます。

CONFIRMED\_DELIVERY\_TO\_HUB の設定が使用される場合は、メッセージトラッキングが使用できず、受信側レーディング パートナから確認応答が送信されることもありません。

## コード例

次のリスト例は、ビジネス メッセージのサービス品質を設定する方法を示します。

### コード リスト 3-11 ビジネス メッセージのサービス品質の設定

```
// 関連するインポート
import com.bea.b2b.protocol.xocp.messaging.XOCPMessage;
import com.bea.b2b.protocol.xocp.messaging.QualityOfService;

XOCPMessage msg = ...
// QoS オブジェクトを作成
QualityOfService qos = new QualityOfService();
// 送り先への確認済み配信を指定
qos.setConfirmedDeliveryToDestination(true);
msg.setQoS(qos);
```

## メッセージ配信確認レベルの設定

メッセージ配信確認のレベルを指定するために、XOCP アプリケーションは QualityOfService インスタンスにある以下のメソッドのいずれか呼び出します。このとき、Boolean 型の true パラメータを渡して必要なオプションを有効にします。

表 3-3 メッセージ配信確認のレベル

持続性レベル	説明
setConfirmedDeliveryToDestination	メッセージ配信を送り先まで確認する (true)、または仲介機能までのみ確認する (false) を指定。
setConfirmedDeliveryToHub	メッセージ配信を仲介機能まで確認する (true)、または確認しない (false) を指定。
setConfirmedDeliveryToRouter	メッセージ配信を仲介機能の XOCP ルータまで確認する (true)、または仲介機能までのみ確認する (false) を指定。
setConfirmedDeliveryToApplication	ビジネス メッセージのサービス品質を設定する。メッセージ配信を対象のアプリケーションまで確認する (true)、または仲介機能までのみ確認する (false) を指定。

次のリスト例は、メッセージ確認レベルを送り先までに設定する方法を示します。

### コード リスト 3-12 メッセージ配信確認レベルの設定

---

```
qos.setConfirmedDeliveryToDestination(true);
```

---

メッセージ配信確認の詳細については、3-19 ページの「手順 5: ビジネス メッセージの配信状態の確認」を参照してください。

## メッセージ タイムアウトの設定

指定された場合、メッセージ タイムアウトは送信側が応答確認を待機する時間を決めます。ビジネス メッセージが無効になった（タイムアウトした）場合、受信側はそれを処理せず、関連する他の処理もすべて、応答確認の処理および配信の再試行まで含めて放棄されます。

### タイムアウトのアルゴリズム

B2B エンジンには、そのコンポーネントによって使用されている別のクロックを同期しません。それは、こうしたクロックが、別の場所にある別のマシンのものである可能性があるためです。代わりに、B2B エンジンには相対的な時間アルゴリズムを使用します。

このアルゴリズムに基づいて、ビジネス メッセージがタイムアウトするまでの残り時間（コンポーネントがビジネス メッセージの処理を完了する絶対時刻との相対値）をビジネス メッセージに指定した上で、他のコンポーネントへ送信します。

受信側コンポーネントでは、タイムアウトの計算はメッセージの処理に残された時間に基づいて、絶対時刻（メッセージの到着を示す）と相対時間（メッセージ自身に埋め込まれている）の両方を使用して表現されます。このアルゴリズムにより、少なくとも、システムでの実際のメッセージ タイムアウトが、アプリケーションによって指定された本来のタイムアウトの後で発生することになります。

Message Timeout on the Hub = Message timeout specified by the XOCP application when sending a message

Message Timeout on the Sending XOCP Application = Message Timeout on the Hub +  $N \times \Delta$

以上の設定では、次のようになります。

- $N$  = システムで定義済みの数値、たとえば 10
- $\Delta$  = メッセージが送信側トレーディング パートナと仲介機能の間を行き来するために必要な時間の推定値

## 配信再試行の回数の設定

断続的なネットワーク障害によってビジネス メッセージの配信が失敗する場合、**B2B** エンジンのメッセージング サービスは、以下のいずれかに該当するまでビジネス メッセージの送信を繰り返し試みます。

- ビジネス メッセージが配信される（つまり、配信成功）
- 再試行の回数が超過した
- メッセージがタイムアウトした
- ビジネス メッセージの送信元である会話が終了したかタイムアウトした

