



お願い

本書および本書で紹介する製品をご使用になる前に、247 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM WebSphere Partner Gateway Enterprise Edition (製品番号 5724-L69) バージョン 6.2、リリース 0、モディフィケーション 0、および Advanced Edition (製品番号 5724-L68) バージョン 6.2、リリース 0、モディフィケーション 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： WebSphere® IBM WebSphere Partner Gateway Enterprise and Advanced Editions  
Version 6.2  
Enterprise Integration Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2010.4

© Copyright International Business Machines Corporation 2007, 2008.

# 目次

<b>第 1 章 本書について</b> . . . . .	<b>1</b>
対象読者 . . . . .	1
表記上の規則 . . . . .	1
関連文書 . . . . .	2
本リリースの新機能 . . . . .	3
リリース 6.2 の新機能 . . . . .	3
<b>第 2 章 バックエンド統合の概要</b> . . . . .	<b>5</b>
概説 . . . . .	5
文書処理の概要 . . . . .	5
バックエンド統合の概要 . . . . .	10
バックエンド統合の計画 . . . . .	11
ビジネス・プロトコルの選択 . . . . .	11
使用するパッケージ化 . . . . .	28
使用するメッセージ・トランスポート . . . . .	40
バックエンド・アプリケーションへのアクセス方法 . . . . .	49
メッセージ処理 . . . . .	50
WebSphere Partner Gateway の構成 . . . . .	51
<b>第 3 章 WebSphere Process Server との統合</b> . . . . .	<b>61</b>
WebSphere Process Server 統合の概要 . . . . .	61
概説 . . . . .	61
WebSphere Process Server との統合の計画 . . . . .	63
WebSphere Partner Gateway と WebSphere Process Server を統合するための作業の概要 . . . . .	66
バックエンド統合パッケージ化メッセージの処理 . . . . .	68
BCGBackEndIntegrationDataBindingUtil クラス . . . . .	82
BCGBackEndIntegrationJMSDataBindingImpl クラス . . . . .	87
コンポーネント実装へのデータ・バインディング・クラスの組み込み . . . . .	89
トランスポートとしての HTTP と WebSphere Process Server の統合 . . . . .	90
WebSphere Process Server へのメッセージの送信方法 . . . . .	90
WebSphere Process Server からのメッセージの送信方法 . . . . .	92
WebSphere Process Server への文書の送信 . . . . .	94
WebSphere Process Server からの文書の送信 . . . . .	96
トランスポートとしての JMS と WebSphere Process Server の統合 . . . . .	98
JMS トランスポート・プロトコルを使用した文書の送信 . . . . .	98
JMS トランスポート・プロトコルを使用した文書の受信 . . . . .	99
WebSphere Application Server での WebSphere Process Server 環境のセットアップ . . . . .	101

WebSphere Application Server 上に WebSphere Partner Gateway をインストール中の JMS の構成 . . . . .	104
WebSphere Process Server と SOAP/HTTP の統合 . . . . .	110
HTTP プロトコルを使用した SOAP メッセージの交換方法 . . . . .	110
WebSphere Process Server によってホストされる Web サービスの起動 . . . . .	112
外部パートナーによってホストされる Web サービスの起動 . . . . .	113
トランスポートとしてのファイル・システムと WebSphere Process Server の統合 . . . . .	114
ファイル・システム・プロトコルを使用した文書の送信 . . . . .	115
ファイル・システム・プロトコルを使用した文書の受信 . . . . .	115
WebSphere Process Server 環境の設定 . . . . .	115
WebSphere Partner Gateway 環境の設定 . . . . .	117

<b>第 4 章 WebSphere InterChange Server との統合</b> . . . . .	<b>119</b>
InterChange Server 統合の概要 . . . . .	119
InterChange Server との統合の計画 . . . . .	120
WebSphere Partner Gateway を InterChange Server 用に構成する . . . . .	123
InterChange Server の構成 . . . . .	126
添付ファイルのある文書の処理 . . . . .	131
HTTP を通じての InterChange Server の統合 . . . . .	152
ICS での HTTP トランスポート・プロトコルの使用 . . . . .	152
HTTP/S を通じた SOAP 文書の送信 . . . . .	167
JMS を通じての InterChange Server との統合 . . . . .	170
JMS トランスポートを通じた文書に必要なコンポーネント . . . . .	170
JMS トランスポート用環境の設定 . . . . .	176
JMS 用のビジネス・オブジェクト定義の作成 . . . . .	179
JMS 用の ICS 成果物の作成 . . . . .	184

<b>第 5 章 その他のバックエンド・システムとの統合</b> . . . . .	<b>187</b>
WebSphere Message Broker との統合 . . . . .	187
WebSphere Message Broker との統合の計画 . . . . .	188
WebSphere Partner Gateway を WebSphere Message Broker 用に構成する . . . . .	189
WebSphere Message Broker の構成 . . . . .	192
WebSphere Message Broker での HTTP トランスポート・プロトコルの使用 . . . . .	193
SOAP 文書の送信 . . . . .	197
WebSphere Message Broker での JMS トランスポート・プロトコルの使用 . . . . .	198

WebSphere Data Interchange との統合	205
本書の対象読者	205
本章とともに使用できるリソース	206
概要	206
WebSphere Data Interchange との統合の計画	209
メッセージ交換用の環境の構成	209
WebSphere Partner Gateway - Express の構成	226
要約	228
<b>第 6 章 その他の製品との統合</b>	<b>229</b>
WebSphere Transformation Extender との統合	229
概要	229
WebSphere Transformation Extender との統合の計画	230
WebSphere Message Broker と WebSphere Transformation Extender の統合	232
WebSphere Transformation Extender ユーザー出口の統合	232
WebSphere Transformation Extender Trading Manager との統合	238
概要	238
WebSphere Transformation Extender Trading Manager との統合の計画	239
WebSphere Partner Gateway と WebSphere Transformation Extender Trading Manager 間の共通構成	239

WebSphere Partner Gateway の構成	239
WebSphere Transformation Extender Trading Manager の構成	241

## 第 7 章 その他のビジネス・プロトコル 固有の情報

EDI 文書のルーティング	243
EDI ルーティングの概要	243
AS パッケージ化に関する特別な考慮事項	245
インバウンド文書の処理方法	245
アウトバウンド文書の処理方法	245
パートナー・プロファイルの 2 つの ID の設定	245
AS3 MDN 処理	246
AS に関する他のリファレンス	246
RosettaNet パッケージ化に関する特別な考慮事項	246
RosettaNet に関する他のリファレンス	246
ebMS パッケージ化に関する特別な考慮事項	246
ebMS に関する他のリファレンス	246

## 特記事項

プログラミング・インターフェース情報	249
商標	249

## 索引

---

## 第 1 章 本書について

本書では、バックエンド・システムと IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Partner Gateway が通信するために使用するメカニズムであるバックエンド統合インターフェースについて説明します。本書では、さらに、バックエンド統合インターフェースを使用して、WebSphere Process Server、WebSphere InterChange Server、WebSphere Message Broker、および WebSphere Data Interchange を WebSphere Partner Gateway と統合する方法についても説明します。

本書の内容は、WebSphere Partner Gateway Enterprise および Advanced Edition のみに関連します。

---

### 対象読者

本書の対象読者は、WebSphere Partner Gateway とバックエンド・システムとの統合を担当されている方です。

---

### 表記上の規則

本書では、次のような規則を使用しています。

表 1. 表記上の規則

規則	説明
モノスペース・フォント	このフォントのテキストは、ユーザーが入力するテキスト、引き数またはコマンド・オプションの値、例およびコード・サンプル、またはシステムが画面に表示する情報 (メッセージ・テキストまたはプロンプト) を示します。
太字	太字体のテキストは、グラフィカル・ユーザー・インターフェース・コントロール (例えば、オンライン・ボタン名、メニュー名、またはメニュー・オプションなど)、および表やテキストの列見出しを示します。
イタリック	イタリックのテキストは、強調、書籍名、新規用語および本文で定義されている用語、変数名、または文字として使用されるアルファベット文字を示します。
イタリック・モノスペース・フォント	イタリック・モノスペース・フォントのテキストは、モノスペース・フォントのテキスト内に記載された変数名を表します。
<i>ProductDir</i>	<i>ProductDir</i> は、製品のインストール先のディレクトリーを表します。すべての IBM WebSphere Partner Gateway 製品のパス名は、システム上で IBM WebSphere Partner Gateway 製品がインストールされているディレクトリーからの相対パスです。

表 1. 表記上の規則 (続き)

規則	説明
%text% および \$text	パーセント記号 (%) に囲まれたテキストは、Windows <sup>(R)</sup> の text システム変数またはユーザー変数の値を示します。UNIX <sup>(R)</sup> 環境での同等の表記は、\$text であり、UNIX の text 環境変数の値を示します。
下線付きのカラー・テキスト	下線付きのカラー・テキストは、相互参照を示します。テキストをクリックすると、参照先のオブジェクトに移動します。
青のアウトラインのテキスト	(PDF ファイルのみ) テキストの周りの青のアウトラインは、相互参照を示します。アウトラインで囲まれたテキストをクリックすると、参照先のオブジェクトに移動します。この規則は、この表に記載されている「下線付きのカラー・テキスト」の規則の PDF ファイルの場合に相当します。
『 』 (かぎ括弧)	(PDF ファイルのみ) かぎ括弧により、本書の他のセクションへの相互参照を囲んでいます。
{ }	構文行では、中括弧によって囲まれた複数のオプションから、1 つのオプションだけを選択する必要があります。
[ ]	構文の記述行の場合、大括弧 [ ] で囲まれた部分は、オプション・パラメーターです。
< >	名前の変数要素は、不等号括弧で囲んで他の要素と区別しています。例えば、<server_name><connector_name>tmp.log です。
/, ¥	ディレクトリー・パスを区切る文字として、Windows インストール済み環境では円記号 (¥) を使用しています。UNIX システムの場合は、円記号をスラッシュ (/) に置き換えてください。

## 関連文書

この製品に付属する使用可能な資料の完全セットには、WebSphere Partner Gateway Enterprise および Advanced Edition のインストール、構成、管理および使用に関する包括的な情報が記載されています。

この資料は、次のサイトからダウンロードするか、オンラインで直接閲覧できます。

<http://www.ibm.com/software/integration/wspartnergateway/library/>

注: 本書の発行後に公開されたテクニカル・サポートの技術情報や速報に、本書の対象製品に関する重要な情報が記載されている場合があります。この資料は、以下の WebSphere Business Integration Support Web サイトにあります。

<http://www.ibm.com/software/integration/wspartnergateway/support>。関心のあるコンポーネント・エリアを選択し、「Technotes」セクションと「Flashes」セクションを参照してください。

---

## 本リリースの新機能

このセクションでは、IBM WebSphere Partner Gateway の新機能について説明します。

### リリース 6.2 の新機能

WebSphere Partner Gateway V6.2 は、以下の新機能をサポートします。

- WebSphere Partner Gateway の拡張フレームワークを使用した WebSphere Transformation Extender との統合
- ログ・ファイルの収集および伝送のための ISA V4 のサポート
- 証明書のアップロードおよび構成の機能強化
- メッセージの詳細情報があるエラー・メッセージへのリンク
- WebSphere Partner Gateway ファースト・ステップ・ページの機能強化
- 再配置および再デプロイメントのための WebSphere Partner Gateway 設定の更新用スクリプト
- WebSphere Partner Gateway コンポーネントのインストールの最後にインストール検査テスト (IVT) を実行可能
- WebSphere Partner Gateway の完全な構成をエクスポートおよびインポート可能
- 手動でのアップグレード作業を最小限にするための自動アップグレードのサポート
- スケジューラーの付いたコンソール・ベースのアーカイバー
- 既存の WebSphere Application Server セルへ統合可能
- セキュア・ファイル転送プロトコル (SFTP) のサポート
- ebXML メッセージ・サービス (ebMS) のための CPP/CPA エディター
- 機能拡張
  - アーカイバーのパフォーマンスの向上
  - AS2 およびラージ・ファイルの文書スループット・パフォーマンスの向上

6.2 での新機能について詳しくは、<http://www-01.ibm.com/software/integration/wspartnergateway/about/>を参照してください。



---

## 第 2 章 バックエンド統合の概要

以下の章では、WebSphere Partner Gateway とバックエンド・システムとの統合についての概要を説明します。さらにバックエンド統合計画のガイドラインについても説明します。

---

### 概説

この章では、バックエンド統合に関する以下の一般情報を提供します。

- 『文書処理の概要』
- 10 ページの『バックエンド統合の概要』

### 文書処理の概要

WebSphere Partner Gateway を使用して、外部パートナーとビジネス文書を交換します。これらの文書を交換する目的は、情報を伝達することです。そのためには、通常データを処理して結果を戻すことが必要になります。外部パートナーからデータを受け取ったとき、そのデータの処理は、一般に企業のバックエンド・システムで行われます。WebSphere Partner Gateway は、企業から送付されるメッセージまたは企業へ送付されるメッセージが経由するハブ・コミュニティ内部のポイントです。

企業は WebSphere Partner Gateway が接続されているバックエンド・システムを通じてアクセスされます。

6 ページの図 1 は、WebSphere Partner Gateway Enterprise および Advanced Edition での文書フローを示しています。パートナーは、WebSphere Partner Gateway (ハブ) に文書を送信します。WebSphere Partner Gateway は、文書を受信し、事前定義されたすべてのアクション (文書の検証または変換など) を実行します。次に、WebSphere Partner Gateway は、バックエンド・アプリケーションに文書を送信します。そこで文書が処理されます。

注: 図に示されているように、反対方向の通信フローも存在します。バックエンド・アプリケーションは、文書を生成し、その文書をハブに送信できます。ハブは、その文書を処理し、パートナーに送信します。

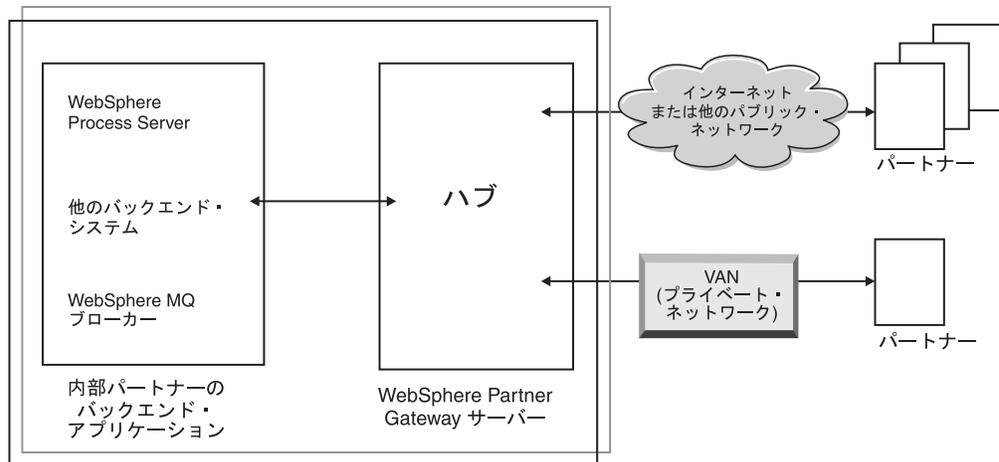


図1. エンドツーエンド文書フロー

本書では、ハブとバックエンド・アプリケーションとの統合を中心に説明します。

**注:** 本書の内容は、WebSphere Partner Gateway Enterprise および Advanced Edition に該当します。軽量で使いやすい B2B 接続ツールである WebSphere Partner Gateway - Express は、WebSphere Partner Gateway Enterprise および Advanced Edition とは異なります。このツールは、コミュニティー統合ソリューションを提供します (これに対して、WebSphere Partner Gateway Enterprise および Advanced Edition は、内部パートナー用のゲートウェイ・ハブ・ソリューションを提供します)。WebSphere Partner Gateway - Express について詳しくは、この製品の「WebSphere Partner Gateway ユーザー・ガイド」を参照してください。

## ハブ・コミュニティー内の役割

WebSphere Partner Gateway Enterprise および Advanced Edition には、ハブ管理者、内部パートナー、および外部パートナーの 3 つのパートナー・タイプがあります。ハブ管理者は、WebSphere Partner Gateway のインストール時に自動的に作成されます。ハブ管理者は、ハブのセットアップおよびハブと対話するパートナーの作成を担当します。

通常の場合、ハブの所有者が内部パートナーになります。内部パートナーは、実際にはハブのパートナーの 1 つであると見なされます。ハブ管理者は、内部パートナーのプロファイルを作成して、内部パートナーが他のパートナーとの間で文書を送受信するために必要な情報を提供します。ハブが文書をバックエンド・システムに送信するときには、内部パートナー用にセットアップされた情報 (URL または JMS キューなど) を使用します。ハブ管理者は、他のパートナーのプロファイルも作成できます。作成できるパートナーの数は任意です。

## ハブ構成プロセス

ハブ管理者は、ハブの管理を担当するユーザーです。ハブ管理者は、内部パートナーおよび外部パートナーとの間でビジネス文書を送受信するように、ハブをセットアップします。内部パートナーからビジネス文書を受信するために、ハブ管理者はトランスポートのレシーバーを作成します。内部パートナーは、このレシーバーを使用して文書を送信します。例えば、内部パートナーがファイル・ディレクトリー・トランスポートと JMS トランスポートを使用する場合、ハブ管理者は内部パ

ートナー用にファイル・ディレクトリーレシーバーと JMS レシーバーをセットアップします。同様に、外部パートナーが HTTP トランスポートと FTP トランスポートを使用する場合、ハブ管理者は外部パートナー用に HTTP レシーバーと FTP レシーバーをセットアップします。

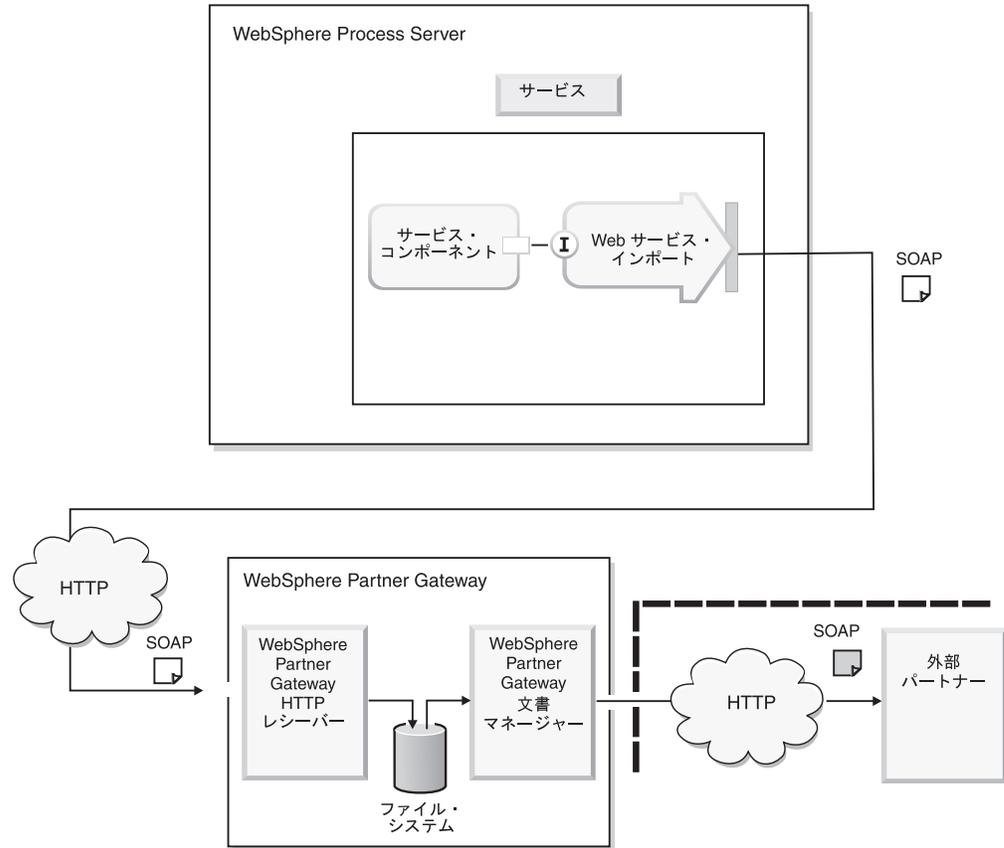


図2. 内部パートナーおよび外部パートナー用のレシーバー

内部パートナーおよび外部パートナー用に、それぞれのトランスポートに対応した宛先が作成されます。パートナーは、これらの宛先を使用して、ハブが送信したビジネス文書を受信します。