メッセージ タイムアウトおよび再試行間隔のデフォルト値は仲介機能のリポジトリで定義されていて、**XOCP** アプリケーション セッションが作成されるときに **XOCP** アプリケーションによって取得されます。**B2B** エンジンのメッセージング サービスは、コンフィグレーションされた間隔の間だけ待機してから、ビジネス メッセージの再送信を試みます。

デフォルトの再試行制限をオーバーライドする場合、**XOCP** アプリケーションは `QualityOfService` インスタンスの `setTimeout` メソッドを呼び出します。このとき、パラメータとしてタイムアウト値（ミリ秒単位）が渡されます。次のリストを参照してください。

### コード リスト 3-13 メッセージ タイムアウトの指定

---

```
qos.setTimeout(10000);
```

---

## ビジネス メッセージの相関 ID の設定

**XOCP** アプリケーションは、ビジネス メッセージにユニークな相関 **ID** を指定して、トレーディング パートナから受信したビジネス メッセージ（要求への応答など）と以前に送信されたメッセージ（要求など）を関連付けることができます。相関 **ID** は、送り先までビジネス メッセージに付随します。受信側トレーディング パートナは、この値を使用して、送信側トレーディング パートナへ返信される応答メッセージを確実に識別することができます。

関連 ID を指定する場合、XOCP アプリケーションは QualityOfService インスタンスの `setCorrelationId` メソッドを呼び出します。このとき、パラメータとして関連 ID を表す文字列が渡されます。次のリストを参照してください。

### コード リスト 3-14 ビジネス メッセージの関連 ID の指定

---

```
qos.setCorrelationId("ABC123");
```

---

## 手順 4 : XOCP ビジネス メッセージの送信

ビジネス メッセージの受信側とサービス品質を指定した後で、XOCP アプリケーションはビジネス メッセージを以下のいずれかの方法で送信します。

- 同期メッセージ配信
- 遅延同期メッセージ配信

XOCP ビジネス メッセージを同期メッセージ配信または遅延同期メッセージ配信で送信するときは、以下の値を設定する必要があります。

- メッセージと関連付けられるコラボレーション アグリーメント ID
- 会話

## 同期メッセージ配信

同期メッセージ配信では、アプリケーションはメッセージが送り先まで配信されるのを待機します。B2B エンジンのメッセージング サービスは、メッセージ送信処理の結果が判明したらアプリケーションに制御を返します。アプリケーションは、以下のイベントのいずれかが発生するまで待機します。

- すべての送り先から応答確認を受信した
- メッセージがタイムアウトした
- メッセージの送信元である会話が終了した

ビジネス メッセージの同期送信を行うために、**XOCP** アプリケーションは **XOCPMessage** インスタンスの以下のメソッドを呼び出します。

- **setCAId** メソッド。コラボレーション アグリーメント **ID** を設定します。
- **setConversation** メソッド。ビジネス メッセージを送信する会話を設定します。
- **sendAndWait** メソッド。タイムアウトまで待機する時間の長さ（ミリ秒単位）を指定します。ゼロ（0）が指定された場合、**XOCP** アプリケーションはビジネス メッセージがそれぞれの送り先に到達するまで待機します。

次の例は、同期メッセージ配信を使用して **XOCP** ビジネス メッセージを送信する方法を示します。

---

#### コード リスト 3-15 同期メッセージ配信を使用したメッセージ送信

---

```
smsg1.setCAId(caId);  
smsg1.setConversation(c);  
MessageToken token = msg.sendAndWait(0);
```

---

## 遅延同期メッセージ配信

遅延同期メッセージ配信では、**B2B** エンジンのメッセージング サービスはメッセージ送信の直後に **XOCP** アプリケーションへ制御を返すほかにメッセージ トークンを返します。このメッセージ トークンは、**XOCP** アプリケーションがメッセージ配信の状態を確認するために使用できます。メッセージ トークンへのアクセスがあると、アプリケーションは指定された期間か、次のいずれかのイベントが発生するまで待機します。

- すべての送り先から応答確認を受信した
- メッセージがタイムアウトした
- メッセージの送信元である会話が終了したかタイムアウトした

ビジネス メッセージの非同期送信を行うために、**XOCP** アプリケーションは **XOCPMessage** インスタンスの以下のメソッドを呼び出します。

- **setCAId** メソッド。コラボレーション アグリーメント **ID** を設定します。

- `setConversation` メソッド。ビジネス メッセージを送信する会話を指定します。
- `send` メソッド。メッセージを送信します。

XOCP アプリケーションはビジネス ロジックの実行を継続し、`MessageToken` インスタンスの `waitForACK` メソッドを呼び出して状態を確認します。次のリストにその例を示します。

### コード リスト 3-16 遅延同期メッセージ配信を使用したメッセージ送信

---

```
smsg1.setCAId(caId);  
smsg1.setConversation(c);  
token = msg.send();  
...  
token.waitForACK();
```

---

`waitForACK` メソッドは、ビジネス メッセージの状態が利用可能になるか（タイムアウト指定がない場合）、指定されているタイムアウト（ミリ秒単位）が経過するまで処理をブロックします。

## 手順 5 : ビジネス メッセージの配信状態の確認

`XOCPMessage` インスタンスで呼び出される `send` および `sendAndWait` メソッドのどちらも、メッセージ トークンを返します。このメッセージ トークンは、XOCP アプリケーションが自身と関連付けられているビジネス メッセージの配信状態の確認に使用することができます。

## メッセージ トークン

メッセージトークンは、`com.bea.b2b.protocol.xocp.messaging.XOCPMessageToken` クラスのインスタンスです。メッセージトークンには、以下の属性があります。

表 3-4 メッセージトークンの情報

属性	説明
メッセージ ID	ビジネス メッセージの一意な ID
例外	ビジネス メッセージが送信元 XOCP アプリケーションを離れる前に発生したすべての例外。例外は通常、メッセージが送信されるときに返される。しかし遅延同期メッセージ配信では、ビジネスメッセージが一時的に内部の送信キューに留め置かれてから、仲介機能へ配信されることになる。
経過時間	ビジネス メッセージをすべての送り先へ配信するためにかかった時間。この情報は、すべてのメッセージ送り先から応答確認が受信されるまで入手できない。この属性を使用できるかどうかは、サービス品質配信オプションの指定に依存する。
配信状態	受信側の送り先からの配信状態。この情報は、関連する情報の可用性に依存する。この属性を使用できるかどうかは、サービス品質配信オプションの指定に依存する。
宛先の数 (ルータ)	仲介機能の XOCP ルータによってビジネスメッセージが処理された後の受信側トレーディングパートナーの数。この属性を使用できるかどうかは、サービス品質配信オプションの指定に依存する。
宛先の数 (フィルタ)	仲介機能の XOCP フィルタによってビジネスメッセージが処理された後の受信側トレーディングパートナーの数。この属性を使用できるかどうかは、サービス品質配信オプションの指定に依存する。

ビジネス メッセージが同期送信の配信オプションを使用して送信された場合、メッセージ トークンを使用して応答確認を待機することはできません。代わりに、`send` メソッドが直ちに返されます。

## 配信状態のトラッキング

ビジネス メッセージがその送り先（送り先トレーディング パートナ ノードの受信キュー）に届くと、システム メッセージが（B2B エンジンのメッセージング サービスによって）送信側へ返されて、サービス品質設定によって応答確認が必要な場合は、メッセージ配信の応答確認に使用されます。

XOCP アプリケーションは、以下のメソッドのいずれかを使用して配信状態を取得できます。

- `getAllDeliveryStatus`。ビジネス メッセージが複数の受信側へ送信された場合
- `getDeliveryStatus`。ビジネス メッセージが単一の受信側へ送信された場合

どちらのメソッドも `DeliveryStatus` オブジェクトを返します。このオブジェクトは、以下の情報を提供する

`com.bea.b2b.protocol.messaging.DeliveryStatus` クラスのインスタンスです。

- 受信側（受信側トレーディング パートナ の名前か、メッセージトラッキング場所）
- ビジネス メッセージ受信のタイムスタンプ
- 状態コード。有効な値を次の表に示します。

表 3-5 メッセージ配信状態コード

状態コード	説明
SUCCESS	ビジネス メッセージは送り先へ正常に配信された。エラーや例外は発生しなかった。
FAILURE	ビジネス メッセージを送り先まで配信する間にエラーが発生した。

表 3-5 メッセージ配信状態コード (続き)

状態コード	説明
RETRIES_EXHAUSTED	すべての配信再試行が失敗に終わり、ビジネスメッセージは破棄された。
TIMEDOUT	メッセージ配信よりも先にタイムアウトが発生してビジネスメッセージが破棄された。