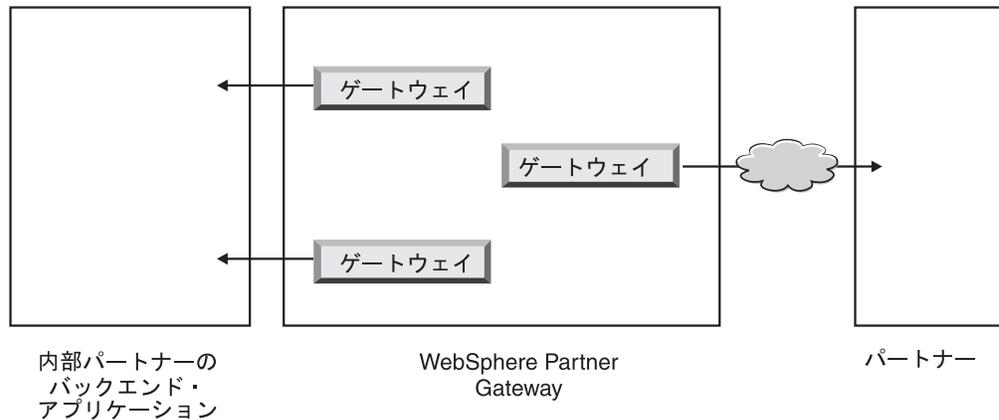


図3. 内部パートナーおよび外部パートナー用の宛先

ハブ管理者は、ハブ構成の一部として、文書フロー定義を設定します。これにより、文書フローの以下のような特性が定義されます。

- パッケージ化。文書のルーティングについての情報を指定します。
- プロトコル。文書が準拠するビジネス・プロトコルです。
- 文書フロー。文書自体を表します。

WebSphere Partner Gateway をインストールすると、文書定義のセットを使用できるようになります。独自に作成した定義またはアップロードした定義を文書定義に追加することもできます。例えば、インストール・メディア上には、各種の RosettaNet PIP に対する文書定義が ZIP ファイルとして格納されています。これらのファイルをアップロードして使用できます。EDI ファイルを交換している場合は、Data Interchange Services クライアントから文書定義と関連マップをインポートできます。

以下の例を考えます。外部パートナーが、RosettaNet PIP 3A4 の購入注文文書を含む RNIF 2.0 メッセージを、WebSphere Partner Gateway の HTTP レシーバーに送信します。このメッセージは内部パートナー用です。内部パートナーには、購入注文を処理するバックエンド・システムがあり、購入注文を受け付けます。この購入注文の主要な部分は、外部パートナーが送信した RNIF メッセージのペイロードです。WebSphere Partner Gateway 内で外部パートナー接続をセットアップする前に、以下の合意が成立していることを確認します。

- 外部パートナーは、RosettaNet PIP 3A4 の購入注文文書を含む RNIF メッセージを、HTTP を使用して送信します。
- WebSphere Partner Gateway は、着信メッセージからビジネス・ペイロードまたは RosettaNet サービス・コンテンツを抽出します。
- 文書は、JMS を使用してバックエンド・システムに転送されます。バックエンド統合パッケージ化が使用されます。
- その後、バックエンド・アプリケーションは、受信した文書を処理します。

バックエンド統合パッケージ化が使用される場合、文書交換に役立つ情報を伝達するために、WebSphere Partner Gateway で定義されたトランスポート・ヘッダーが文書に追加されます。

上記の例では、ハブ管理者が適切な PIP パッケージをアップロードして、RosettaNet PIP 3A4 の交換のために以下の文書定義をセットアップします。

- RNIF パッケージ化、RosettaNet プロトコル、および 3A4 PIP で構成されるフロー
- バックエンド統合パッケージ化、RNSC プロトコル、および 3A4 PIP で構成されるフロー

ハブ管理者は、文書定義を設定した後で、文書フロー定義用のインタラクションを作成します。例えば、ハブ管理者は、ソースからハブへの RNIF/RosettaNet/3A4 文書フロー定義の受け入れを指定できます。

ハブ管理者 (または外部パートナー) は、文書交換に適切な B2B 機能を選択します。この例では、内部パートナーが以下の B2B 機能を使用可能にします。

- パッケージ: バックエンド統合
- プロトコル: RNSC
- 文書フロー: 3A4

外部パートナーが以下の B2B 機能を使用可能にします。

- パッケージ: RNIF
- プロトコル: RosettaNet
- 文書フロー: 3A4

次に、ハブ管理者は、外部パートナー間の接続を作成します。

以下の図では、ハブ管理者が内部パートナーおよび外部パートナーのプロファイルを作成し、文書を受信するためのレシーバーと文書を送信するための宛先を作成しました。また、上記の文書フロー定義を作成し、外部パートナーおよび内部パートナーの B2B 機能を設定し、両者の間の接続を作成しました。



図4. バックエンド・システムへの文書フロー

ハブのセットアップについて詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

## バックエンド統合の概要

すべてのエディションの WebSphere Partner Gateway において、バックエンド・システムに接続できるようになっています。これらのエディションは、以下のように、サポート可能なトランスポート・プロトコルが異なっています。

- WebSphere Partner Gateway - Express ではファイル・ベースの統合を提供します。
- WebSphere Partner Gateway Enterprise および Advanced Edition では、ファイル・ベースの統合を提供します。さらには、HTTP、HTTPS、および JMS プロトコルにまたがる統合を提供します。

外部パートナーと WebSphere Partner Gateway の間で交換される文書には、さまざまなフォーマットを使用できます。SOAP、cXML、XML、EDI、レコード指向データ (ROD)/フラット・ファイル、バイナリーのフォーマット、または外部パートナー間で相互に合意した任意のカスタム・フォーマットの文書を交換可能です。

「*WebSphere Partner Gateway 管理者ガイド*」には、サポートされる文書タイプと、文書の送信に使用可能なトランスポート・プロトコル (HTTP など) の完全なリストが掲載されています。

WebSphere Partner Gateway と内部パートナーのバックエンド・システムの間で交換可能な文書、ならびに文書に関連付けられるトランスポート・タイプを、41 ページの表 16、41 ページの表 17、および 42 ページの表 18 に示します。

11 ページの図 5 に、WebSphere Partner Gateway がバックエンド統合インターフェースを使用して、内部パートナーにおいてバックエンド・システムと通信する方法を示します。矢印が両方向になっていることに注意してください。すなわち、内部パートナーのバックエンド・システムが文書の発信元になることも可能です。

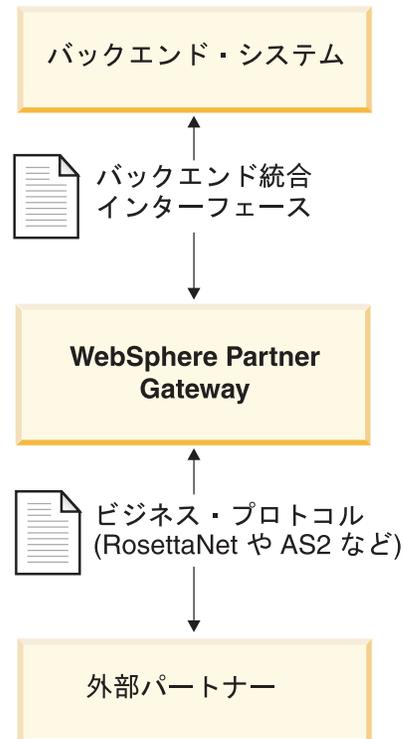


図5. 文書のフローにおけるビジネス・プロトコルおよびパッケージ化の役割

## バックエンド統合の計画

この章では、WebSphere Partner Gateway とバックエンド・システムとの統合を計画する方法を説明します。バックエンド統合を計画するときに決定するさまざまな事項を説明します。

- 『ビジネス・プロトコルの選択』
- 28 ページの『使用するパッケージ化』
- 40 ページの『使用するメッセージ・トランスポート』
- 49 ページの『バックエンド・アプリケーションへのアクセス方法』

以下の事項も説明します。

- 50 ページの『メッセージ処理』
- 51 ページの『WebSphere Partner Gateway の構成』

## ビジネス・プロトコルの選択

メッセージのビジネス・プロトコルによって、文書のフォーマットが決まります。ビジネス・プロトコルは、バックエンド・システムへの統合を計画するときに下す決定の多くに影響します。どのビジネス・プロトコルを選択するかによって、使用しなければならないパッケージ化方式が決まり、それが使用可能なメッセージ・トランスポート・プロトコルに影響します。

ビジネス・プロトコルの詳細については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。このセクションでは、以下のビジネス・プロトコルに固有の統合情報について説明します。

- 『Web サービス (SOAP)』
- 『cXML』
- 13 ページの『EDI』

注: EDI のセクションでは、XML およびレコード指向データ (ROD)/フラット文書の処理方法についても説明します。

- 18 ページの『RosettaNet』

## Web サービス (SOAP)

WebSphere Partner Gateway では、以下の Web サービスをハブ・コミュニティのメンバーに提供することができます。

- 内部パートナーが提供する Web サービスは、外部パートナーが利用できます。

外部パートナーに対し、WebSphere Partner Gateway が生成する Public WSDL を提供する必要があります。外部パートナーが Web サービスを呼び出す URL が、Web サービスをアップロードする間に指定された Web サービス・パブリック URL であることに注意する必要があります。WebSphere Partner Gateway はプロキシとして機能します。パートナーから SOAP メッセージを受信し、対応するプライベート Web サービスを見つけ出します。次に、同じ SOAP メッセージを使用して、プライベート Web サービス (内部パートナーが提供) を呼び出します。そして、内部パートナーが戻した応答が、外部パートナーに戻されます。

- 外部パートナーが提供する Web サービスは、内部パートナーが利用できます。

同じ Web サービス・インターフェースが複数のパートナーから提供されてもかまわないことに注意してください。WebSphere Partner Gateway は、Web サービスのアップロード時にコンソールで指定された Web サービス URL で、内部パートナーが Web サービスを利用できるようにします。さらに、内部パートナーは、「受信側パートナー」を示すための URL パラメーターを提供する必要があります。詳細については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。WebSphere Partner Gateway はプロキシとして機能します。内部パートナーから SOAP メッセージを受信し、対応する Web サービスおよび「受信側パートナー」を見つけ出します。次に、同じ SOAP メッセージを使用して、パートナーが提供する Web サービスを呼び出します。パートナーから戻された応答メッセージは、次に内部パートナーへ戻されます。

文書フロー定義を Web サービス用にセットアップする方法などの詳細については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

## cXML

外部パートナーへ cXML 文書を送信することも、外部パートナーから cXML 文書を受信することも可能です。WebSphere Partner Gateway は、外部パートナーから cXML 文書を受け取ると、その文書の妥当性を検査し、内部パートナーのバックエンド・システムへ送信する前に (指定があれば) 変換します。同期 cXML メッセージには変換を使用しないようにする必要があります。同期交換では、バックエン

ド・システムが応答を生成し、それを WebSphere Partner Gateway が外部パートナーに戻します (メッセージに該当する場合)。

cXML 文書を送信する必要のある内部パートナーのバックエンド・システムは、以下の 2 つの処理のいずれかを実行することができます。

- cXML 文書を生成および送信します。WebSphere Partner Gateway がこれを外部パートナーにパススルーします。
- XML 文書を生成および送信します。外部パートナーに送信する前に WebSphere Partner Gateway がこれを cXML に変換します。

注: 外部パートナーとの同期要求/応答トランザクションに XML 文書変換が使用される場合、応答は非同期的にバックエンド・システムに戻されます。

文書フロー定義を cXML 用にセットアップする方法などの詳細については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

## EDI

WebSphere Partner Gateway は、パートナーから付加価値通信網 (VAN) およびインターネット経由で送信される EDI 文書を受け入れます。VAN 経由で送受信される EDI 文書は、FTP スクリプト・トランスポートを使用します。FTP スクリプト・トランスポートは、インターネット経由で文書を送受信するときにも使用できます。FTP スクリプト・トランスポートについて詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

EDI 文書は、EDI エンベロープの形式でハブに着信し、ハブから発信されます。これは交換と呼ばれます。交換には、個々の EDI トランザクションまたはトランザクションのグループが含まれます。

EDI 交換が (エンベロープを解除しないで) ハブを通過する場合は、ハブと内部パートナーの間に 1 つの接続を作成します。

ただし、EDI 交換のエンベロープを解除する場合は、インタラクションと接続を作成するためのプロセスが、他のビジネス・プロトコルとは異なります。交換のエンベロープを解除し、個々のトランザクションを処理する必要があります。トランザクションは、通常、Data Interchange Services クライアントからインポートされた変換マップに基づいて、別の形式に変換されます。EDI トランザクションが XML またはレコード指向データ (ROD)/フラット文書に変換された場合は、これらの文書が内部パートナーまたは外部パートナーに送信されます。トランザクションが他の EDI フォーマットに変換された場合は、内部パートナーまたは外部パートナーに送信される前に、まずトランザクションがエンベロープされます。

**バックエンド・アプリケーションからパートナーへのフロー:** バックエンド・アプリケーションは、以下のタイプの文書を送信できます。

- 1 つ以上のトランザクションを含む単一 EDI 交換

WebSphere Partner Gateway は、個々の EDI トランザクションのエンベロープを解除し、これらの個々のトランザクションを変換します。トランザクションが EDI に変換された場合は、トランザクションがエンベロープされ、パートナーに転送されます。バックエンド・アプリケーションは、パッケージ化なしまたはバ

ックエンド・パッケージ化のどちらかを使用でき、41 ページの表 17 で定義される各種のトランスポートを使用して交換を送信できます。

図 6 は、3 つのトランザクションがエンベロープ解除される X12 交換を示しています。トランザクションは、EDIFACT フォーマットに変換されてから、エンベロープされ、パートナーに送信されます。

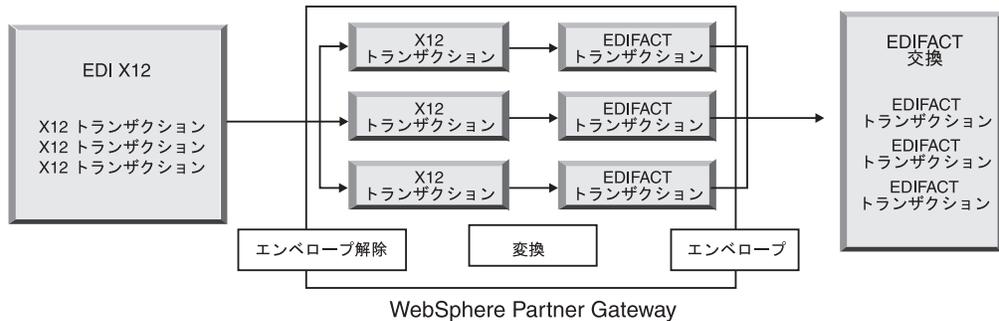


図 6. バックエンド・アプリケーションからパートナーへの EDI 交換

トランザクションのそれぞれに変換マップが関連付けられており、そのマップにはトランザクションをどのように変換するかが指定されています。トランザクションは、単一トランザクションに変換することができます。マップの作成時にマップ・チェーニングが使用されている場合は、複数のトランザクションに変換できます。

トランザクションが XML または ROD/フラット文書に変換された場合は、そのトランザクションの「パートナー接続」ビューでの構成に従ってトランザクションが転送されます。

図 7 は、EDI X12 交換のエンベロープが解除される様子、XML 文書に変換される様子、およびその文書がパートナーに送信される様子を示しています。

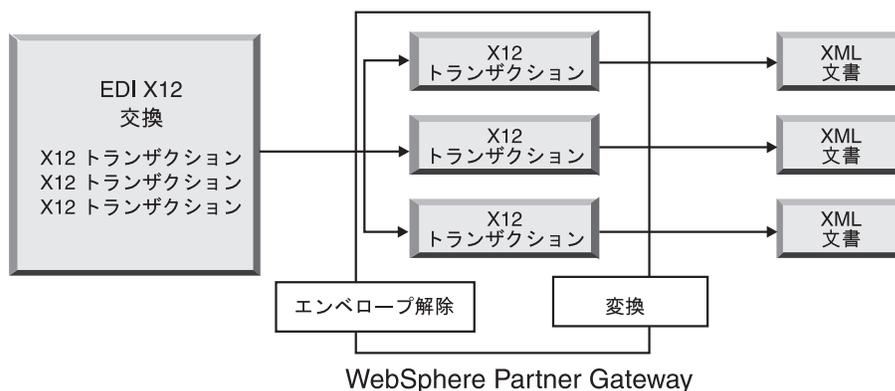


図 7. バックエンド・アプリケーションからパートナーへ (XML 文書として) 送信される EDI 交換

トランザクションは、単一文書に変換することができます。マップの作成時にマップ・チェーニングが使用されている場合は、複数の文書に変換できます。

- XML または ROD/フラット文書などの単一文書

WebSphere Partner Gateway は、文書を EDI トランザクションに変換し、トランザクションをエンベロープし、そのトランザクションをパートナーに送信します。バックエンド・アプリケーションは、パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用でき、41 ページの表 17 で定義される各種のトランスポートを使用して文書を送信できます。

図 8 は、X12 トランザクションに変換されてからエンベロープされる XML 文書を示しています。

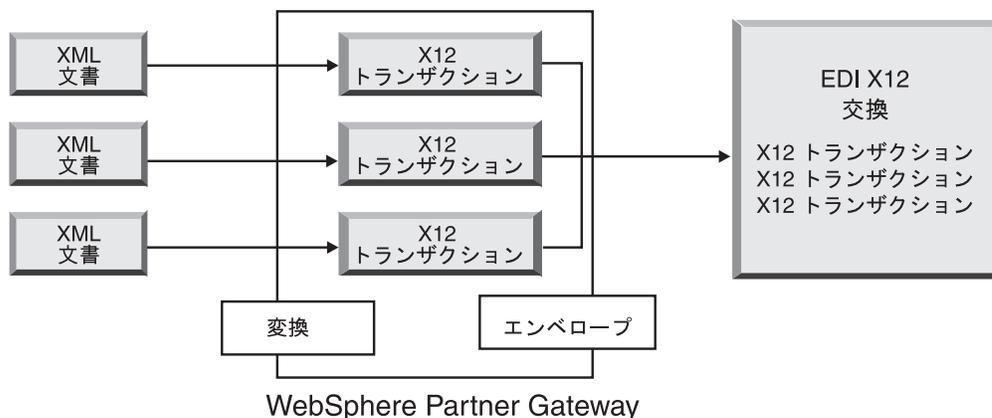


図 8. バックエンド・アプリケーションからパートナーへ (1 つの EDI 交換として) 送信される複数の XML 文書

1 つの文書を複数のトランザクションに変換することができ (マップ・チェーニングを使用してマップが作成された場合)、その各トランザクションを別の交換としてエンベロープすることができます。

図 9 は、3 つの X12 トランザクションに変換される XML 文書を示しています。このうちの 2 つのトランザクションは、一緒にエンベロープされます。残りの 1 つは、別のエンベロープに含まれます。