## メッセージ トラッキング場所

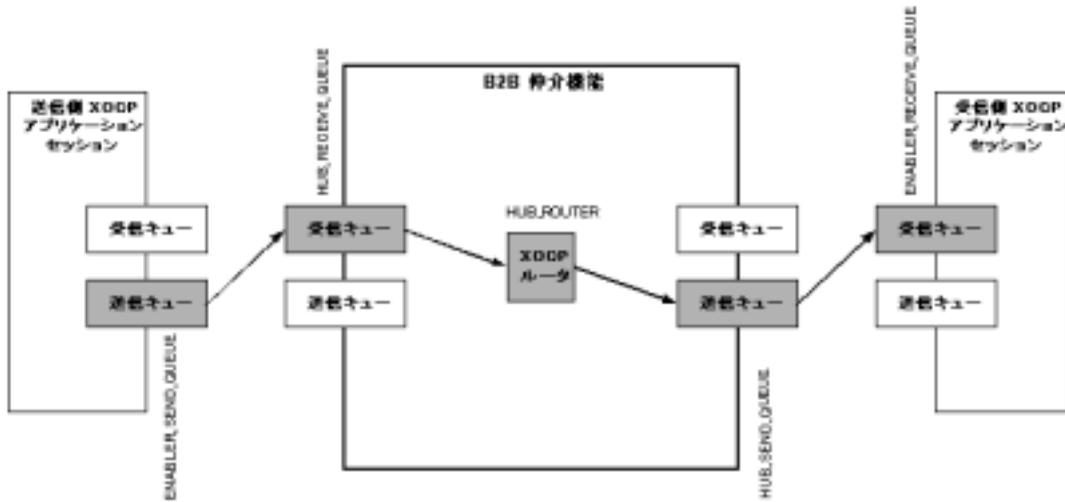
B2B エンジンのメッセージング サービスでは、管理者向けのトラッキング機能が用意されていて、ビジネスメッセージが自身の送り先までにあらかじめ定義された場所を移動していくの確認することができます。B2B Console には、ビジネスメッセージがこれらのトラッキング地点を通過していくステータス情報を表示できます。管理者はメッセージトラッキング情報を使用して、デバッグやアプリケーションのボトルネックの特定に活かすことができます。

**注意:** メッセージトラッキング場所を利用できるかどうかは、WebLogic Integration システムのコンフィグレーションと、ビジネスメッセージに指定されたサービス品質、たとえば CONFIRMED\_DELIVERY\_TO\_DESTINATION(S) などに依存します。サービス品質の設定の詳細については、表 3-1 を参照してください。たとえば、XOCP アプリケーションと仲介機能が同じノードに配置されている場合は、一部の場所が利用できません。同様に、トラッキング場所の一部は、同期メッセージ配信には使用できません。

## メッセージトラッキング場所の図

次の図は、B2B エンジンのメッセージング サービスでのメッセージトラッキング場所を示します。

図 3-1 メッセージトラッキング場所



## メッセージトラッキング場所の説明

以下のメッセージトラッキング場所を B2B Console で表示できます。

表 3-6 メッセージトラッキング場所

場所	場所の説明	実行されるアクティビティ
ENABLER_SEND_QUEUE	送信側トレーディング パートナの XOCP アプリケーション セッション内の送信キュー	メッセージが送信のためキューに入れられる。

### 3 XOCP ビジネス メッセージの送信

表 3-6 メッセージトラッキング場所 (続き)

場所	場所の説明	実行されるアクティビティ
HUB_RECEIVE_QUEUE	仲介機能の送信側トレーディング パートナ用の受信キュー	メッセージは仲介機能の送信側トレーディング パートナの受信キューに入れられる。
HUB_ROUTER	仲介機能の XOCP ルータ	メッセージが XOCP ルータに到達する。
HUB_SEND_QUEUE	仲介機能の受信側トレーディング パートナ用の送信キュー	メッセージは仲介機能の受信側トレーディング パートナで配信用のキューに入れられる。
ENABLER_RECEIVE_QUEUE	受信側トレーディング パートナの XOCP アプリケーションセッション内の受信キュー	メッセージは受信側トレーディング パートナの XOCP アプリケーションセッションのリスナスレッドでキューに入れられる。

---

## 4 XOCP ビジネス メッセージの受信

以下の節では、XOCP アプリケーションで XOCP ビジネス メッセージを受信する方法について説明します。

- XOCP ビジネス メッセージを受信する方法
- XOCP ビジネス メッセージの受信

### XOCP ビジネス メッセージを受信する方法

XOCP アプリケーションは、ビジネス メッセージを受信および処理するために `MessageListener` インタフェースの `onMessage` メソッドを実装する必要があります。 `onMessage` メソッドのシグネチャは次のとおりです。

---

#### コード リスト 4-1 `onMessage` メソッドのシグネチャ

---

```
public void onMessage(XOCPMessage msg)
```

---

ビジネス メッセージが届くと常に、XOCP アプリケーションは `onMessage` メソッドを呼び出し、入力パラメータとしてビジネス メッセージを渡します。XOCP アプリケーションはビジネス メッセージを格納している `XOCPMessage` オブジェクトを取得してから、このインスタンス上でメッセージを処理するメソッドを呼び出します。

XOCP アプリケーションが会話の中で複数のビジネス ドキュメントを受信した場合、`onMessage` の実装ではまず受信したドキュメントの文書型が判別され（たとえば入札要求や入札報酬など）、それに応じてドキュメントが処理されます。

また、onMessage の実装は、ビジネス メッセージを作成および送信するコードも含む場合があります。たとえば、会話参加者の XOCP アプリケーションでは、onMessage で要求を受信したり、要求を処理したり、返信のドキュメントを作成および返信するように実装されます。

# XOCP ビジネス メッセージの受信

コード リスト 4-2 は、Messaging API サンプルアプリケーションである MdmTp2Servlet での onMessage メソッドの実装方法を示します。この onMessage の実装は、MsmTp1Servlet から送信された初期ビジネスドキュメント（要求）を処理します。次にこの実装は、会話開始者への応答のドキュメントを作成および返信します。

次のリストは、Messaging API サンプルアプリケーションである MdmTp2Servlet での onMessage メソッドの実装です。

### コード リスト 4-2 MdmTp2Servlet での onMessage の実装

---

```
public void onMessage(XOCPMessage rmsg) {
    try{

        QualityOfService qos = rmsg.getQoS();

        PayloadPart[] payload = rmsg.getPayloadParts();
        Document rq = null;

        // シングル パート ドキュメントを使用
        if (payload != null && payload.length > 0){
            BusinessDocument bd = (BusinessDocument)payload[0];
            rq = bd.getDocument();
        }
        if (rq == null){
            throw new Exception("Did not get a request document");
        }
        Conversation conv = rmsg.getConversation();

        Element root = rq.getDocumentElement();
        String name = root.getNodeName();
        Text revStr = (Text)root.getFirstChild();
```

```
// 応答のドキュメントを作成
DOMImplementationImpl domi = new DOMImplementationImpl();
DocumentType dType = domi.createDocumentType("reply", null, "reply.dtd");
rq = new DocumentImpl(dType);
root = rq.createElement("reply");
String sendStr = new String(revStr.getData());
sendStr="Partner2 -- " + sendStr;
root.appendChild(rq.createTextNode(sendStr.toLowerCase()));
rq.appendChild(root);

XOCPMessage msg = new XOCPMessage("");
msg.addPayloadPart(new BusinessDocument(rq));

msg.setQoS(qos);

//TradingPartnerFilter filter = new TradingPartnerFilter("Partner1");
msg.setExpression("//trading-partner[@name='Partner1']");
msg.setCAId(rmsg.getCAId());
msg.setConversation(conv);

msg.sendAndWait(0);

} catch (Exception e) {
    e.printStackTrace();
}
}
```

---

onMessage のコードは、以下の主要なタスクを実行します。

1. XOCPMessage インスタンスの getQoS メソッドを呼び出して、ビジネスメッセージのサービス品質設定を取得します。  
アプリケーションは、元のメッセージで使用されたのと同じサービス品質設定で応答メッセージを送信します。
2. XOCPMessage インスタンスの getPayloadParts メソッドを呼び出して、ビジネスメッセージのペイロード部分を取得します。
3. PayloadPart[] 配列で最初の（そして唯一の）ビジネスドキュメントを取得します。
4. BusinessDocument インスタンスの getDocument メソッドを呼び出して、関連付けられた XML ドキュメントを抽出します。

5. Document インスタンスの各メソッドを使用して、XMLドキュメントの各部を取得および分析します。XMLドキュメントは、World Wide Web Consortium ([www.w3.org](http://www.w3.org)) によって公開されている `org.w3c.dom` パッケージにある `org.w3c.dom.Document` クラスのインスタンスです。