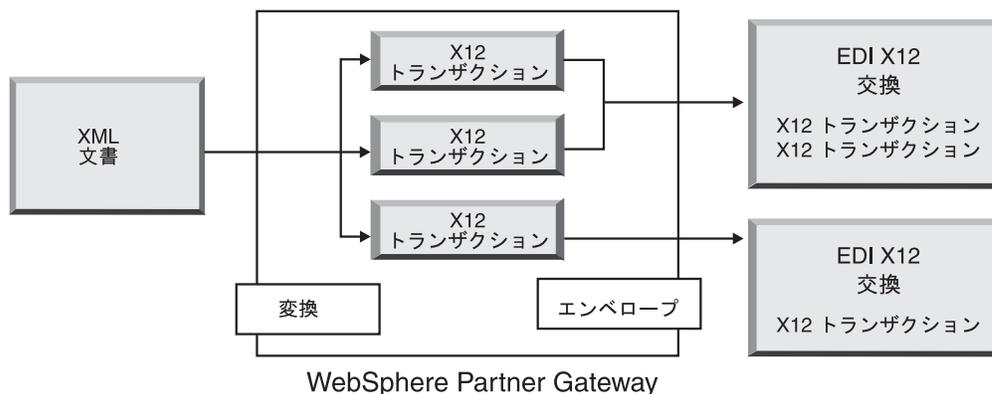


図 9. バックエンド・アプリケーションからパートナーへ (複数の EDI 交換として) 送信される 1 つの XML 文書

文書が別の XML 文書または別の ROD/フラット文書に変換された場合は、その文書の「パートナー接続」ビューでの構成に従って文書が転送されます。

- 複数の XML または ROD/フラット文書を含む単一ファイル

WebSphere Partner Gateway は文書を分割し、変換します。文書が EDI トランザクションに変換された場合、WebSphere Partner Gateway はトランザクションをエンベロープし、そのエンベロープをパートナーに送信します。バッチ ID が XML または ROD/フラット文書に割り当てられた場合、WebSphere Partner Gateway は複数の EDI トランザクションを 1 つのエンベロープに (1 つのバッチとして) 送信しようとしています。バックエンド・アプリケーションは、パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用でき、41 ページの表 17 で定義される各種のトランスポートを使用して文書を送信できます。

図 10 は、個々の XML 文書に分割される一連の XML 文書を示しています。各 XML 文書が X12 トランザクションに変換され、それぞれのトランザクションがエンベロープされます。

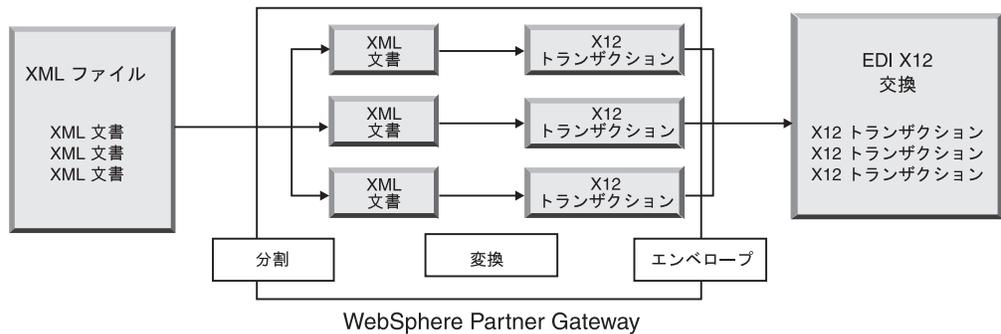


図 10. バックエンド・アプリケーションから送信され、分割され、パートナーへ (1 つの EDI 交換として) 送信される複数の XML 文書

図 10 では、文書は分割され、変換されたトランザクションと一緒にエンベロープされます。文書を分割できるようにするには、文書を送信するために使用するレシーバーのスプリッター・ハンドラー (この場合には、XML スプリッター・ハンドラー) を構成する必要があります。このシナリオを実行するためには、XML スプリッター・ハンドラーの BCG\_BATCHDOCS オプションがオン (デフォルト値) に設定されていることが必要です。BCG\_BATCHDOCS は、バッチ ID を XML 文書に割り当てて、生成されるトランザクションが同じエンベロープに格納されるようにします。XML スプリッター・ハンドラーおよび BCG\_BATCHDOCS 属性については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

文書が他の XML 文書または他の ROD/フラット文書に変換された場合は、その文書の「パートナー接続」ビューでの構成に従って文書が転送されます。

- 複数の EDI 交換を含む単一ファイル

WebSphere Partner Gateway はファイルを個々の交換に分割します。次に、交換のエンベロープを解除して個々のトランザクションに戻し、トランザクションを変換します。文書が EDI トランザクションに変換された場合、WebSphere Partner Gateway はトランザクションをエンベロープし、そのエンベロープをパートナーに送信します。バックエンド・アプリケーションは、パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用でき、41 ページの表 17 で定義される各種のトランスポートを使用して文書を送信できます。

文書が XML 文書または ROD/フラット文書に変換された場合は、その文書の「パートナー接続」ビューでの構成に従って文書が転送されます。

**パートナーからバックエンド・アプリケーションへのフロー:** パートナーは、以下のタイプの文書を送信できます。

- 1 つ以上のトランザクションを含む単一 EDI 交換

WebSphere Partner Gateway は、個々の EDI トランザクションのエンベロープを解除し、これらのトランザクションを変換します。トランザクションが EDI に変換された場合は、トランザクションがエンベロープされ、バックエンド・アプリケーションに転送されます。バックエンド・アプリケーションは、パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用でき、トランザクションは 42 ページの表 18 で定義される各種のトランスポートを使用して送信できます。

トランザクションが XML 文書または ROD/フラット文書に変換された場合は、そのトランザクションの「パートナー接続」ビューでの構成に従ってトランザクションが転送されます。

- XML または ROD/フラット文書などの単一文書

WebSphere Partner Gateway は、文書を EDI トランザクションに変換し、トランザクションをエンベロープし、そのエンベロープをバックエンド・アプリケーションに転送します。パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用できます。

文書が別の XML 文書または別の ROD/フラット文書に変換された場合は、その文書の「パートナー接続」ビューでの構成に従って文書が転送されます。

- 複数の XML または ROD/フラット文書を含む単一ファイル

WebSphere Partner Gateway は文書を分割し、変換します。文書が EDI トランザクションに変換された場合、WebSphere Partner Gateway はトランザクションをエンベロープし、そのエンベロープをバックエンド・アプリケーションに送信します。バッチ ID が XML または ROD/フラット文書に割り当てられた場合、WebSphere Partner Gateway は複数の EDI トランザクションを 1 つのエンベロープに (1 つのバッチとして) 送信しようとしています。パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用できます。

文書が他の XML 文書または他の ROD/フラット文書に変換された場合は、その文書の「パートナー接続」ビューでの構成に従って文書が転送されます。

- 複数の EDI 交換を含む単一ファイル

WebSphere Partner Gateway はファイルを個々の交換に分割します。次に、交換のエンベロープを解除して個々のトランザクションに戻し、トランザクションを変換します。文書が EDI トランザクションに変換された場合、WebSphere Partner Gateway はトランザクションをエンベロープし、そのエンベロープをバックエンド・アプリケーションに送信します。パッケージ化なしまたはバックエンド統合パッケージ化のどちらかを使用できます。

文書が XML 文書または ROD/フラット文書に変換された場合は、その文書の「パートナー接続」ビューでの構成に従って文書が転送されます。

**機能確認通知:** 機能確認通知は、EDI 交換が受信されたことを示します。機能確認通知は、送達される前に必ずエンベロープされます。

**注:** 機能確認通知は、WebSphere Partner Gateway によってエンベロープが解除された交換または WebSphere Partner Gateway によって生成された交換のみに適用されます。WebSphere Partner Gateway を通過する交換には適用されません。

WebSphere Partner Gateway が受信した交換の場合

- バックエンド・アプリケーションから交換を受信した場合、WebSphere Partner Gateway は、バックエンド・アプリケーションへの応答として機能確認通知を送信できます。
- パートナーから交換を受信した場合、WebSphere Partner Gateway は、パートナーへの応答として機能確認通知を送信できます。

WebSphere Partner Gateway が生成した交換の場合

- パートナーに交換を送信した場合、パートナーは、WebSphere Partner Gateway への応答として機能確認通知を送信できます。WebSphere Partner Gateway は、この機能確認通知をバックエンド・システムへは送信しません。
- バックエンド・アプリケーションに交換を送信した場合、バックエンド・アプリケーションは、WebSphere Partner Gateway への応答として機能確認通知を送信できます。WebSphere Partner Gateway は、この機能確認通知をパートナーへは送信しません。

## RosettaNet

WebSphere Partner Gateway は、RosettaNet 1.1 および 2.0 標準に準拠した文書を送受信するためのサポートを提供します。パートナーがハブに RosettaNet メッセージを送信するときには、パートナー接続のターゲット側でバックエンド統合を指定しておく必要があります。ハブは、メッセージのペイロードを RNSC フォーマットに変換し、メッセージをバックエンド・システムに送信します。バックエンド統合パッケージ化が使用されるため、ハブはトランスポート・レベル・ヘッダーをメッセージに追加します。メッセージは、HTTP または JMS トランスポート・プロトコルを使用して送信されます。トランスポート・レベル・ヘッダーは、PIP に属さないメタ情報を保持し、これにより WebSphere Partner Gateway が適切にメッセージの経路を定めることができます。

同様に、内部パートナー・バックエンド・システムがハブに RNSC メッセージを送信するときには、パートナー接続のソース側でバックエンド統合パッケージ化を指定しておく必要があります。バックエンド・システムがトランスポート・レベル・ヘッダーを提供する必要があります。

例えばアプリケーションが、HTTP で送信される RosettaNet を使用して、外部パートナーへメッセージを送信するとします。アプリケーションは RosettaNet サービス・コンテンツを提供し、トランスポート・レベル・ヘッダーを追加します。ヘッダーは、要求を処理する外部パートナー、送信される PIP、PIP のバージョン、およびその他の情報を示します。この情報を用いて、WebSphere Partner Gateway は正しい PIP を外部パートナーに送信することができます。

RosettaNet サポートのセットアップおよび PIP の構成について詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

**イベント通知:** WebSphere Partner Gateway は、内部パートナー・バックエンド・アプリケーションの代わりに、外部パートナーとの RNIF PIP プロセスを実行します。したがって、WebSphere Partner Gateway は、RNIF PIP プロセス実行についてのさまざまな情報をバックエンド・アプリケーションに通知するメカニズムとして、イベント通知を提供します。イベント通知機能によって、例えば WebSphere Partner Gateway がパートナーへ PIP を送信できないことを、WebSphere Partner Gateway がアプリケーションに通知できるようにします。これにより、アプリケーションはその失敗を処理できます。

イベント通知メッセージは、WebSphere Partner Gateway またはアプリケーション内部で発生したイベントに関する情報を伝える XML 文書です。これらのメッセージは、WebSphere Partner Gateway に出入りする他のメッセージと同じ構造を持っています。すなわち、トランスポート・レベル・ヘッダーとペイロードを含んでいます。WebSphere Partner Gateway は、イベント通知メッセージを送信するか、または送信しないかをオプションとして構成できます。

表 2 に WebSphere Partner Gateway がバックエンド・システムに送信できるイベント通知メッセージを示します。

表 2. バックエンド・システムに送信されるイベント通知メッセージ

イベント条件	イベント通知メッセージ
WebSphere Partner Gateway が、RosettaNet 文書を外部パートナーに送達し、受信確認を受け取る。	Event StatusCode 100
RNIF アクション実行中の一般エラー。WebSphere Partner Gateway が、0A1 メッセージを生成し、それを外部パートナーに送達して、PIP をキャンセルする。  これが PIP での最後のアクションである場合、このアクション・コードでは、WebSphere Partner Gateway が 0A1 メッセージを生成し、それを外部パートナーに送達して、この PIP をキャンセルする。それ以外の場合、WebSphere Partner Gateway は「一般例外」をパートナーに送信する。	Event StatusCode 800
RNIF アクション実行中のエラー。PIP が指定するディクショナリーと照合して、サービス・コンテンツを検証する。WebSphere Partner Gateway が、0A1 メッセージを生成し、それを外部パートナーに送達して、PIP をキャンセルする。	Event StatusCode 801
WebSphere Partner Gateway が、外部パートナーから受信確認例外または一般例外を受け取る。	Event StatusCode 900

WebSphere Partner Gateway は、除外リスト管理を使用して 0A1 メッセージを送信するように構成されている場合、他の PIP の場合と同様にこれらのメッセージを宛先アプリケーションへ送信することができます。「*WebSphere Partner Gateway 管理者ガイド*」の『除外リストの管理 (Managing Exclusion Lists)』を参照してください。

アプリケーションは、WebSphere Partner Gateway にイベント通知メッセージを送信して、RosettaNet PIP をキャンセルすることができます。

WebSphere Partner Gateway が使用するイベント・コードについて詳しくは、27 ページの『WebSphere Partner Gateway が使用するイベント・コード』を参照してください。

## ebMS

WebSphere Partner Gateway は、ebMS 2.0 標準に準拠した文書を送受信するためのサポートを提供します。パートナーがハブに ebMS メッセージを送信するときには、パートナー接続のターゲット側でバックエンド統合を指定しておく必要があります。ハブは、メッセージからペイロードを抽出し、メッセージをバックエンド・システムに送信します。バックエンド統合パッケージ化が使用されるため、ハブはトランスポート・レベル・ヘッダーをメッセージに追加します。メッセージは、HTTP または JMS トランスポート・プロトコルを使用して送信されます。トランスポート・レベル・ヘッダーは、ebMS メッセージに属さないメタ情報を保持し、これにより WebSphere Partner Gateway が適切にメッセージの経路を定めることができます。

同様に、内部パートナー・バックエンド・システムがハブにペイロードを送信するときには、パートナー接続のソース側でバックエンド統合パッケージ化を指定しておく必要があります。バックエンド・システムがトランスポート・レベル・ヘッダーを提供する必要があります。

例えばアプリケーションが、HTTP で ebMS を使用して、外部パートナーへメッセージを送信するとします。アプリケーションは、ebMS サービス、アクション、パートナー ID (受信側と送信側)、プロトコルとパッケージ化の情報、メッセージ ID、および会話 ID を提供し、トランスポート・レベル・ヘッダーを追加します。ヘッダーは、要求を処理する外部パートナーを示します。この情報を用いて、WebSphere Partner Gateway は ebMS メッセージを作成し、バックエンドからのペイロードを ebMS ペイロードとして書き込み、そのペイロードを外部パートナーに送信することができます。

表3 のヘッダーは、ebMS フローのバックエンドが設定する必要があります。

表3. ebMS フローの場合に必要なバックエンド・ヘッダー

ヘッダー名	説明	値の例	必須かどうか
x-aux-sender-id	メッセージ送信側の ID (DUNS 番号など)。	987654321	はい
x-aux-receiver-id	メッセージ受信側の ID (DUNS 番号など)。	123456789	はい
x-aux-protocol	メッセージ内容のプロトコル。この値は、「パートナー接続」ビューのソース側に向かうプロトコル値と一致しなければなりません。	ebMS	はい
x-aux-protocol-version	メッセージ内容のプロトコルのバージョン。この値は、「パートナー接続」ビューのソース側に向かうプロトコルのバージョンと一致しなければなりません。	2.0	はい

表 3. ebMS フローの場合に必要なバックエンド・ヘッダー (続き)

ヘッダー名	説明	値の例	必須かどうか
x-aux-process-type	実行されるプロセスまたは送信しようとするメッセージのタイプ。この値は、「パートナー接続」ビューのソース側に向かう文書タイプ値と一致しなければなりません。	Purchase Order Request	はい
x-aux-process-version	プロセスのバージョン。この値は、「パートナー接続」ビューのソース側の文書タイプ・バージョンと一致しなければなりません。この値は、ebXML メッセージのサービス・タイプと同じです。	ALL	はい
x-aux-activity	送信されたメッセージのタイプに対して実行するアクティビティ。この値は、「パートナー接続」ビューのソース側のアクティビティ値と一致しなければなりません。ebMS フローの場合、これは x-aux-process-type と同じです。	Purchase Order Request	はい
x-aux-action	ebMS メッセージに対して実行するアクション。この値は、ebXML Soap メッセージの Action エlementと同じです。さらに、「パートナー接続」ビューのソース側のアクション値と等しくなければなりません。	Purchase Order Action	はい
x-aux-system-msg-id	メッセージのグローバル固有 ID (GUID)。重複チェックに使用されます。この ID は、重複チェックを実行するために WebSphere Partner Gateway によって使用されます。ebMS フローでは使用されません。	28282828282828	はい
x-aux-production	メッセージのルーティング。有効値は Production および Test です。この値は両方向の要求に取り込まれます。	Production	はい
x-aux-msg-id	固有のメッセージ ID。ebXML Soap メッセージでは MessageId として設定します。	123454321@abc.com	いいえ
contentType	メッセージの content-Type。バックエンドからの wbiPackaging にラップされたメッセージを送信している場合は不要です。	text/xml	いいえ
x-aux-process-instance-id	ebXML メッセージの processId または ConversationId。	329878738@abc.com	いいえ

表 3. ebMS フローの場合に必要なバックエンド・ヘッダー (続き)

ヘッダー名	説明	値の例	必須かどうか
x-aux-seq-number	メッセージのシーケンス番号 (文書フロー定義のターゲット側で「メッセージ順序セマンティクス」が「保証済み」に設定されている場合)。これは、その会話で送信されるメッセージの順序です。	2	いいえ
x-aux-request-msg-id	このヘッダーは、応答文書の場合にのみ設定する必要があります。このヘッダーの値は、現在の応答文書に関連付けられた要求文書の messageId になります。	123455521@abc.com	いいえ
x-aux-role	WBIPackaging1.2 を使用している場合は、ユーザー自身が XML でこの属性を設定できるため、このフィールドは必須ではありません。それ以外の場合、このフィールドの値は ebXML Soap メッセージの Manifest/Refererence エレメントの role 属性の値です。	http://reqrep.org/gci/purchaseOrder	いいえ
x-aux-role-schema	役割を検証するスキーマ。WebSphere Partner Gateway はこの検証を実行しません。このフィールドの値は、ebXML Soap メッセージに Schema 属性として書き込みます。	http://reqrep.org/gci/purchaseOrder/po.xsd	いいえ
x-aux-role-schema-version	上記のスキーマのバージョン	2.0	いいえ
x-aux-description	ペイロードの説明。この値は、ebXML Soap メッセージの Description エレメントの値として書き込まれます。	Purchase Order Request	いいえ

ebMS サポートのセットアップおよび ebMS 接続の構成について詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

#### イベント通知:

WebSphere Partner Gateway は、ebMS プロセス実行についてのさまざまな情報をバックエンド・アプリケーションに通知するメカニズムとして、イベント通知を提供します。イベント通知について詳しくは、『EventNotification/XMLEvent』セクションを参照してください。

23 ページの表 4 は、WebSphere Partner Gateway と ebMS のバックエンドとの間で送受信されるイベント通知メッセージの要約を示しています。

表 4. バックエンド・システムに送信されるイベント通知メッセージ

イベント条件	イベント通知メッセージ
WebSphere Partner Gateway が、パートナーから受信確認通知を受信した後に、このイベントを生成する。	Event StatusCode 100
WebSphere Partner Gateway が、外部パートナーから受信確認例外または一般例外を受け取る。	Event StatusCode 900
WebSphere Partner Gateway が、すべての再試行が完了した場合またはメッセージ存続時間が経過してもメッセージの確認通知が受信されなかった場合に、このイベントを生成する。	Event StatusCode 902
WebSphere Partner Gateway が、送達の失敗が発生した場合に、このイベントを生成する。つまり、パートナー宛先がダウンしているかまたはトランスポートおよび宛先レベルのすべての再試行回数を超えたために、WebSphere Partner Gateway がメッセージをパートナーに送信できない。	Event StatusCode 903

WebSphere Partner Gateway が使用するイベント・コードについては、27 ページの『WebSphere Partner Gateway が使用するイベント・コード』を参照してください。

## XMLEvent

WebSphere Partner Gateway は、メッセージの状況についての情報をバックエンドに通知する方法を提供します。例えば、WebSphere Partner Gateway がパートナーにメッセージを送信し、パートナーがその応答として確認通知を送信する場合、WebSphere Partner Gateway はイベント・コード 100 (イベント・コードのリストについては 27 ページの『WebSphere Partner Gateway が使用するイベント・コード』を参照) を含む XML イベントを生成し、そのイベントをバックエンドに送信します。XML イベントの構造は、24 ページの『イベントメッセージ構造』に示されています。XMLEvent は、WebSphere Partner Gateway とバックエンド・アプリケーションの間でのみ使用されます。パートナーに送信されることはありません。WebSphere Partner Gateway V6.2 では、XMLEvent は RosettaNet と ebMS のフローでのみ生成されます。