また、XOCP アプリケーションはサードパーティによって実装された同様のパッケージも使用できます。たとえば、Apache XML Project ([www.apache.org](http://www.apache.org)) で提供されている `org.apache.xerces.dom` パッケージなどです。このパッケージは、Messaging API のサンプルアプリケーションでビジネスドキュメントの作成および処理に使用されています。

6. ビジネスドキュメントに埋め込まれているデータ文字列 ("ABCDEFGHI") を取得して、すべて小文字に変換します。
7. 応答のドキュメントを作成してから、要求のドキュメントに指定されていたのと同じサービス品質設定を指定します。
8. コラボレーション アグリーメント ID および会話を設定してから、ドキュメントを Partner 1 という名前のトレーディング パートナへ送信します。

---

# 索引

## A

ACTIVE ステート 1-13

API

    Messaging API 1-5

## C

com.bea.b2b.protocol.xocp.application

    パッケージ 1-5

## D

DROPPED OUT ステート 1-13

## M

Messaging API クラス ライブラリ

    インタフェースの実装 2-2

    概要 1-5

    トレーディング パートナの追加  
        1-17

Messaging API クラス ライブラリのイン

    タフェースの実装 2-2

## R

REGISTERED ステート 1-13

## S

Secure Sockets Layer (SSL) 1-14

## X

XML ドキュメント、作成 3-2

XOCP アプリケーション

    Messaging API 1-5

    XML ドキュメントの作成 3-2

    XOCP アプリケーションセッション  
        の作成 1-14

    XOCP アプリケーションについて  
        1-3

    XOCP ビジネス メッセージの作成  
        3-4

XPath 式の指定 3-6

    アプリケーション ステップ 2-1

    会話からの退出 1-18

    会話でのロールの登録 1-15

    会話の開始 1-17

        ãJén 1-17

    会話の終了 1-18

    実行時の情報フロー 1-20

    受信側の指定 3-5

    主要なタスク 1-14

    添付ファイルの作成 3-4

    トレーディング パートナの指定 3-6

    ビジネス メッセージの作成 3-1

XOCP アプリケーション セッション

    作成 1-14

XOCP ビジネス メッセージ

    コンポーネント 1-7

    図 1-7

XPath 式 3-6

## い

印刷、製品のマニュアル x

## か

会話

開始 1-17

開始者 1-10

関与 1-16

コーディネータ 1-12

参加者 1-10

終了 1-18

退出 1-18

ロールの登録 1-15

会話の開始 1-17

会話の終了 1-18

会話への関与 1-16

拡張プロパティ 3-8

カスタマ サポート情報 xi

関連情報 x

## く

グローバル会話コーディネータ 1-12

## さ

再試行

回数 3-16

作成

XMLドキュメント 3-2

XOCP ビジネス メッセージ 3-4

添付ファイル 3-4

ペイロード部分 3-2

サービス品質

QualityOfService クラス 3-10

値 3-11

オプション 3-11

再試行 3-16

自動的な機能 3-10

設定 3-11

相関 ID 3-16

メッセージ タイムアウト 3-15

メッセージ配信確認 3-14

サポート

テクニカル xi

## し

受信

ビジネス メッセージ 4-1

受信側

XPath 式 3-6

指定 3-5

トレーディング パートナ 3-6

状態、トレーディング パートナ 1-13

## せ

セキュアなメッセージング 1-14

## そ

相関 ID 3-16

送信

ビジネス メッセージ 3-17

## た

退出

会話 1-18

タイムアウト

メッセージ タイムアウト 3-15

## ち

遅延同期 メッセージ配信 3-18

## て

添付ファイル

作成 3-4

---

## と

同期メッセージ配信 3-17

登録

会話でのロール 1-15

トラッキング

配信状態 3-21

トレーディング パートナ

状態 1-13

追加 1-17

トレーディング パートナの追加 1-17

情報フロー 1-8

説明 1-6

メッセージ配信

確認 3-14

遅延同期 3-18

同期 3-17

メッセージ配信の確認 3-14

## ろ

ローカル会話コーディネータ 1-13

## は

配信

回数 3-16

状態、トラッキング 3-21

パッケージ

com.bea.b2b.protocol.xocp.application  
n 1-5

## ひ

ビジネス メッセージ

作成 3-1

受信 4-1

説明 1-6

送信 3-17

## へ

ペイロード部分

作成 3-2

追加 3-4

## め

メッセージ

タイムアウト 3-15

トークン 3-20

トラッキング場所 3-22

メッセージ エンベロープ