WebSphere Partner Gateway とバックエンド・アプリケーションの間で作成される XMLEvent を使用可能または使用不可に設定することができます。詳しくは『XMLEvents の使用可能化または使用不可能化』を参照してください。

### XMLEvents の使用可能化または使用不可能化:

#### このタスクについて

XMLEvent を使用可能に設定するには、以下を行います。

1. RosettaNet または ebMS の文書定義フローに対して適切に B2B 機能を有効にします。

ソース B2B 機能は以下の要件を満たす必要があります。

パッケージ: バックエンド統合 (1.0)

プロトコル: XMLEvent (1.0)

文書タイプ: XMLEvent

ターゲット B2B 機能は以下の要件を満たす必要があります。

パッケージ: バックエンド統合 (1.0)

プロトコル: XMLEvent (1.0)

文書タイプ: XMLEvent

2. 以下の手順でソースとターゲットの間のインタラクションを作成します。
  - a. 「ハブ管理」 > 「ハブ構成」 > 「文書定義」の順にクリックします。
  - b. 「インタラクションの管理」をクリックします。
  - c. 「インタラクションの作成」をクリックします。
  - d. 「ソース」の下にある「パッケージ: バックエンド統合 (1.0)」 > 「プロトコル: XMLEvent」を展開します。
  - e. 「文書タイプ: XMLEvent (1.0)」を選択します。
  - f. 前述の手順 (手順 d と手順 e) を「ターゲット」欄で繰り返します。
  - g. 「アクション」リストから「パススルー」を選択し、「保存」をクリックします。
3. 次の手順で XMLEvent をアクティブ化します。
  - a. 「アカウント管理」 > 「接続」をクリックします。「接続の管理」ページが表示されます。
  - b. 「ソース」と「ターゲット」を選択します。
  - c. 「検索」をクリックして、指定した検索条件に一致する接続を検索します。
  - d. XMLEvent を使用可能に設定するには、「アクティブ化」をクリックします。「接続の管理」ページが再表示され、必要な接続が緑色で強調表示されます。このページにはソースとターゲットに対するパッケージ、プロトコル、および文書タイプが表示されます。さらにパートナー接続状況およびパラメーターの表示と変更ができます。

XMLEvent を使用不可に設定するには、次のいずれかを行います。

- 「ハブ管理」 > 「文書定義」 > 「文書定義の管理」で表示される「プロトコル: XMLEvent」の「無効化」を選択します。または
- 以下を行います。
  1. 「アカウント管理」 > 「除外リスト」をクリックします。
  2. 「パートナー名」を選択します。
  3. 「編集」をクリックします。
  4. 「バックエンド・イベント通知の非送信」チェック・ボックスを選択します。

**イベント通知:** RosettaNet と ebMS の処理では、XML イベントがバックエンドに送信されるかまたはバックエンドから受信される場合があります。これらのプロトコルで XML イベントを使用する場合に該当するプロトコル固有の情報については、19 ページの『イベント通知』(RosettaNet 処理の場合) および 22 ページの『イベント通知』(ebMS 処理の場合) を参照してください。

**イベントメッセージ構造:** イベント通知メッセージの標準トランスポート・レベル・ヘッダーでは、x-aux-process-type フィールドが XMLEvent に設定されています。ただし、メッセージのペイロードは、25 ページの図 11 の XML スキーマに示

すように特有の構造になっています。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace=
    "http://www.ibm.com/websphere/bcg/2003/v1.0/xmleventnotification"
  xmlns:evntf=
    "http://www.ibm.com/websphere/bcg/2003/v1.0/xmleventnotification"
  elementFormDefault="qualified">

  <!-- EventNotification version 1.0 document element -->
  <xsd:element name="EventNotification">
    <xsd:complexType>
      <xsd:all>
        <xsd:element ref="evntf:StatusCode"/>
        <xsd:element ref="evntf:StatusMessage"/>
        <xsd:element ref="evntf:EventMessageID"/>
        <xsd:element ref="evntf:BusinessObjectID"/>
        <xsd:element ref="evntf:GlobalMessageID"/>
        <xsd:element ref="evntf:Timestamp"/>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>

  <!-- StatusCode element -->
  <xsd:element name="StatusCode">
    <xsd:simpleType>
      </xsd:simpleType>
    </xsd:element>

  <!-- StatusMessage element -->
  <xsd:element name="StatusMessage">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
  </xsd:element>

  <!-- EventMessageID element -->
  <xsd:element name="EventMessageID">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
  </xsd:element>

  <!-- BusinessObjectID element -->
  <xsd:element name="BusinessObjectID">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
  </xsd:element>

  <!-- GlobalMessageID element -->
  <xsd:element name="GlobalMessageID">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
  </xsd:element>

  <!-- Timestamp element -->
  <xsd:element name="Timestamp">
    <xsd:simpleType>
      <xsd:restriction base="xsd:dateTime"/>
    </xsd:simpleType>
  </xsd:element>
</xsd:schema>
```

図 11. イベント通知メッセージの XML スキーマ

表 5 では、イベント・ペイロード内部の各フィールドについて説明しています。

表 5. イベント通知 XML フィールド

フィールド	説明
StatusCode	メッセージのタイプ。有効な値については、27 ページの『WebSphere Partner Gateway が使用するイベント・コード』を参照してください。
StatusMessage	このイベント通知メッセージの英数字記述。
EventMessageID	この特定のイベント通知メッセージの英数字 ID。

表 5. イベント通知 XML フィールド (続き)

フィールド	説明
BusinessObjectID	このメッセージ通知イベントに影響を受けるメッセージのトランスポート・レベル・ヘッダー内の x-aux-msg-id。これにより下のメッセージのペイロードがこのイベントにリンクされます。
GlobalMessageID	このメッセージ通知イベントを発生させたメッセージのトランスポート・レベル・ヘッダー内の x-aux-system-msg-id。
Timestamp	イベント発生日時。次の UTC タイム・スタンプ形式を使用します。 CCYY-MM-DDThh:mm:ssZ  これには、小数点以下の精度の秒 (...ss.ssssZ) が含まれます。日付スタンプは、dateTime の XML スキーマ・データ型 (w3.org/TR/2001/REC-xmlschema-2-20010502#dateTime) に準拠する必要があります。

**イベント通知の例:**

27 ページの図 12 に、HTTP プロトコルを使用して送信されたイベント通知メッセージの例を示します。

```

POST /builderURL HTTP/1.1
Content-Type: application/xml
Content-length: 250
x-aux-sender-id: 000000001
x-aux-receiver-id: 000000002
x-aux-third-party-bus-id: 000000003
x-aux-create-datetime: 2002-10-28T23:05:02Z
x-aux-protocol: XMLEvent
x-aux-protocol-version: 1.0
x-aux-process-type: XMLEvent
x-aux-process-version: 1.0
x-aux-payload-root-tag: evtmf:EventNotification
x-aux-msg-id: 98732
x-aux-system-msg-id: 12345
x-aux-production: Production
x-aux-process-instance-id: 3456
x-aux-event-status-code: 100
x-aux-transport-retry-count: 0

<?xml version="1.0" encoding="UTF-8"?>
<evntmf:EventNotification xmlns:evntmf=
"http://www.ibm.com/websphere/bcg/2003/v1.0/xmleventnotification">
  <evntmf:StatusCode>100</evntmf:StatusCode>
  <evntmf:StatusMessage>The message was delivered</evntmf:StatusMessage>
  <evntmf:EventMessageID>12345</evntmf:EventMessageID>
  <evntmf:BusinessObjectID>34234</evntmf:BusinessObjectID>
  <evntmf:GlobalMessageID>98732</evntmf:GlobalMessageID>
  <evntmf:Timestamp>2001-01-31T13:20:00Z</evntmf:Timestamp>
</evntmf:EventNotification>

```

図 12. HTTP を使用したイベント通知メッセージの例

### WebSphere Partner Gateway が使用するイベント・コード:

このセクションでは、WebSphere Partner Gateway が使用するすべてのイベント・コードの要約を示します。特定のプロトコルでのこれらのイベント・コードの使用方法については、19 ページの『イベント通知』(RosettaNet プロトコルの場合) および 22 ページの『イベント通知』(ebMS プロトコルの場合) を参照してください。

表 6 は、WebSphere Partner Gateway が使用するイベント・コードの要約を示しています。

表 6. WebSphere Partner Gateway が使用するイベント・コード

イベント 状況コード	使用される プロトコル	送信元	送信先
100	RosettaNet ebMS	WebSphere Partner Gateway	バックエンド
800	RosettaNet	バックエンド	WebSphere Partner Gateway
801	RosettaNet	バックエンド	WebSphere Partner Gateway
900	RosettaNet ebMS	WebSphere Partner Gateway	バックエンド
902	ebMS	WebSphere Partner Gateway	バックエンド
903	ebMS	WebSphere Partner Gateway	バックエンド

## 使用するパッケージ化

パッケージ化のタイプによって、WebSphere Partner Gateway がバックエンド・システムにメッセージを送信する際のフォーマットおよびバックエンド・システムが WebSphere Partner Gateway にメッセージを送信する際のフォーマットが決まります。

コミュニティー・コンソールを使用して、外部パートナーとの接続を確立し、WebSphere Partner Gateway およびバックエンド・システム間で使用するパッケージを指定します。使用するパッケージを決定するには、以下の問題を理解する必要があります。

- バックエンド・システムで使用するにはどのパッケージ化のタイプが有効であるか
- 特定のビジネス・プロトコルのメッセージではどのパッケージ化のタイプが有効であるか

パートナー接続のセットアップ方法については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

統合のために WebSphere Partner Gateway を使用する場合、すべてのパッケージ化のタイプが有効というわけではありません。表 7 に、WebSphere Partner Gateway が内部パートナーのバックエンド・アプリケーションとの間で、文書またはメッセージを交換している場合に関連するパッケージ化のタイプをリストします。

表 7. バックエンド統合に関連するパッケージ化のタイプ

パッケージ化のタイプ	説明
パッケージ化なし	ヘッダー・データなしで バックエンド・システムまたはハブにメッセージを送信します。
バックエンド統合パッケージ化	メッセージ・ヘッダーに属性を追加します。オプションで、メッセージの内容を XML トランスポート・エンベロープ内にラップします。

**注:** WebSphere Partner Gateway では 他のパッケージ化のタイプ (AS など) も使用できます。ただし、バックエンド・システムとの統合には、パッケージ化なしのタイプか、バックエンド統合パッケージ化のタイプを使用することをお勧めします。

### パッケージ化なし

パッケージ化が「なし」と設定されている場合、WebSphere Partner Gateway は、バックエンド・システムにメッセージを送信するときにトランスポート・レベル・ヘッダーを追加しません。また、バックエンド・システムからメッセージを受信したときにトランスポート・レベル・ヘッダーを考慮した処理は行いません。WebSphere Partner Gateway は、メッセージのみをバックエンド・システムに送信します。ルーティングは、文書内の情報によって制御されます。

## バックエンド統合パッケージ化

パッケージ化がバックエンド統合に設定されている場合、バックエンド・システムに送受信されるメッセージには、以下のコンポーネントが含まれます。

- トランスポート・レベル・ヘッダー。メッセージに関するメタ情報が含まれています。(必須)
- ペイロード。メッセージの内容が含まれています。(必須)
- 添付ファイル (オプション)

ヘッダーおよびペイロードは必須ですが、添付ファイルはオプションです。以降のセクションでは、バックエンド統合パッケージ化を使用する文書の各コンポーネントについて説明します。

### トランスポート・レベル・ヘッダーの内容:

トランスポート・レベル・ヘッダーには、WebSphere Partner Gateway がメッセージを処理して正しい宛先にルーティングするために使用する情報が含まれています。トランスポート・レベル・ヘッダーは、両方向に対応しており、WebSphere Partner Gateway に入出力するすべてのメッセージには、必須フィールドと、適合するオプション・フィールドがあります。

表 8 にトランスポート・レベル・ヘッダーのフィールドをリストします。

表 8. トランスポート・レベル・ヘッダーのフィールド

ヘッダー・フィールド	説明	必須かどうか
x-aux-sender-id	メッセージ送信側の ID (DUNS 番号など)。	はい
x-aux-receiver-id	メッセージ受信側の ID (DUNS 番号など)。	はい
x-aux-protocol	メッセージ内容のプロトコル。有効な値は RNSC (RosettaNet サービス・コンテンツ)、XMLEvent、および Binary です。WebSphere Partner Gateway では、このフィールドの値がペイロード内のどのプロトコル・フィールドの値よりも優先されます。	はい
x-aux-protocol-version	メッセージ内容のプロトコルのバージョン。	はい
x-aux-process-type	実行されるプロセスまたは送信しようとするメッセージのタイプ。RosettaNet メッセージの場合、3A4 などの PIP コードになります。イベント・メッセージの場合は XMLEvent、2 項メッセージの場合は Binary です。WebSphere Partner Gateway では、このフィールドの値がペイロード内のどのプロセス・フィールドの値よりも優先されます。	はい
x-aux-process-version	プロセスのバージョン。RosettaNet メッセージの場合、PIP のバージョン番号になります。	はい
x-aux-create-datetime	メッセージが正常に通知された日時。UTC タイム・スタンプ形式 (CCYY-MM-DDThh:mm:ssZ) を使用します。	
x-aux-msg-id	ペイロード内容の ID。例えば、RosettaNet メッセージ用の RNPIPSERVICECONTENT インスタンスの ID や、所有文書 ID を指定することができます。これにより、トレース目的で、メッセージのペイロードがメッセージ送信側のシステム内の何かにリンクされます。	

表 8. トランスポート・レベル・ヘッダーのフィールド (続き)

ヘッダー・フィールド	説明	必須かどうか
x-aux-production	メッセージのルーティング。有効値は Production および Test です。この値は両方向の要求に取り込まれます。メッセージが、外部パートナーによって開始された両方向 PIP への応答である場合、WebSphere Partner Gateway は、要求内の GlobalUsageCode を使用して、トランスポート・レベル・ヘッダー内の値を無視します。	
x-aux-system-msg-id	メッセージのグローバル固有 ID (GUID)。重複チェックに使用されま	はい
x-aux-payload-root-tag	ペイロードのルート・タグ・エレメント。例えば、3A4 RosettaNet サービス・コンテンツの場合、このフィールドの値は Pip3A4PurchaseOrderRequest になります。イベント通知メッセージの場合は、このフィールドの値は EventNotification になります。	
x-aux-process-instance-id	複数のメッセージ・ビジネス・プロセス内の文書を固有プロセス・インスタンスにリンクする ID。RosettaNet の場合、この ID は最後の 30 日間 RosettaNet プロセスに固有でなければなりません。RosettaNet プロセス・インスタンスの一部として交換されるすべてのメッセージは、再試行の場合も含め、同じプロセス・インスタンス ID を使用します。	
x-aux-event-status-code	イベント通知の状況コード。24 ページの『イベントメッセージ構造』の StatusCode フィールドを参照してください。	
x-aux-third-party-bus-id	メッセージを送達した団体の DUNS 番号などの ID。これは、サード・パーティーがコミュニティー所有者に代わって WebSphere Partner Gateway のホスティングを行っている場合には、x-aux-sender-id および x-aux-receiver-id のどちらとも異なる場合があります。	
x-aux-transport-retry-count	今回の試行の前にこのメッセージの送付を試みて失敗した回数。最初の試行でメッセージが正しく送付された場合は、このフィールドの値は 0 になります。	
x-aux-in-file-name	JMS または HTTP を経由して WebSphere Partner Gateway に送信されるメッセージの元のファイル名。(注 3 (31 ページ) を参照。)	いいえ
x-out-filename	バックエンド統合パッケージ化で JMS を使用して送信されるメッセージの元のファイル名。(注 2 (31 ページ) を参照。)	いいえ
content-type	メッセージのコンテンツ・タイプ	
content-length	メッセージの長さ (バイト単位)。	

表 8. トランスポート・レベル・ヘッダーのフィールド (続き)

ヘッダー・フィールド	説明	必須かどうか
x-aux-SyncResponse	<p>この属性は、この要求に対して SyncResponse が必要であることをバックエンドに通知するために ebMS フローで使用されます。バックエンドからの同期応答が不要である場合は、存在しません。以下の値を取ります。</p> <p>注: この属性の値はすべて小文字です。</p> <ul style="list-style-type: none"> <li>• <b>signalonly</b>: シグナルは、受け入れシグナルまたは例外シグナルです。どのシグナルになるかは、バックエンドによって決まります。属性値が signalonly である場合、送信側のパートナーは、同期接続でシグナルのみを受け入れます。</li> <li>• <b>responseonly</b>: 応答文書は、受信された文書のビジネス応答です。値が responseonly である場合、送信側のパートナーは、同期接続で応答を受け入れます。</li> <li>• <b>signalsandresponse</b>: この場合は、同期接続でシグナルと応答の両方を送信する必要があります。この属性はルーティング・オブジェクト属性です。これは CPA 内に存在します。CPA が存在しない場合には、パートナー接続レベルで設定できます。</li> </ul>	いいえ
x-aux-TimeToAccept	<p>この属性は ebMS フローで使用され、CPA 内の「timeToAcknowledgeAcceptance」の値を格納します。CPA がアップロードされると、この値は、ソース文書定義属性として TimeToPerform に設定されます。これは分を表す数値です。この値は、ビジネス文書がビジネス・ルール検証で受け入れられた後で、受信した側がビジネス文書の受け入れを非実証的に確認通知する必要がある時間の上限を指定します。</p>	いいえ
x-aux-IntelligibleCheckRequired	<p>この属性は ebMS フローで使用されます。可能な値は <b>yes</b> と <b>no</b> です。値が「yes」の場合、バックエンドはビジネス文書が文字化けしていないこと (つまりスキーマ妥当性検査に合格していること) を必ず検証してから、受信通知シグナルを戻します。</p>	

**注記:**

1. IBM WebSphere MQ (JMS プロバイダー) との互換性を持たせるため、JMS プロトコル・メッセージのフィールドでは、ハイフンではなく下線を使用します。例えば、JMS メッセージ内では x-aux-sender-id フィールドではなく x\_aux\_sender\_id フィールドになります。
2. ゲートウェイが「HTTP」として指定され、パッケージが「なし」として指定された場合は、HTTP ヘッダーで元のファイル名が「Content-Disposition: attachment;po.xml」として設定されます。  
  
レシーバーが「JMS」として指定され、パッケージが「バックエンド統合」として指定された場合は、元のファイル名が x-out-filename および他の x-aux-\* ヘッダーに書き込まれます。
3. レシーバーが「HTTP」として指定され、ソース・パッケージが「なし」として指定された場合は、HTTP ヘッダーで元のファイル名が「Content-Disposition: attachment;po.xml」として設定されます。

レシーバーが「JMS」として指定され、ソース・パッケージが「バックエンド統合」として指定された場合は、元のファイル名が「x\_aux\_in\_file\_name: po.xml」に設定されます。HTTP レシーバーの場合は「Content-Disposition: attachment;po.xml」に設定されます。

29 ページの表 8 にトランスポート・レベル・ヘッダー情報の概要を示します。以下のセクションで、特定のビジネス・プロトコルに固有のトランスポート・レベル・ヘッダー情報を説明します。

- 『トランスポート・レベル・ヘッダーおよび RosettaNet メッセージ』
- 33 ページの『トランスポート・レベル・ヘッダーと AS2 メッセージ』
- 34 ページの『トランスポート・レベル・ヘッダーと AS1 メッセージ』

### トランスポート・レベル・ヘッダーおよび RosettaNet メッセージ:

表 9 では、WebSphere Partner Gateway が、トランスポート・レベル・ヘッダーの各フィールド用の値を RosettaNet メッセージのどの部分から取得するかを説明します。

表 9. トランスポート・レベル・ヘッダーのフィールドと RosettaNet の内容

ヘッダー・フィールド	値のソース: RosettaNet 2.0	値のソース: RosettaNet 1.1
x-aux-sender-id	<(DeliveryHeader)> <messageSenderIdentification> <PartnerIdentification> <GlobalBusinessIdentifier>	<ServiceHeader> <ProcessControl> <TransactionControl> <ActionControl> or <SignalControl> <PartnerRouter> <fromPartner> <PartnerDescription> <BusinessDescription> <GlobalBusinessIdentifier>
x-aux-receiver-id	<(DeliveryHeader)> <messageReceiverIdentification> <PartnerIdentification> <GlobalBusinessIdentifier>	<ServiceHeader> <ProcessControl> <TransactionControl> <ActionControl> or <SignalControl> <PartnerRouter> <toPartner> <PartnerDescription> <BusinessDescription> <GlobalBusinessIdentifier>
x-aux-protocol	RosettaNet の値 RNSC を設定	RosettaNet 2.0 の場合と同じ
x-aux-protocol-version	値 1.0 を設定	RosettaNet 2.0 の場合と同じ
x-aux-process-type	ソース XPath は次のとおり。 /ServiceHeader/ProcessControl/ pipCode/GlobalProcessIndicatorCode	ソース XPath は次のとおり。 /ServiceHeader/ProcessControl/ ProcessIdentity/GlobalProcessIndicatorCode
x-aux-process-version	ソース XPath は次のとおり。 /ServiceHeader/ProcessControl/ pipVersion/VersionIdentifier  各 PIP のバージョン ID の値は、PIP 仕様 に示されています。	ソース XPath は次のとおり。 /ServiceHeader/ProcessControl/ ProcessIdentity/VersionIdentifier  各 PIP のバージョン ID の値は、PIP 仕様 に示されています。
x-aux-payload- root-tag	Pip3A4PurchaseOrderRequest などの PIP 名	RosettaNet 2.0 の場合と同じ

表9. トランスポート・レベル・ヘッダーのフィールドと RosettaNet の内容 (続き)

ヘッダー・フィールド	値のソース: RosettaNet 2.0	値のソース: RosettaNet 1.1
x-aux-process-instance-id	アプリケーションによって開始されたプロセスの場合、値はプロセス・インスタンスの ID です。外部パートナーによって開始された、パススルー・ワークフローではないプロセスの場合、値は最初の RosettaNet 要求内のプロセス ID です。  <ServiceHeader> <ProcessControl> <InstanceIdentifier>	<ServiceHeader> <ProcessControl> <ProcessIdentity> <InstanceIdentifier>
x-aux-msg-id	<(RNPipServiceContent)> <thisDocumentIdentifier> <ProprietaryDocumentIdentifier>	RosettaNet 2.0 の場合と同じ
x-aux-production	<ServiceHeader> <ProcessIndicator> <GlobalUsageCode>	<Preamble> <GlobalUsageCode>

### トランスポート・レベル・ヘッダーと AS2 メッセージ:

表 10 では、WebSphere Partner Gateway が、トランスポート・レベル・ヘッダーの各フィールド用の値を AS2 メッセージのどの部分から取得するかを説明します。

注: これらの値では大文字小文字が区別されます。

表 10. AS2 の内容からのトランスポート・レベル・ヘッダーのフィールド

ヘッダー・フィールド	外部パートナーからハブに AS/2 メッセージが送信されるとき値のソース	外部パートナーに AS2 メッセージが送信されるとき値のソース
x-aux-sender-id	AS2 メッセージの AS2-From ヘッダー・フィールドが、内部パートナーに送信されるバックエンド統合メッセージの x-aux-sender-id フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-sender-id フィールドが AS2 メッセージの AS2-From ヘッダー値として使用されます。
x-aux-receiver-id	AS2 メッセージの AS2-To ヘッダー・フィールドが、内部パートナーに送信されるバックエンド統合メッセージの x-aux-receiver-id フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-receiver-id フィールドが AS2 メッセージの AS2-To ヘッダー値として使用されます。
x-aux-protocol	パートナー接続の ToProtocol が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-protocol フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-protocol フィールドがパートナー接続の FromProtocol を決定するために使用されます。
x-aux-protocol-version	パートナー接続の ToProtocolVersion が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-protocol-version フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-protocol-version フィールドがパートナー接続の FromProtocolVersion として使用されます。

表 10. AS2 の内容からのトランスポート・レベル・ヘッダーのフィールド (続き)

ヘッダー・フィールド	外部パートナーからハブに AS/2 メッセージが送信される時の値のソース	外部パートナーに AS2 メッセージが送信される時の値のソース
x-aux-process-type	パートナー接続の ToProcessCode が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-process-type フィールドを設定するために使用されます。	着信バックエンド統合メッセージの x-aux-process-type フィールドがパートナー接続の FromProcessCode として使用されます。
x-aux-process-version	パートナー接続の ToProcessVersion が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-process-version フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-process-version フィールドがパートナー接続の FromProcessVersion として使用されます。
x-aux-payload-root-tag	カスタム XML プロトコルの場合のみ、XPath で指定されたルート・タグがメッセージから構文解析によって取り出され、x-aux-payload-root-tag フィールド内で使用されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。
x-aux-process-instance-id	このフィールドは AS2 に使用されます。	このフィールドは AS2 に使用されます。
x-aux-msg-id	カスタム XML プロトコルの場合のみ、XPath で指定された文書 ID がメッセージから構文解析によって取り出され、x-aux-msg-id フィールド内で使用されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。
x-aux-system-msg-id	このフィールドは、このメッセージ用に内部で生成された固有 ID に設定されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。
x-aux-production	このフィールドは AS2 には使用されません。	このフィールドは AS2 には使用されません。

### トランスポート・レベル・ヘッダーと ASI メッセージ:

表 11 では、WebSphere Partner Gateway が、トランスポート・レベル・ヘッダーの各フィールド用の値を AS1 メッセージのどの部分から取得するかを説明します。

注: これらの値では大文字小文字が区別されます。

表 11. ASI の内容から取得したトランスポート・レベル・ヘッダーのフィールド

ヘッダー・フィールド	外部パートナーからハブに AS/1 メッセージが送信される時の値のソース	外部パートナーに AS/1 メッセージが送信される時の値のソース
x-aux-sender-id	AS1 メッセージの「Subject: ToID;FromID」ヘッダー・フィールド内の FromID が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-sender-id フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-sender-id フィールドが AS1 メッセージの "Subject: ToID;FromID" ヘッダー値内の FromID として使用されます。
x-aux-receiver-id	AS1 メッセージの「Subject: ToID;FromID」ヘッダー・フィールド内の ToID が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-receiver-id フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-receiver-id フィールドが AS1 メッセージの "Subject: ToID;FromID" ヘッダー値内の ToID として使用されます。

表 11. ASI の内容から取得したトランスポート・レベル・ヘッダーのフィールド (続き)

ヘッダー・フィールド	外部パートナーからハブに ASI メッセージが送信される時の値のソース	外部パートナーに ASI メッセージが送信される時の値のソース
x-aux-protocol	パートナー接続の ToProtocol が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-protocol フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-protocol フィールドがパートナー接続の FromProtocol として使用されます。
x-aux-protocol-version	パートナー接続の ToProtocolVersion が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-protocol-version フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-protocol-version フィールドがパートナー接続の FromProtocolVersion として使用されます。
x-aux-process-type	パートナー接続の ToProcessCode が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-process-type フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-process-type フィールドがパートナー接続の FromProcessCode として使用されます。
x-aux-process-version	パートナー接続の ToProcessVersion が、内部パートナーに送信されるバックエンド統合メッセージの x-aux-process-version フィールド内に設定されます。	着信バックエンド統合メッセージの x-aux-process-version フィールドがパートナー接続の FromProcessVersion として使用されます。
x-aux-payload-root-tag	カスタム XML プロトコルの場合のみ、XPath で指定されたルート・タグがメッセージから構文解析によって取り出され、x-aux-payload-root-tag フィールド内に設定されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。
x-aux-process-instance-id	このフィールドは ASI には使用されません。	このフィールドは ASI には使用されません。
x-aux-msg-id	カスタム XML プロトコルの場合のみ、XPath で指定された文書 ID がメッセージから構文解析によって取り出され、x-aux-msg-id フィールド内で使用されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。
x-aux-system-msg-id	このフィールドは、このメッセージ用に内部で生成された固有 ID に設定されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。
x-aux-production	このフィールドは ASI には使用されません。	このフィールドは ASI には使用されません。

### トランスポート・レベル・ヘッダーと ebMS メッセージ:

34 ページの表 11 では、WebSphere Partner Gateway が、トランスポート・レベル・ヘッダーの各フィールド用の値を ebMS メッセージのどの部分から取得するかを説明します。

注: これらの値では大文字小文字が区別されます。

表 12. ebMS の内容からのトランスポート・レベル・ヘッダーのフィールド

ヘッダー・フィールド	外部パートナーからハブに ebMS 文書が送信されるとき の値のソース	外部パートナーに ebMS 文書が 送信されるとき の値のソース
x-aux-receiver-id	(SOAP ヘッダー) <eb:MessageHeader><eb:To><eb:PartyId>	この属性の値は、バックエンドからバックエンド・ヘッダーとして取得されます。
x-aux-sender-id	(SOAP ヘッダー) <eb:MessageHeader><eb:From><eb:PartyId>	この属性の値は、バックエンドからバックエンド・ヘッダーとして取得されます。
x-aux-SyncResponse	ソース文書定義属性 bcg.ro.ebxml.SYNCREPLYMODE から取得されます。  可能な値: <ul style="list-style-type: none"> <li>• responseOnly</li> <li>• signalAndResponse</li> <li>• signalOnly</li> </ul>	ターゲット文書定義属性から取得されます。
x-aux-process-instance-id	(SOAP ヘッダー) <eb:MessageHeader><eb:ConversationId>	バックエンドからバックエンド・ヘッダーとして取得されます。バックエンドで値が設定されていない場合は、WebSphere Partner Gateway が生成します。  ebMS フローの必須の ConversationId エlementは、2つの関係者の間の会話を構成する一連の関連メッセージを識別する文字列です。同じ会話のメッセージは ConversationId が同じになり、messageId で区別されます。
x-aux-process-type	(SOAP ヘッダー) <eb:MessageHeader><eb:Service>	バックエンドからバックエンド・ヘッダーとして取得されます。
x-aux-process-version	(SOAP ヘッダー) <eb:MessageHeader><eb:Service> "\$" 文字の後の Element 値のテキスト。Element 値に「\$」文字がない場合または Element 値が URI でない場合は、デフォルトの「ALL」が使用されます。	バックエンドからバックエンド・ヘッダーとして取得されます。
x-aux-protocol	"EBMS"	「パートナー接続」ビューのターゲット・プロトコルから取得されます。
x-aux-protocol-version	(SOAP ヘッダー) <eb:MessageHeader><eb:version>	「パートナー接続」ビューのターゲット・プロトコル・バージョンから取得されます。
x-aux-third-party-bus-id	「コンソール」>「システム管理」>「DocMgr プロパティ (DocMgr properties)」	N/A
x-aux-TimeToAccept	ソース文書定義属性から取得されます。	
x-aux-system-msg-id	このフィールドは、このメッセージ用に内部で生成された固有 ID に設定されます。	このフィールドを着信バックエンド統合メッセージ内に設定する必要はありません。

表 12. ebMS の内容からのトランスポート・レベル・ヘッダーのフィールド (続き)

ヘッダー・フィールド	外部パートナーからハブに ebMS 文書が送信されるとき の値のソース	外部パートナーに ebMS 文書が 送信されるとき の値のソース
x-aux-msg-id	<soapenv:Header><eb:MessageHeader> <eb:MessageData><eb:MessageId>	バックエンドでバックエンド・ヘッダーとして設定されることがあります。バックエンドで値が設定されていない場合は、WebSphere Partner Gateway が内部で生成します。
x-aux-IntelligibleCheckRequired	ソース文書定義属性 bcg.ro.ebxml.ISINTELLIGIBLECHECKREQUIRED から 取得されます。	N/A

ebMS トランスポート・ヘッダーの例:

```

Connection : Keep-Alive
x-aux-process-version : ALL
x-aux-receiver-id : 124456789
x-aux-SyncResponse : signalsonly
Content-Length : 1326186
x-aux-process-instance-id : 918423619611581617869031
x-aux-process-type : ALMService
x-aux-protocol : ebMS
x-aux-sender-id : 987654421
x-aux-third-party-bus-id :
Content-Type : application/xml
x-aux-create-datetime : 2006-09-13T16:15:01Z
x-aux-TimeToAccept : 1
x-aux-transport-retry-count : 3
Host : 9.184.251.32:58080
x-aux-system-msg-id : 1158164092823000D606BBA390013320000000000000021
ReferenceId : 1158164101686000D606BBA390013320000000000000003
x-aux-msg-id : 1158163556558000C7627E08C025182D1C3AD7C5B55F7A3
                @wks184446wss.in.ibm.com
Keep-Alive : timeout=5, max=10000
x-aux-IntelligibleCheckRequired : no
x-aux-protocol-version : 2.0
    
```

ペイロード:

メッセージのペイロードには、メッセージの実際の内容が含まれています。ペイロードのロケーションは、表 13 に示すようにメッセージを送信するトランスポート・プロトコルに依存します。

表 13. ペイロードのロケーション

トランスポート・プロトコル	ペイロードのロケーション
HTTP プロトコル・メッセージ	HTTP ポストの本文内
JMS プロトコル・メッセージ	JMS メッセージの本文内
RosettaNet メッセージ	PIP からのサービス・コンテンツ
EDI	EDI エンベロープ
ROD/フラットまたは XML 文書	ROD/フラットまたは XML 文書

次のいずれの場合も、ペイロードは Base64 でエンコードされて、XML トランスポート・エンベロープ内に格納されます。

- 文書に添付ファイルが含まれる場合

添付ファイルを持つ文書は、XML トランスポート・エンベロープ内にラップする必要があります。添付ファイルのフォーマット設定については、『添付ファイル』を参照してください。

- バックエンド統合パッケージ化のエンベロープ・フラグを「はい」に設定した場合

文書を XML トランスポート・エンベロープ内にラップするには、添付ファイルの有無にかかわらず、プロファイルの「B2B 機能」ビューでバックエンド統合エンベロープ・フラグを「はい」に設定します。例えば、内部パートナーのプロファイルでこのフラグを設定するには、以下の作業を行います。

1. 「アカウント管理」>「プロファイル」をクリックします。
2. 内部パートナーの名前を入力します (または、すべての外部パートナーに対する検索を実行します)。
3. 内部パートナーの名前の横にある「詳細の表示」アイコンをクリックします。
4. 「B2B 機能」をクリックします。
5. 「バックエンド統合 (Backend Integration)」の横にある「編集」アイコンをクリックします。
6. 「エンベロープ・フラグ (Envelop Flag)」を「はい」に設定します。

この XML トランスポート・エンベロープは、<transport-envelope> ルート・タグ内に文書をラップします。このルート・タグの内部には、文書ペイロードを格納する <payload> タグがあります。添付ファイルが存在する場合、各添付ファイルは <attachment> タグ内に格納されます。これらのタグの構造については、『添付ファイル』を参照してください。

WebSphere Partner Gateway には、バックエンド統合 XML トランスポート・エンベロープ構造を記述した、次の W3C XML スキーマ・ファイルが含まれています。

wbipackaging\_v1.1\_ns.xsd

注: ebMS フローの場合は、パッケージ化 wbipackaging\_v1.2\_ns.xsd を使用する必要があります。

このスキーマ・ファイルは、インストール・メディア上の次のディレクトリーにあります。

B2BIntegrate¥packagingSchemas

任意の XML 編集ツールを使用して、このスキーマ・ファイルに対するバックエンド統合 XML の妥当性を検査し、文書マネージャーに送信する前に文書が有効であることを確認することができます。

#### 添付ファイル:

ビジネス・メッセージ・プロトコルで許可されていれば、各文書に 1 つ以上の添付ファイルを含めることができます。文書に添付ファイルがある場合、その文書は、37 ページの『ペイロード』で説明したように XML トランスポート・エンベロープにラップする必要があります。39 ページの表 14 で、ペイロードおよび添付ファイ

ルのタグ内の XML 属性について説明します。

表 14. ペイロードと添付ファイルのタグの XML 属性

XML 属性	説明	必須かどうか
コンテンツ・タイプ	MIME タイプ/サブタイプを示します (text/xml や image/gif など)。	はい
エンコード	エンコード方式を示します。添付ファイルとペイロードは Base64 でエンコードする必要があるため、この属性で有効な値は "Base64" のみです。	いいえ

図 13 に、ペイロードと 1 つの添付ファイルが含まれた XML トランスポート・エンベロープ内の文書の例を示します。

注: この例のネーム・スペースは必要です。

```
xmlns="http://www.ibm.com/websphere/bcg/2003/v1.0/wbipackaging"
```

```
<?xml version="1.0" encoding="utf-8"?>
<transport-envelope
  xmlns="http://www.ibm.com/websphere/bcg/2003/v1.0/wbipackaging">
  <payload encoding="base64" contentType="application/xml">
    ...base64 encoded XML message...
  </payload>
  <attachment encoding="base64" Content-Type="text/xml">
    ...base64 encoded XML attachment...
  </attachment>
</transport-envelope>
```

図 13. ペイロードと 1 つの添付ファイルが含まれた XML トランスポート・エンベロープの例

注記:

1. WebSphere Interchange Server を使用して XML トランスポート・エンベロープにラップされた文書进行处理するため、WebSphere Partner Gateway には Attachment Data Handler があります。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。
2. WebSphere Process Server 上で添付ファイル付きの文書进行处理するために、WebSphere Partner Gateway は、バックエンド統合パッケージ化データ・バインディングを提供します。68 ページの『バックエンド統合パッケージ化メッセージの処理』を参照してください。

## 文書に適合するパッケージ化のタイプ

特定のビジネス・プロトコルの文書は、特定のタイプのパッケージ化のみを使用できます。例えば RosettaNet 文書または ebMS 文書は、バックエンド統合のパッケージ化が指定された場合のみ 処理することができます。文書タイプとそれに関連す

るパッケージ化タイプの完全なリストについては、41 ページの表 16、41 ページの表 17、および 42 ページの表 18 を参照してください。

## HTTP でのバックエンド統合パッケージ化の例

図 14 に、HTTP トランスポート・プロトコルを使用した、WebSphere Partner Gateway からアプリケーションへのメッセージの例を示します。メッセージに添付ファイルは含まれていないので注意してください。

```
POST /sample/receive HTTP/1.1
Host: sample.COM
Content-Type: application/xml
Content-Length: nnn
x-aux-sender-id: 000000001
x-aux-receiver-id: 000000002
x-aux-third-party-bus-id: 000000003
x-aux-create-datetime: 2002-10-28T23:05:02Z
x-aux-protocol: RNSC
x-aux-protocol-version: 1.0
x-aux-process-type: 3A4
x-aux-process-version: V02.00
x-aux-payload-root-tag: Pip3A4PurchaseOrderRequest
x-aux-msg-id: 1021358129419
x-aux-system-msg-id: 2
x-aux-production: Production
x-aux-process-instance-id: 123456
x-aux-transport-retry-count: 0

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Pip3A4PurchaseOrderRequest SYSTEM
    "3A4PurchaseOrderRequestMessageGuideline_v1_2.dtd">
<Pip3A4PurchaseOrderRequest>
  <PurchaseOrder>
    ...
  </PurchaseOrder>
  ...
  <thisDocumentIdentifier>
    <ProprietaryDocumentIdentifier>1021358129419
    </ProprietaryDocumentIdentifier>
  </thisDocumentIdentifier>
  <GlobalDocumentFunctionCode>Request</GlobalDocumentFunctionCode>
</Pip3A4PurchaseOrderRequest>
```

図 14. HTTP トランスポート・プロトコルを使用したメッセージの例

## 使用するメッセージ・トランスポート

バックエンド・システムと WebSphere Partner Gateway が相互にメッセージを送信する場合、それぞれが同じメッセージ・トランスポート・プロトコルを使用する必要があります。メッセージ・トランスポート・プロトコルは、メッセージ送信に使用される通信プロトコルを定義します。

WebSphere Partner Gateway は、バックエンド統合インターフェースを通じてバックエンド・システムと通信します。41 ページの表 15 に、このバックエンド統合インターフェースでサポートされるトランスポート・プロトコルを示します。

表 15. バックエンド統合でサポートされるトランスポート・プロトコル

トランスポート・プロトコル	詳細の参照先
HTTP または HTTPS	43 ページの『HTTP トランスポート・プロトコル』
ファイル・システム・ファイル	48 ページの『ファイル・システム・プロトコル』
JMS	45 ページの『JMS プロトコル』

表 16 は、ハブからバックエンド・システムに文書を送信するときに各パッケージ化タイプおよびビジネス・プロトコルでサポートされるトランスポート・プロトコルを示しています。

表 16. *WebSphere Partner Gateway* からバックエンド・システムへの、サポートされるトランスポート・プロトコル

パッケージ化のタイプ	ビジネス・プロトコル	HTTP または HTTPS	JMS	ファイル・システム
バックエンド統合	RosettaNet (RNSC)	はい	はい	いいえ
	ebMS	はい	はい	いいえ
	2 項	はい	はい	いいえ
	EDI (EDI について詳しくは、42 ページの表 18 を参照)			
	XML	はい	はい	いいえ
	ROD/フラット	はい	はい	いいえ
なし	EDI (EDI について詳しくは、42 ページの表 18 を参照)			
	cXML のみ	はい	いいえ	いいえ
	SOAP のみ	はい	いいえ	いいえ
	2 項	はい	はい	はい
	XML	はい	はい	はい
	ROD/フラット	はい	はい	はい

表 17 は、バックエンド・システムからハブに文書を送信するときに各パッケージ化タイプおよびビジネス・プロトコルでサポートされるトランスポート・プロトコルを示しています。

表 17. バックエンド・システムから *WebSphere Partner Gateway* へのサポートされるトランスポート・プロトコル

パッケージ化のタイプ	ビジネス・プロトコル	HTTP または HTTPS	JMS	ファイル・システム
バックエンド統合	RosettaNet (RNSC)	はい	はい	いいえ
	ebMS	はい	はい	いいえ
	XML	はい	はい	いいえ
	2 項	はい	はい	いいえ

表 17. バックエンド・システムから WebSphere Partner Gateway へのサポートされるトランスポート・プロトコル (続き)

パッケージ化のタイプ	ビジネス・プロトコル	HTTP または HTTPS	JMS	ファイル・システム
	ROD/フラット	はい	はい	いいえ
なし	XML のみ	はい	はい	はい
	EDI (EDI について詳しくは、表 18を参照)			
	cXML のみ	はい	いいえ	いいえ
	SOAP のみ	はい	いいえ	いいえ
	2 項のみ	いいえ	いいえ	いいえ
	ROD/フラットのみ	はい	はい	はい

表 18 は、各種の EDI、XML、およびレコード指向データ (ROD)/フラット文書でサポートされるトランスポート・プロトコルおよびパッケージ化タイプを示しています。

表 18. WebSphere Partner Gateway と EDI のバックエンド・システムとの間でサポートされるトランスポート・プロトコル

パッケージ化のタイプ	文書	HTTP または HTTPS	JMS	ファイル・システム
バックエンド統合	単一トランザクションを含む単一交換 (エンベロープ内に X12 850 トランザクションがある場合など)	はい	はい	いいえ
	複数のトランザクションを含む単一交換 (同じエンベロープ内に X12 850 トランザクションと X12 890 トランザクションがある場合など)	はい	はい	いいえ
	単一トランザクションを含む複数の交換 (同じファイル内に 2 つの X12 エンベロープがあり、それぞれに単一トランザクションが含まれる場合など)	はい	はい	いいえ
	複数のトランザクションを含む複数の交換 (同じファイル内に 2 つの X12 エンベロープがあり、それぞれに複数のトランザクションが含まれる場合など)	はい	はい	いいえ
	EDI トランザクション (X12 850 トランザクションなど。トランザクションは EDI 交換の内部にある必要があるため、それ自体では送達できません)	いいえ	いいえ	いいえ
	後から EDI トランザクションに変換される文書 (XML など)	はい	はい	いいえ

表 18. WebSphere Partner Gateway と EDI のバックエンド・システムとの間でサポートされる  
 トラnsポート・プロトコル (続き)

パッケージ化の タイプ	文書	HTTP また は HTTPS	JMS	ファイル・ システム
なし	単一トランザクションを含む単 一交換	はい	はい	はい
	複数のトランザクションを含む 単一交換	はい	はい	はい
	単一トランザクションを含む複 数の交換	はい	はい	はい
	複数のトランザクションを含む 複数の交換	はい	はい	はい
	EDI トランザクション (未サポ ート、交換エンベロップが必要)	いいえ	いいえ	いいえ
	後から EDI トランザクションに 変換される文書 (XML など)	はい	はい	はい

上記の表は、ハブとバックエンド・システムの間で有効なトラnsポート・プロト  
 コルを示しています。ハブは、パートナーとの間で文書を送受信するために、追加  
 のトラnsポート・プロトコルを使用できます。例えば、ハブは、FTP スクリプ  
 ト・トラnsポートを使用して、リモート FTP サーバーに文書を送信できます。  
 FTP スクリプト・トラnsポートを使用して、文書を受信することもできます。  
 FTP スクリプト・トラnsポートの説明は、「*WebSphere Partner Gateway* ハブ構  
 成ガイド」に記載されています。FTP スクリプト・トラnsポートの使用は、イン  
 ターネット経由で文書を送受信する場合にはオプションですが、付加価値通信網  
 (VAN) 経由で文書を送受信する場合には必須です。

## HTTP トラnsポート・プロトコル

HTTP プロトコルを使用したメッセージを送信するには、WebSphere Partner  
 Gateway は HTTP/S 1.1 を使用します。バックエンド・システムからのメッセ  
 ージを受信するため、WebSphere Partner Gateway は HTTP/S バージョン 1.0 および  
 1.1 の両方をサポートします。

HTTP メッセージには統合パッケージ化属性を組み込むことができます。これらの  
 属性が組み込まれるかどうかは、パートナーの接続に関連付けられたパッケージ化  
 のタイプによって次のように異なります。

- パートナー接続が、HTTP メッセージにバックエンド統合パッケージ化を含める  
 ように指定した場合、HTTP メッセージのトラnsポート・レベル・ヘッダーに  
 は、内容のプロトコル、メッセージの ID、およびメッセージの送信者などのメ  
 セッセージに関する情報を格納する追加属性が組み込まれます。ヘッダー内のフィー  
 ルドの完全なリストについては、29 ページの『トラnsポート・レベル・ヘッ  
 ーの内容』を参照してください。

RosettaNet メッセージはバックエンド統合パッケージ化を使用する必要がありま  
 ず。

- パートナー接続がパッケージ化なしを指定した場合、HTTP メッセージにはこれらの追加属性は含まれず、WebSphere Partner Gateway がメッセージを構文解析してこの情報を取得します。

SOAP および cXML メッセージは、パッケージ化なしを使用する必要があります。

**注:** XML メッセージでは、「なし」とバックエンド統合パッケージ化のどちらを使用してもかまいません。同様に、EDI 文書では、「なし」とバックエンド統合のどちらを使用してもかまいません。バックエンド・システムから受信した 2 項メッセージは、バックエンド統合パッケージ化を使用していなければなりません。ただし、その逆は当てはまりません。WebSphere Partner Gateway ではアプリケーションへ 2 項メッセージを送信するにはどちらのタイプのパッケージ化も使用できるためです。

### プロセス:

#### このタスクについて

WebSphere Partner Gateway とアプリケーションの間で、非同期の交換用に HTTP または HTTPS メッセージが送信される場合は、以下のステップが発生します。

1. ソース・システム (WebSphere Partner Gateway またはバックエンド・システム) が固有の URL を使用してレシーバーシステムへの HTTP メッセージを通知します。
2. レシーバーシステムはメッセージを受信し、プロトコル・レベルの肯定応答 HTTP 200 または 202 を送信して所有権の変更を知らせます。ソース・システムはこの肯定応答メッセージの本文を無視します。この処理の間にエラーが発生した場合、レシーバーシステムは HTTP 500 メッセージをソース・システムに戻します。
3. WebSphere Partner Gateway がレシーバーシステムである場合 (つまり WebSphere Partner Gateway がメッセージを受信する場合) は、メッセージを永続化し、ソース・システムへの接続を解放します。
4. すると、レシーバーシステムは非同期でメッセージを処理できるようになります。

#### タスクの結果

交換が同期の場合 (SOAP または cXML 文書の場合など)、応答は同じ HTTP 接続で HTTP 200 メッセージとともに戻されます。

### HTTP プロトコルを使用したバックエンド・システムからのメッセージの送信: このタスクについて

HTTP プロトコルを使用して WebSphere Partner Gateway にメッセージを送信するには、バックエンド・システムは以下のステップを実行します。

1. メッセージを作成します。

トランスポート・レベル・ヘッダー内の Content-Type 属性で、メッセージに使用されるエンコード方式を指定しています。

2. 接続のパッケージ化タイプ設定に従ってメッセージをパッケージ化します。

バックエンド統合パッケージ化の場合、バックエンド・システムは WebSphere Partner Gateway が必要とするプロトコル・ヘッダー属性を追加します。

3. WebSphere Partner Gateway が使用する URL にメッセージを通知して、これらのメッセージを受信します。
4. 交換が同期の場合、バックエンド・システムは、要求に使用されたものと同じ接続での応答の受信を待ち合わせます。

HTTP メッセージをこの方向で交換できるようにするには、コミュニティー・コンソールの「レシーバーの詳細」ページを使用して、ハブでインバウンド文書のレシーバーをセットアップします。このレシーバーは URL を指定します。バックエンド・システムがハブに文書を送信するには、このアドレスを知っている必要があります。

#### HTTP プロトコルを使用したバックエンド・システムでのメッセージの受信: このタスクについて

HTTP プロトコルを使用して WebSphere Partner Gateway からメッセージを受信するために、バックエンド・システムは以下のステップを実行します。

1. 特定の URL でメッセージを listen します。
2. メッセージを受信したら、以下のようにメッセージを処理します。
  - 接続がパッケージ化なしである場合、バックエンド・システムはメッセージを構文解析して処理方法を決定する必要があります。
  - 接続がバックエンド統合パッケージ化を使用する場合、アプリケーションはバックエンド統合属性を使用してメッセージの処理方法を決定します。
3. 交換が同期の場合、バックエンド・システムは、要求に使用されたものと同じ接続で応答を戻します。

HTTP メッセージをこの方向で交換できるようにするには、コミュニティー・コンソールの「宛先」ページを使用して、バックエンド・システムでの文書の送達先を指定する宛先をセットアップします。

## JMS プロトコル

JMS プロトコルは Java<sup>TM</sup> Message Service (JMS) に基づくもので、IBM WebSphere MQ などが提供するトランザクションの永続 JMS キューを通じてメッセージを転送します。JMS プロトコルは、以下の JMS メッセージ・タイプをサポートします。

- StreamMessage (バイト配列として)
- BytesMessage (バイト配列として)
- TextMessage

JMS プロトコルでは、1 つのシステムが別のシステムに JMS メッセージを送信します。2 番目のシステムは、メッセージを受信した後で、そのメッセージをキューから除去します。この時点から、受信システムはメッセージを非同期に処理することができます。

JMS メッセージには統合パッケージ化属性を組み込むことができます。これらの属性が組み込まれるかどうかは、パートナーの接続に関連付けられたパッケージ化のタイプによって次のように異なります。

- パートナー接続が、JMS メッセージにバックエンド統合パッケージ化を組み込むように指定した場合、JMS メッセージには、メッセージ内部の JMS プロパティータとしてトランスポート・レベル情報 (内容のプロトコル、メッセージの ID、およびメッセージの送信者など) が含まれます。プロパティータの完全なリストについては、29 ページの『トランスポート・レベル・ヘッダーの内容』を参照してください。

**注:** WebSphere MQ JMS との互換性のため、JMS メッセージ内のプロパティータではプロパティータ名にハイフンではなく下線を使用します。例えば、JMS メッセージ内でプロパティータが `x_aux_system_msg_id` であれば、同等の HTTP ヘッダー・フィールドは `x-aux-system-msg-id` です。WebSphere Partner Gateway が JMS メッセージを処理するとき、このようなプロパティータの下線をハイフンに変換します。

- パートナー接続がパッケージ化なしを指定した場合、JMS メッセージにこれらの追加属性は含まれません。

2 項メッセージを除き、WebSphere Partner Gateway では JMS メッセージの送受信にどちらのタイプのパッケージ化も使用することができます。アプリケーションから受信した 2 項メッセージは、バックエンド統合パッケージ化を使用していなければなりません。ただし、その逆は当てはまりません。WebSphere Partner Gateway ではアプリケーションへ 2 項メッセージを送信する際にはどちらのタイプのパッケージ化も使用できるためです。

**JMS 環境の設定:** JMS 環境をセットアップするには、以下のプロバイダーが必要です。

- JMS プロバイダー

JMS プロバイダーは、メッセージング用の JMS API サポートのインプリメンテーションを提供します。文書の交換に使用するバックエンド・システムによって、使用する JMS プロバイダーが決定します。

- WebSphere Interchange Server を使用して文書を交換する場合は、JMS プロバイダーとして WebSphere MQ を使用します。
- WebSphere Process Server を使用して文書を交換する場合は、JMS プロバイダーとして WebSphere Platform Messaging およびサービス統合バスを使用します。WebSphere Process Server とともに使用できる JMS プロバイダーについての詳細は、98 ページの『トランスポートとしての JMS と WebSphere Process Server の統合』に記載されています。

通常の場合、JMS プロバイダーは、JMS 環境をセットアップするために使用できるプログラムを提供します。例えば、WebSphere MQ は JMSAdmin プログラムを提供します。このプログラムを使用して、JMS が必要とするオブジェクト (JMS 接続ファクトリーと JMS キュー・オブジェクト) を作成できます。これらのオブジェクトが作成されるときに、オブジェクトへの参照が JNDI 内に格納されます。

**注:** メッセージングの場合、WebSphere Partner Gateway は Point-to-Point モデルのみをサポートします。

- JNDI プロバイダー

JNDI プロバイダーは、JMS オブジェクトへの参照を格納するために使用される JNDI のインプリメンテーションを提供します。

バックエンド・アプリケーションが JMS プロトコルを使用して WebSphere Partner Gateway にビジネス文書を送信できるようにするには、JMS レシーバーを構成する必要があります。JMS レシーバーは JMS キューからメッセージを受信し、WebSphere Partner Gateway ワークフローに文書が導入されます。JMS レシーバー構成には、JNDI にアクセスするための必須パラメーターおよび JMS オブジェクトの名前が含まれます。バックエンド・システムとの統合のために、バックエンド・システムは、JMS レシーバー内で構成されたキューから JMS メッセージを送信します。

同様に、JMS 宛先は、パートナーがビジネス文書を受信するためのキューに WebSphere Partner Gateway がビジネス文書を送信するために使用されます。したがって、バックエンド・システムにメッセージを送信するには、内部パートナーのプロファイル内で JMS 宛先を構成する必要があります。この宛先は、バックエンド・システムが受信するキューに送信できるように構成する必要があります。JMS 宛先構成には、JNDI にアクセスするための必須パラメーターおよび JMS オブジェクトの名前が含まれます。

**JMS 環境の設定の概要:** WebSphere Partner Gateway およびバックエンド・システムが、JMS トランスポート・プロトコルで通信するには、通信のそれぞれの方向について JMS キューが必要です。そのため、以下の手順を実行して適切な JMS キューを提供する必要があります。

- JMS 環境を構成します。
- キュー・マネージャー、および必要なキュー (伝送キュー、リモート・キュー、および受信側キューなど) を作成します。

JMS キュー・マネージャーは、以下をはじめとする任意のコンピューターに置くことができます。

- バックエンド・システムがあるコンピューター
- WebSphere Partner Gateway があるコンピューター

さらには、バックエンド・システムがあるコンピューターと WebSphere Partner Gateway があるコンピューターの両方にキュー・マネージャーを置くこともできます。この場合は、セットアップ・チャンネルを使用して、2 つのキュー・マネージャーを結合してください。この方法を使用した場合、どちらの側もネットワーク上でクライアント接続する必要はありません。

WebSphere MQ バージョン 6.0 と最新のフィックスパックを使用して JMS トランスポート・プロトコル・メカニズムを構成する手順については、「*WebSphere Partner Gateway ハブ構成ガイド*」に記載されています。WebSphere Process Server を使用して文書を交換する場合に JMS 環境を構成する手順については、98 ページの『トランスポートとしての JMS と WebSphere Process Server の統合』に記載されています。

**JMS プロトコルを使用したバックエンド・システムからのメッセージの送信:**

## このタスクについて

JMS プロトコルを使用して WebSphere Partner Gateway にメッセージを送信するには、バックエンド・システムは以下のステップを実行します。

1. メッセージを作成します。
2. 接続のパッケージ化タイプ設定に従ってメッセージをパッケージ化します。

バックエンド統合パッケージ化の場合、アプリケーションは必要な JMS ヘッダー属性を追加します。

3. バックエンド・システムが WebSphere Partner Gateway にメッセージを送信するために使用する JMS キューに、メッセージを送信します。

## JMS プロトコルを使用したバックエンド・システムでのメッセージの受信: このタスクについて

JMS プロトコルを使用して WebSphere Partner Gateway からメッセージを受信するために、バックエンド・システムは以下のステップを実行します。

1. JMS キューでメッセージを listen します。
2. メッセージを受信したら、以下のようにメッセージを処理します。
  - 接続がパッケージ化なしである場合、バックエンド・システムはメッセージを構文解析して処理方法を決定する必要があります。
  - 接続がバックエンド統合パッケージ化を使用する場合、アプリケーションはバックエンド統合属性を使用してメッセージの処理方法を決定します。

## ファイル・システム・プロトコル

ファイル・システム・プロトコルによって、WebSphere Partner Gateway はメッセージを定義済みディレクトリー構造に配置して、メッセージを送信することができます。WebSphere Partner Gateway はディレクトリー構造からメッセージを読み込むことにより、メッセージを受信します。ファイル・システム・プロトコルは、パッケージ化なしタイプのみをサポートします。

## ファイル・システム・プロトコルを使用したバックエンド・システムからのメッセージの送信: このタスクについて

ファイル・システム・プロトコルを使用して WebSphere Partner Gateway にメッセージを送信するには、アプリケーションは以下のステップを実行する必要があります。

1. 一時ディレクトリー内にメッセージ・ファイルを作成します。
2. ファイルの用意ができれば、そのファイルを WebSphere Partner Gateway がポーリングするディレクトリーに移動します。

ファイル・システム・メッセージをこの方向で交換できるようにするには、コミュニティー・コンソールの「レシーバーの詳細」ページを使用して、インバウンド文書のレシーバーをセットアップします。メッセージの受信側によって、WebSphere Partner Gateway がポーリングするディレクトリーが決定します。レシーバーを作成すると、WebSphere Partner Gateway はレシーバー用の Documents ディレクトリーとそのサブディレクトリーを以下のように作成します。

```

<doc_root>
  Documents
    実動
    Test
</other destination types>

```

WebSphere Partner Gateway は、メッセージ・ファイルを検出するため定期的に Documents ディレクトリーとそのサブディレクトリーをポーリングします。メッセージを検出した場合、WebSphere Partner Gateway はメッセージを永続化し、次にディレクトリーからそのメッセージを削除します。次に、WebSphere Partner Gateway は正常にメッセージを処理します。レシーバーの作成方法について詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

**ファイル・システム・プロトコルを使用したバックエンド・システムでのメッセージの受信:**

**このタスクについて**

ファイル・システム・プロトコルを使用してメッセージを受信するには、アプリケーションは以下を実行する必要があります。

1. メッセージ・ファイルを検出するために適切なディレクトリーをポーリングします。
2. メッセージがあれば、それを永続化します。
3. ディレクトリーからメッセージを削除します。
4. メッセージを処理します。

ファイル・システム・メッセージをこの方向で交換できるようにするには、コミュニティー・コンソールの「宛先」ページを使用して、文書の送達先を指定する宛先をセットアップします。WebSphere Partner Gateway はメッセージ・ファイルを Documents ディレクトリーに置きますが、このディレクトリーは宛先によって定義されます。宛先に従って宛先ディレクトリーを定義することによって、各パートナー接続のディレクトリーを異なるものにすることができます。宛先について詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

## バックエンド・アプリケーションへのアクセス方法

WebSphere Partner Gateway には、多数のさまざまなバックエンド・アプリケーションとの統合機能があります。通常、バックエンド・アプリケーションへは、統合ブローカーなどのバックエンド・システムを介してアクセスします。本書では、表 19 にリストされたバックエンド・システムへの統合について取り上げます。

表 19. *WebSphere Partner Gateway* 用にサポートされるバックエンド・システム

バックエンド・システム	詳細の参照先
WebSphere Process Server	61 ページの『 <i>WebSphere Process Server</i> 統合の概要』
WebSphere Interchange Server	119 ページの『 <i>InterChange Server</i> 統合の概要』
WebSphere Message Broker	187 ページの『 <i>WebSphere Message Broker</i> との統合』

表 19. WebSphere Partner Gateway 用にサポートされるバックエンド・システム (続き)

バックエンド・システム	詳細の参照先
WebSphere Data Interchange	205 ページの『WebSphere Data Interchange との統合』

## メッセージ処理

このセクションでは、メッセージの送達に影響する以下の状態を WebSphere Partner Gateway がどのように処理するかを説明します。

- 『キューに入れられた送達』
- 『通信エラー処理』
- 51 ページの『重複メッセージ』

### キューに入れられた送達

WebSphere Partner Gateway は、特定の宛先に送信するすべての文書に関する情報をキューに通知します。文書マネージャーは、キューが受け取った順序でこれらのメッセージを処理し (FIFO)、各メッセージごとのスレッドを使用してそれらを送信します。宛先 (例えばトランスポート・プロトコルが HTTP であれば URL、トランスポート・プロトコルが JMS であれば JMS 宛先) がオフラインになるように構成されている場合 (『通信エラー処理』参照)、メッセージは宛先が使用可能 (オンライン) になるまでキューにとどまります。文書マネージャーは、スレッド内でエラーを受け取った場合、他のスレッドがメッセージを送信しようとするのを止めます。文書マネージャーは、エラーが発生したメッセージを送達できるようになるまで、これらのメッセージをキューに戻しておきます。

試行に失敗した回数が最大試行数を超えた場合、文書マネージャーはメッセージを失敗ディレクトリーに格納し、次に、宛先がオフラインである場合を除き、キュー内の次のメッセージの送達を試みます。

### 通信エラー処理

WebSphere Partner Gateway が送信者で、アプリケーションがエラー (例えば HTTP プロトコル使用時の 200 または 202 以外の HTTP 応答メッセージ) を戻したとき、WebSphere Partner Gateway は、この特定の宛先用にどのように構成されているかによって、再度そのメッセージの送信を試みる場合があります。各宛先 (HTTP の場合は URL) には、再試行の回数とメッセージの送信方法に影響する以下のオプションがあります。

表 20. 宛先構成オプション

構成オプション	説明
再試行カウント	エラーを受けとった場合の文書の再試行の回数
再試行間隔	再試行の時間間隔
オンライン/オフライン	送達の試行を開始および停止します。
スレッド数	宛先ごとにメッセージを処理する通知スレッドの数

WebSphere Partner Gateway がメッセージの送信を再試行するように構成されていない場合、あるいはすべての送達試行が失敗した場合、WebSphere Partner Gateway は以下のアクションのいずれかまたは全部を実行して問題を通知します。

- 文書ビューアーや RosettaNet ビューアーなどのコミュニティー・コンソールのさまざまなビューでエラーを表示する。
- 送達の失敗イベントの E メール・アラートが設定されている場合、E メールを適切な人に送信し、適切な処置ができるように問題を通知する。
- イベント文書を作成し、その文書をレシーバーに送信する。

詳細については、「WebSphere Partner Gateway 管理者ガイド」の『宛先構成の管理 (Managing destination configurations)』を参照してください。

## 重複メッセージ

WebSphere Partner Gateway に送受信されるすべてのメッセージには、グローバル固有 ID (GUID) が必要です。WebSphere Partner Gateway は GUID を使用して重複メッセージを検出します。バックエンド統合パッケージ化が使用される場合、各メッセージはトランスポート・レベル・ヘッダー内に GUID を格納しています。例えば HTTP プロトコルの場合、GUID は x-aux-system-msg-id フィールド内に含まれています (29 ページの『トランスポート・レベル・ヘッダーの内容』参照)。メッセージの送信者が GUID を生成します。ファイル・システム・プロトコルは重複メッセージの検査をサポートしません。

メッセージを送信しようとしてエラーが発生した場合、WebSphere Partner Gateway は再試行のたびにそのメッセージの GUID を再利用します。WebSphere Partner Gateway は、重複した GUID を含むメッセージを受信した場合 (HTTP 200 など)、肯定応答を返しますが、重複したメッセージは処理しません。

**注:** RosettaNet が使用されている場合、WebSphere Partner Gateway は、RosettaNet プロセス・レベルで重複メッセージを検査します。XML が使用されている場合も、重複メッセージを検査します。

## WebSphere Partner Gateway の構成

6 ページの『ハブ構成プロセス』では、ハブ管理者がハブを構成するステップの概要について説明しました。このセクションでは、バックエンド・システムで使用できるように WebSphere Partner Gateway を構成するステップについて説明します。この構成ステップでは、ハブ・コミュニティーに外部パートナーがすでに構成済みであることを前提としています。このセクションでは、特に、以下の構成が実行済みであることを前提としています。

- 内部パートナーのパートナー・プロファイルがすでに作成されている。
- 文書の発信元 (または宛先) の外部パートナーがすでに作成されている。
- 外部パートナーの B2B プログラムで、内部パートナーのパートナー・プロファイルが作成されている。
- レシーバーが定義されている。これにより、WebSphere Partner Gateway のレシーバーは、適切なトランスポート・プロトコルを使用した外部パートナーからの着信文書を listen できます。

- 外部パートナー (文書の送信元) のプロファイル内で B2B 機能が定義済みで、使用可能になっている。これにより、WebSphere Partner Gateway はそのソースから文書が送信されることを予期できます。
- 内部パートナーと外部パートナーの間にパートナー接続が存在し、それによって外部パートナーおよび内部パートナーが文書を受信 (または送信) できる。

注: ハブ管理者 (hubadmin) としてログインする必要があります。

ハブ・コミュニティをサポートするように WebSphere Partner Gateway を構成する方法について詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

外部パートナーの構成が完了したら、バックエンド・システムと通信できるように WebSphere Partner Gateway を構成する必要があります。このセクションでは、バックエンド・システムをハブ・コミュニティに組み込む方法について説明します。セクションの構成は以下のとおりです。

- 『バックエンド・システムへの文書の送信』
- 56 ページの『バックエンド・システムからの文書を受信』

## バックエンド・システムへの文書の送信

バックエンド・システムに文書を送信するために、WebSphere Partner Gateway は以下の手順を実行します。

1. 外部パートナーから文書を受信します。

レシーバーは、外部パートナーからの着信メッセージとその関連トランスポート・プロトコルに関してハブに定義されているレシーバーからこのソース文書を取り出します。文書をバックエンド・システムに送信するとき、ソース文書は、ある外部パートナーから受信する文書です。そのため、これはパートナー文書と呼ばれます。

2. パートナー文書を、バックエンド・システムが必要とするフォーマットの宛先文書に変換します。

WebSphere Partner Gateway の文書マネージャーが、この変換を実行して宛先文書とします。文書をバックエンド・システムに送信するとき、宛先文書は、バックエンド・システムへ送信される文書です。そのため、これはバックエンド文書と呼ばれます。

3. バックエンド文書をバックエンド・システムに送信します。

文書マネージャーは、バックエンド・システムへの発信メッセージに関してハブに定義されている宛先を通じて、バックエンド文書を送信します。

そのため、ハブがバックエンド・システムに文書を送信できるようにするには、表 21 にまとめられた構成が WebSphere Partner Gateway 内部で実行されたことを確認する必要があります。

表 21. バックエンド・システムに文書を送信するための構成手順

構成手順	WebSphere Partner Gateway 手順	詳細の参照先
1. 文書の送信先を定義します。	1. バックエンド・システムへの宛先を作成します。	53 ページの『パートナー文書の送信先の定義』

表 21. バックエンド・システムに文書を送信するための構成手順 (続き)

構成手順	WebSphere Partner Gateway 手順	詳細の参照先
2. 文書の処理方法を定義します。	<p>2. ソースと宛先のフォーマット用の文書フロー定義を作成します。</p> <p>3. バックエンド・システムへ送信される文書の文書フロー定義用に、B2B 機能を使用可能にします。</p> <p>4. ソースと宛先の文書フロー定義の間に、文書フロー定義の対話を作成します。</p>	54 ページの『パートナー文書の処理方法の定義』
3. バックエンド・システムへの接続方法を定義します。	5. バックエンド・システムへ文書を送信するパートナー接続を作成します。	56 ページの『バックエンド・システムへの接続方法の定義』

### パートナー文書の送信先の定義: このタスクについて

バックエンド・システムに文書を送信するために、ハブは宛先を定義しておく必要があります。この宛先は、変換済み文書の宛先を指定します。すなわち、ハブがバックエンド文書を送信するロケーションを (URI として) 指定します。このロケーションは、バックエンド・システムが着信メッセージを listen するロケーションと同じです。宛先は、(バックエンド・システム内部の) エンタープライズ・アプリケーション層への入り口ポイントを識別します。WebSphere Partner Gateway 内では、宛先の存在を確認するのは文書マネージャーです。文書マネージャーは、文書进行处理すると、宛先内の指定されたロケーションで、変換済みの文書をバックエンド・システムに送信します。

WebSphere Partner Gateway 内で宛先を定義するには、以下のようにします。

1. 「アカウント管理」 > 「プロファイル」 をクリックします。
2. 「検索」 をクリックして、パートナーのリストを表示します。
3. 内部パートナーの横にある「詳細の表示」 アイコンを選択します。
4. 「宛先」 をクリックします。
5. 「作成」 をクリックします。

宛先を定義するときに、ハブとバックエンド・システムの両方がバックエンド文書の転送に使用するトランスポート・プロトコルを指定します。41 ページの表 16 に示すように、トランスポート・プロトコルの選択は、文書のフォーマットによって異なります。このフォーマットには、文書フロー定義で定義されているパッケージ化タイプとビジネス・プロトコルが含まれます。

**注:** WebSphere Partner Gateway での宛先の作成方法については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

トランスポート・プロトコルの選択は、ご使用の特定のバックエンド・システムがサポートするトランスポート・プロトコルにも依存します。詳しくは、本書の特定のバックエンド・システムの統合に関する章を参照してください。

文書に有効なトランスポート・プロトコルを選択すると、宛先に関して定義する必要のあるその他の情報を「宛先」ビューで指定できます。

**パートナー文書の処理方法の定義:** 文書マネージャーは、パートナー文書を処理するためには、この文書をどのフォーマットに変換する必要があるかを知っていなければなりません。すなわち、バックエンド文書のフォーマットを知る必要があります。バックエンド統合の一部として、WebSphere Partner Gateway 内で以下のエンティティーが定義されていることを確認する必要があります。

- パートナー文書とバックエンド文書の両方のフォーマットを定義するため、文書定義が存在している必要があります。
- 内部パートナーの B2B 機能では、宛先 (ターゲット) としてのバックエンド文書の文書フロー定義を使用可能にする必要があります。
- ソースとしてのパートナー文書と宛先としてのバックエンド文書を一緒にまとめた、文書フロー定義の対話が存在しなければなりません。

**文書定義:** 文書定義はそれぞれ、WebSphere Partner Gateway が特定の文書を処理する方法を定義します。この定義には、文書のパッケージ化タイプとビジネス・プロトコルが含まれています。WebSphere Partner Gateway では、いくつかの事前定義パッケージ化タイプとプロトコル定義が用意されています。これらの事前定義フォーマットによってパートナー文書とバックエンド文書を正しく定義できるのであれば、自分で文書定義を定義する必要はありません。ただし、事前定義フォーマットによってパートナー文書とバックエンド文書を適切に定義できない場合は、その文書向けに有効な文書定義を作成する必要があります。WebSphere Partner Gateway 内で文書フロー定義を定義するには、「文書フロー定義の管理」ページ (「ハブ管理」 > 「ハブ構成」 > 「文書フロー定義」 > 「文書フロー定義の作成 (Create Document Flow Definition)」) を使用します。

**注:** 事前定義文書フロー定義および WebSphere Partner Gateway での文書フロー定義の作成方法について詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

バックエンド統合の場合、バックエンド文書のパッケージ化タイプは次のいずれかとする必要があります。

- パッケージ化なし
- バックエンド統合パッケージ化

文書のビジネス・プロトコルと使用している特定のバックエンド・システムに基づいて、これらのパッケージ化タイプのどちらが適用されるかを判断する必要があります。バックエンド・システム使用時のパッケージ化タイプについては、28 ページの『使用するパッケージ化』を参照してください。サポートされるバックエンド・システムについては、49 ページの『バックエンド・アプリケーションへのアクセス方法』を参照してください。

#### **送信用の B2B 機能の設定:**

##### **このタスクについて**

文書マネージャーがソース文書を変換するためには、必要とされる宛先文書のフォーマットを処理できるかどうかを判断する必要があります。これを判断するため、文書マネージャーは、内部パートナー・プロファイルの B2B 機能をチェックしま

す。B2B 機能は、内部パートナーに対してどの文書定義が使用可能になっているかを定義します。サポートされる文書定義は、コンポーネント文書タイプ (パッケージ化タイプ、ビジネス・プロトコル、文書タイプなど) をそれぞれ使用可能にします。特定の文書フロー定義を使用可能にするには、WebSphere Partner Gateway の「B2B 機能」ページを使用します。このビューにアクセスするには、以下の手順を実行します。

1. 「アカウント管理」>「プロファイル」>「パートナー」をクリックします。
2. 「検索」をクリックして、パートナーのリストを表示します。
3. 内部パートナーの横にある「詳細の表示」アイコンを選択します。
4. 「B2B 機能」をクリックします。
5. バックエンド統合の場合は、バックエンドに送信される各文書タイプが、「ターゲットの設定」文書の該当するパッケージ化、プロトコル、および文書タイプの下で使用可能になっていることを確認します。「ターゲットの設定」で、バックエンド文書の文書定義内の各文書タイプ・コンポーネントを使用可能にします。
6. また、ハブがバックエンド・システムから文書を受信する場合は、「B2B 機能」ページを表示したまま、必要な B2B 機能を使用可能にできます。この場合は、「ソースの設定」で、バックエンド文書の文書タイプ・コンポーネントを使用可能にします。

注: WebSphere Partner Gateway での B2B 機能の設定方法について詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

#### 送信用文書インタラクションの定義

文書マネージャーは、パートナー文書の変換方法を把握するために、パートナー文書とバックエンド文書の文書フロー定義を結合し、どちらがソース・パートナーでどちらが宛先パートナーであるかを識別するインタラクションを見つけることができる必要があります。

文書マネージャーは、変換済みの文書をバックエンド・システムに送信しようとするとき、ソース・パートナーと宛先パートナー (バックエンド・システム) 間のパートナーの接続を見つけることができなければなりません。ただし、パートナー接続が存在するためには、ソース文書と宛先文書間に有効なインタラクションが存在している必要があります。WebSphere Partner Gateway 内で文書フロー定義インタラクションを定義するには、「ハブ管理」>「ハブ構成」>「文書定義」>「インタラクションの管理」>「インタラクションの作成」をクリックします。

注: WebSphere Partner Gateway での文書フロー定義インタラクションの作成方法について詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

バックエンド・システムに文書を送信する場合は、表 22 に示すように、ソース文書と宛先 (ターゲット) 文書間のインタラクションを定義します。

表 22. 文書送信用の対話の作成

「インタラクションの管理」セクション	アクション
ソース	パートナー 文書の文書タイプ定義内で、コンポーネント文書タイプを選択します。

表 22. 文書送信用の対話の作成 (続き)

「インタラクションの管理」 セクション	アクション
ターゲット	バックエンド 文書の文書タイプ定義内で、コンポーネント文書タイプを選択します。

### バックエンド・システムへの接続方法の定義:

文書マネージャーは、変換済み文書をバックエンド・システムに送信するためには、有効なパートナー接続を検出する必要があります。パートナーの接続はソース・パートナーおよび宛先パートナーを識別し、これら 2 つのパートナーが通信するロケーションを提供します。パートナー接続を作成するには、WebSphere Partner Gateway 内の「接続の管理」ページを使用します。このビューにアクセスするには、「アカウント管理」 > 「接続」をクリックします。

ただし、パートナー接続を定義するためには、ソース文書と宛先文書間に文書定義の対話がすでに存在している必要があります。「接続の管理」ビューで、まずソース・パートナーと宛先 (ターゲット) パートナーを指定して、既存のインタラクションがあるかどうかを調べます。表 23 に、バックエンド・システムへの文書送信用のパートナー接続を定義するために「接続の管理」ページで選択するパートナーをリストします。

表 23. 文書送信用のパートナー接続の作成

「接続の管理」 ドロップダウン・リスト	外部パートナーの名前
ソース	文書を内部パートナーに送信している外部パートナーの名前
ターゲット	外部パートナーから文書を受信する内部パートナーの名前

ソースおよびターゲットを指定してから、「検索」をクリックして既存の文書フロー定義のインタラクションがあるかどうかを調べます。インタラクションが存在しない場合、パートナー接続の作成を続行する前に、インタラクションを作成する必要があります。インタラクションが存在する場合は (ソースがパートナー文書フロー定義、ターゲットがバックエンド文書フロー定義であるインタラクション)、バックエンド・システムとの通信用のパートナー接続を構成できます。

**注:** WebSphere Partner Gateway でのパートナー接続の作成方法については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

バックエンド統合の場合、このパートナー接続は、53 ページの『パートナー文書の送信先の定義』で定義した宛先をターゲット宛先として指定する必要があります。

### バックエンド・システムからの文書の受信

バックエンド・システムから文書を受信するために、ハブは以下の手順を実行します。

1. バックエンド・システムから文書を受信します。

WebSphere Partner Gateway のレシーバーは、バックエンド・システムからの着信メッセージと関連トランスポート・プロトコルに関してハブに定義されている

レシーバーからこのソース文書を取り出します。バックエンド・システムから文書を受信する場合、ソース文書は、バックエンド・システムから受信する文書です。そのため、この文書はバックエンド文書と呼ばれます。

2. バックエンド文書を、指定された外部パートナーが必要とするフォーマットの宛先 (ターゲット) 文書に変換します。

文書マネージャーが、この変換を実行して宛先文書とします。文書をバックエンド・システムから受信するとき、宛先文書は、ある外部パートナーに送信される文書です。そのため、この文書はパートナー文書と呼ばれます。

3. パートナー文書を適切な外部パートナーに送信します。

文書マネージャーは、適切な外部パートナーへの発信メッセージに関してハブに定義されている宛先を通じて、パートナー文書を送信します。

そのため、ハブがバックエンド・システムから文書を受信できるようにするには、表 24 にまとめられた構成が WebSphere Partner Gateway 内部で実行されたことを確認する必要があります。

表 24. バックエンド・システムから文書を受信するための構成手順

構成手順	WebSphere Partner Gateway 手順	詳細の参照先
1. 文書を取り出す場所を定義します。	1. バックエンド・システムから着信メッセージを受信するレシーバーを作成します。	『バックエンド文書を取り出す場所の定義』
2. 文書の処理方法を定義します。	2. ソースと宛先のフォーマット用の文書フロー定義を作成します。 3. バックエンド・システムから受信される文書の文書フロー定義用に、B2B 機能を使用可能にします。 4. ソースと宛先の文書フロー定義の間に、文書フロー定義の対話を作成します。	58 ページの『バックエンド文書の処理方法の定義』
3. WebSphere Partner Gateway への接続方法を定義します。	5. WebSphere Partner Gateway へ文書を送信するパートナー接続を作成します。	59 ページの『WebSphere Partner Gateway への接続方法の定義』

**バックエンド文書を取り出す場所の定義:** バックエンド・システムから文書を受信するために、ハブはレシーバーを定義しておく必要があります。このレシーバーは、文書のソースを指定します。すなわち、ハブが着信文書を listen するロケーションを (URI として) 示します。このロケーションは、バックエンド・システムが文書を送信するロケーションと同じです。レシーバーは (WebSphere Partner Gateway 内部での) レシーバーへの入り口ポイントを示します。WebSphere Partner Gateway 内では、WebSphere Partner Gateway への文書を受信するのはレシーバーです。レシーバーは、文書を処理すると、後で文書マネージャーが取り出すことができるように変換済みの文書を永続共用ストレージに保管します。

WebSphere Partner Gateway 内のレシーバーを定義するには、「ハブ管理」>「ハブ構成」>「レシーバー」をクリックします。

**注:** WebSphere Partner Gateway でのレシーバーの作成方法について詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

レシーバーを定義するときに、ハブとバックエンド・システムの両方がバックエンド文書の転送に使用するトランスポート・プロトコルを指定します。41 ページの表 17 に示すように、トランスポート・プロトコルの選択は、文書のフォーマットによって異なります。このフォーマットには、文書フロー定義で定義されているパッケージ化タイプとビジネス・プロトコルが含まれます。

**注:** トランスポート・プロトコルの選択は、ご使用の特定のバックエンド・システムがサポートするトランスポート・プロトコルにも依存します。詳しくは、本書の特定のバックエンド・システムへの統合に関する章を参照してください。

文書に有効なトランスポート・プロトコルを選択すると、レシーバーに関して定義する必要のあるその他の情報を「レシーバーの詳細」ビューで指定できます。

#### バックエンド文書の処理方法の定義:

文書マネージャーは、バックエンド文書を処理するためには、この文書をどのフォーマットに変換する必要があるかを知っていなければなりません。すなわち、パートナー文書のフォーマットを知る必要があります。バックエンド統合の一部として、表 25 にまとめられているエンティティが WebSphere Partner Gateway 内で定義されていることを確認する必要があります。

表 25. バックエンド文書の変換方法の定義

手順	詳細の参照先
1. パートナー文書とバックエンド文書の両方のフォーマットを定義するため、文書定義が存在している必要があります。	54 ページの『文書定義』
2. 内部パートナーの B2B 機能では、ソースとしてのバックエンド文書の文書定義を使用可能にする必要があります。	『受信用の B2B 機能の設定』
3. ソースとしてのバックエンド文書と宛先としてのパートナー文書を一緒にまとめた、文書定義のインタラクションが存在しなければなりません。	『受信用の対話の定義』

**受信用の B2B 機能の設定:** バックエンド統合に適用されるとき B2B 機能の要約については、54 ページの『送信用の B2B 機能の設定』を参照してください。

バックエンド文書の文書タイプ・コンポーネントをまだ使用可能にしていない場合は、「ソースの設定」で使用可能にします。

**受信用の対話の定義:** バックエンド統合に適用されるとき文書フロー定義の対話の要約については、55 ページの『送信用文書インタラクションの定義』を参照してください。このセクションでは、バックエンド・システムからの文書の受信用対話を定義する方法を要約しています。

バックエンド・システムから文書を受信する場合は、59 ページの表 26 に示すように、ソース文書と宛先 (ターゲット) 文書の文書フロー定義間のインタラクションを定義します。

表 26. 文書受信用の対話の作成

「インタラクションの管理」セクション	アクション
ソース	バックエンド 文書の文書フロー定義内で、コンポーネント文書タイプを選択します。
ターゲット	パートナー 文書の文書フロー定義内で、コンポーネント文書タイプを選択します。

**WebSphere Partner Gateway への接続方法の定義:** 文書マネージャーは、バックエンド・システムからの文書进行处理するためには、有効な接続を検出する必要があります。接続はソース・パートナーおよび宛先パートナーを識別し、これら 2 つのパートナーが通信するロケーションを提供します。バックエンド統合に適用される際の接続の要約については、56 ページの『バックエンド・システムへの接続方法の定義』を参照してください。

表 27 に、バックエンド・システムから文書を受信するためのパートナー接続を定義するときに WebSphere Partner Gateway の「接続の管理」ページで選択するパートナーを示します。

表 27. 文書受信用のパートナー接続の作成

「接続の管理」ドロップダウン・リスト	外部パートナーの名前
ソース	内部パートナーの名前
ターゲット	文書を内部パートナーから受信している外部パートナーの名前

ソースおよびターゲットを指定してから、「**検索**」をクリックして既存の文書フロー定義のインタラクションがあるかどうかを調べます。インタラクションが存在しない場合、パートナー接続の作成を続行する前に、インタラクションを作成する必要があります。インタラクションが存在する場合は (ソースがバックエンド文書フロー定義、ターゲットがパートナー文書定義であるインタラクション)、バックエンド・システムとの通信用のパートナー接続を構成できます。

**注:** WebSphere Partner Gateway でのパートナー接続の作成方法については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。



---

## 第 3 章 WebSphere Process Server との統合

以下の章では、WebSphere Partner Gateway を WebSphere Process Server と統合する手順について詳しく説明します。

---

### WebSphere Process Server 統合の概要

この章では、WebSphere Partner Gateway と WebSphere Process Server との統合の概要について説明します。

**注:** WebSphere Partner Gateway とバックエンド・システムの統合に使用する一般的なプロセスについては、11 ページの『バックエンド統合の計画』を参照してください。

この章では以下のセクションを扱います。

- 『概説』
- 63 ページの『WebSphere Process Server との統合の計画』
- 66 ページの『WebSphere Partner Gateway と WebSphere Process Server を統合するための作業の概要』
- 68 ページの『バックエンド統合パッケージ化メッセージの処理』
  - 82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』
  - 87 ページの『BCGBackEndIntegrationJMSDataBindingImpl クラス』

### 概説

このセクションでは、WebSphere Partner Gateway を使用して WebSphere Process Server に B2B 機能を提供する方法について説明します。

**注:** WebSphere Process Server について詳しくは、WebSphere Process Server のインフォメーション・センターを参照してください。

WebSphere Process Server は、WebSphere Partner Gateway の B2B 機能を使用して取引パートナー (WebSphere Partner Gateway では外部パートナーと呼びます) との対話を管理します。例えば、WebSphere Process Server で実行されているサービスが文書を外部パートナーに送信する必要があるとします。WebSphere Process Server が文書を WebSphere Partner Gateway に送信すると、WebSphere Partner Gateway が、外部パートナーが要求する形式に文書を変換するために使用する変換マップ (必要な場合) を判別します。WebSphere Partner Gateway は、すべての外部パートナー・プロファイル情報も処理します。

ここで、WebSphere Process Server が文書を複数の外部パートナーに送信する必要がある場合について考えてみます。外部パートナーは、それぞれ異なる形式で文書を受信します。WebSphere Process Server が処理する必要がある形式は 1 種類 (WebSphere Partner Gateway に送信するもの) のみです。外部パートナーとのインタラクションは WebSphere Partner Gateway が処理します。

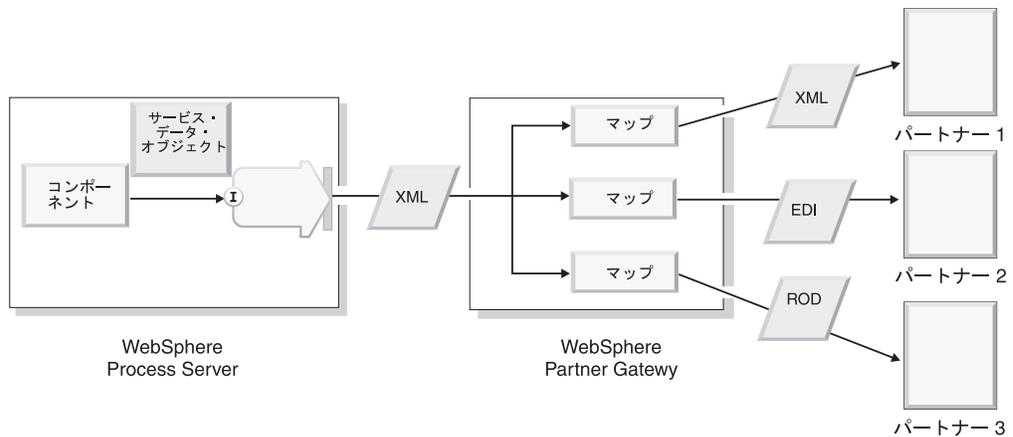


図 15. WebSphere Partner Gateway は、複数の形式の文書を外部パートナーに送信します

外部パートナーによって要求される形式に文書を変換する変換マップを作成できます。また、Data Interchange Services クライアント・プログラムからマップをインポートすることもできます。

同様に、外部パートナーから送信された文書を WebSphere Process Server が受信するときには、それらの文書が WebSphere Partner Gateway によって処理されます。さまざまな形式の文書を扱うことができます。WebSphere Partner Gateway は文書を変換し、WebSphere Process Server の内部パートナーに対して定義された宛先に送信します。

## WebSphere Process Server と WebSphere Partner Gateway が通信する仕組み

WebSphere Partner Gateway は外部パートナーからの文書を WebSphere Process Server に送信し、WebSphere Process Server 上のサービスが文書を処理できるようにします。WebSphere Process Server 上のサービスを他のアプリケーション (WebSphere Partner Gateway を含みます) が使用できるようにするためには、エクスポート・バインディングを使用します。

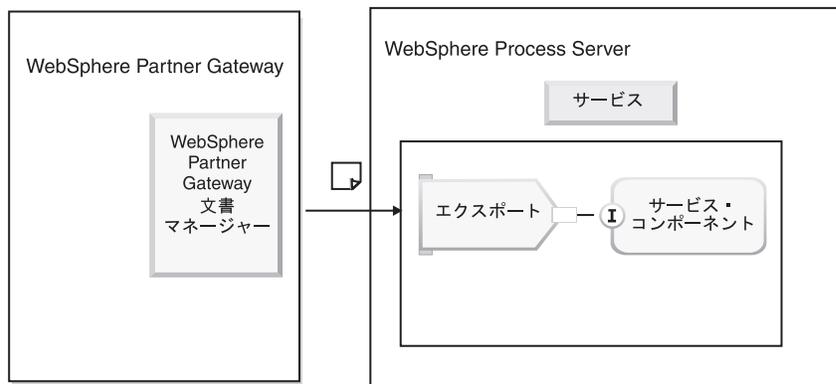


図 16. WebSphere Partner Gateway がエクスポート・バインディングを使用して文書を WebSphere Process Server に送信する仕組み

WebSphere Process Server コンポーネントは、サービスを記述するインターフェース (使用可能なメソッドおよび入出力データ) とバインディング (この場合はエクスポート・バインディング) を持ちます。

62 ページの図 16 に、エクスポート・バインディングが WebSphere Process Server 上のサービスを呼び出す仕組みの概要を示します。メッセージの送信に使用するトランスポート・タイプ (JMS や HTTP など) は、メッセージの送信先および取得方法にも影響します (詳しくは、64 ページの『WebSphere Process Server がサポートするメッセージ・トランスポート』を参照してください)。

同様に、WebSphere Process Server 上のサービスがビジネス文書を外部パートナーに送信するときには、インポート・バインディングを使用します。インポートは、モジュールの外部にあるサービスを識別し、モジュール内から呼び出せるようにします。この場合は、WebSphere Process Server がインポート・バインディングを使用して WebSphere Partner Gateway を呼び出し、WebSphere Partner Gateway が文書を処理して外部パートナーに送信します。

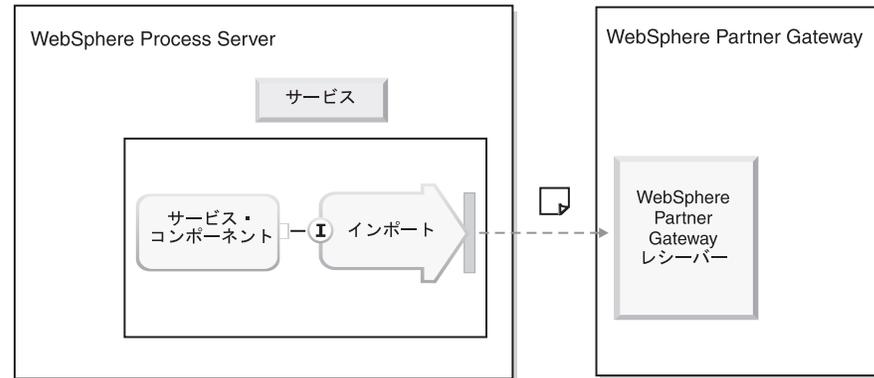


図 17. WebSphere Process Server がインポート・バインディングを使用して WebSphere Partner Gateway を呼び出す仕組み

## WebSphere Process Server との統合の計画

WebSphere Process Server との統合を計画するには、11 ページの『バックエンド統合の計画』で説明されている手順を実行してください。

### WebSphere Partner Gateway がサポートする WebSphere Process Server のバージョン

WebSphere Partner Gateway バージョン 6.2 は、WebSphere Process Server バージョン 6.0.2 との統合をサポートします。

WebSphere Process Server は、Windows 2000 や UNIX ベースのプラットフォームを含む、いくつかのプラットフォームで使用可能です。詳しくは、WebSphere Process Server のインフォメーション・センターに収録されている WebSphere Process Server のインストール・ガイドを参照してください。

## サポートされるインストール・シナリオ

以下の表の各行に、サポートされる WebSphere Partner Gateway インストールと WebSphere Process Server インストールの組み合わせを示します。

注: インストール中に、WebSphere Partner Gateway はコンポーネントごとに別個の WebSphere Application Server プロファイルを作成します。これらのプロファイルは、WebSphere Partner Gateway で使用されるだけです。WebSphere Process Server などの WebSphere Application Server アプリケーションをこれらのプロファイルにデプロイしないでください。

表 28. サポートされるインストール・シナリオ

WebSphere Partner Gateway インストール	WebSphere Process Server インストール	詳細
WebSphere Application Server 6.1 にインストールされているか、WebSphere Application Server Network Deployment 6.1 の 1 つ以上のインスタンスにインストールされている。	WebSphere Process Server のインストーラーが、サポートされる WebSphere Application Server ND を WebSphere Process Server とともにインストールする。	
WebSphere Application Server 6.1 にインストールされているか、WebSphere Application Server Network Deployment 6.1 の 1 つ以上のインスタンスにインストールされている。	WebSphere Partner Gateway のインストール先である WebSphere Application Server 6.1 と同じインストール先にインストールされているが、インスタンスは異なる。WebSphere Process Server に使用するプロファイルは、WebSphere Partner Gateway コンポーネントが使用するプロファイルと異ならない。	これは、WebSphere Partner Gateway と WebSphere Process Server の両方でサポートされるプラットフォーム (オペレーティング・システム およびバージョン) でのみサポートされる。

## WebSphere Process Server がサポートするメッセージ・トランスポート

WebSphere Partner Gateway が特定のメッセージ・トランスポート・プロトコルを介して WebSphere Process Server にメッセージを送信すると、内部パートナーに対して定義されたトランスポート固有の宛先が WebSphere Process Server エンドポイントにメッセージを送信します。WebSphere Process Server は、エンドポイントからメッセージを取得してメッセージを処理します。メッセージ・トランスポートのタイプにより、メッセージが WebSphere Process Server に到着した後の処理方法が以下のように判別されます。

- JMS トランスポートの場合は、JMS キューからメッセージを取得するための JMS エクスポートをセットアップします。
- HTTP トランスポートの場合は、WebSphere Partner Gateway からのメッセージの受信を処理するサーブレットを WebSphere Process Server に作成します。

- SOAP 文書 (HTTP トランスポート・プロトコルを介して送信されます) の場合は、SOAP 要求を取得するための Web サービス・エクスポート・バインディングを WebSphere Process Server でセットアップします。
- ファイル・システムの場合は、メッセージを WebSphere Process Server に送付するためのインバウンド・フラット・ファイル・アダプターをセットアップします。

WebSphere Process Server から WebSphere Partner Gateway にメッセージを送信するときには、WebSphere Partner Gateway 上のトランスポート固有のレシーバー (JMS キューや URL など) に送信します。メッセージ・トランスポートのタイプにより、メッセージの送信方法が判別されます。

- JMS トランスポートの場合は、JMS キューにメッセージを送信するための JMS インポートをセットアップします。
- HTTP トランスポートの場合は、WebSphere Partner Gateway レシーバーに指定された URL に対して HTTP POST を実行するコンポーネントを WebSphere Process Server に作成します。
- SOAP 文書 (HTTP トランスポート・プロトコルを介して送信されます) の場合は、SOAP 要求を WebSphere Partner Gateway URL に送信するための Web サービス・インポート・バインディングを WebSphere Process Server でセットアップします。
- ファイル・システムの場合は、メッセージを WebSphere Partner Gateway 上のディレクトリーに送付するためのアウトバウンド・フラット・ファイル・アダプターをセットアップします。

サポートされるトランスポートを使用したメッセージの送受信について詳しくは、以下のセクションを参照してください。

- 90 ページの『トランスポートとしての HTTP と WebSphere Process Server の統合』
- 98 ページの『トランスポートとしての JMS と WebSphere Process Server の統合』
- 110 ページの『WebSphere Process Server と SOAP/HTTP の統合』
- 114 ページの『トランスポートとしてのファイル・システムと WebSphere Process Server の統合』

## WebSphere Process Server 統合のサポート

このセクションでは、WebSphere Process Server との統合を支援するために WebSphere Partner Gateway が提供するサンプル、資料、およびユーティリティー・クラスについて説明します。

### サンプル:

WebSphere Partner Gateway には、WebSphere Process Server との統合処理を支援するための JMS トランスポート・プロトコルの使用例が用意されています。これらのサンプルは、WebSphere Partner Gateway 製品ディレクトリー内の以下のサブディレクトリーにあります。

Integration/WBI/WPG/samples

表 29 に、samples ディレクトリーのサブディレクトリーをリストします。

表 29. WebSphere Process Server 統合のサンプル

サンプルのタイプ	Samples サブディレクトリー
汎用のサンプル	JMS
RosettaNet 固有のサンプル	RosettaNet/JMS

#### 文書:

本書の説明のほか、WebSphere Partner Gateway には「*PIP Sample for WebSphere Process Server*」という資料も用意されています。この資料では、外部パートナーと WebSphere Process Server の間の PIP フローをセットアップするための方法について順を追って説明しています。

#### ユーティリティー・クラス:

WebSphere Partner Gateway には、以下の 2 つのユーティリティー・クラスが提供されています。これらを使用して、バックエンド統合パッケージ化メッセージをビジネス・オブジェクトに変換したり、ビジネス・オブジェクトをバックエンド統合パッケージ化メッセージに変換したりすることができます。

- 82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』

このクラスは 82 ページの『DataBinding インターフェース』を実装し、ストリング、ストリーム、およびバイト配列を読み書きするためのユーティリティー・メソッドを提供します。このクラスをそのまま使用することも、このクラスを使用して新しいデータ・バインディングを開発することもできます。

- 87 ページの『BCGBackEndIntegrationJMSDataBindingImpl クラス』

このクラスは 88 ページの『JMSDataBinding インターフェース』を実装します。このクラスは、SCA JMS エクスポート・バインディングおよびインポート・バインディングで指定できます。これは、ペイロードを含む JMS メッセージからデータ・オブジェクトを作成したり、データ・オブジェクトを JMS メッセージに書き込んだりします。

## WebSphere Partner Gateway と WebSphere Process Server を統合するための作業の概要

このセクションでは、WebSphere Partner Gateway が WebSphere Process Server と文書を送受信できるようにするために行う作業を示します。これは、WebSphere Process Server との統合方法について説明するトランスポート固有の章の基礎にもなります。

### WebSphere Partner Gateway システム側

このセクションでは、WebSphere Partner Gateway と文書を送受信できるようにハブを構成するために行う作業の概要について説明します。これらの作業は WebSphere Partner Gateway のコミュニティー・コンソールで行います。詳しくは、

「*WebSphere Partner Gateway ハブ構成ガイド*」を参照してください。

- WebSphere Process Server または外部パートナーからハブに送信された文書を受信するためのトランスポート固有のレシーバーをハブに作成します。

- 内部パートナー・プロファイルがまだ存在しない場合は作成します (WebSphere Partner Gateway が文書を WebSphere Process Server に送信するために使用するトランスポート固有の宛先を含めます)。
- 外部パートナー・プロファイルを作成します (WebSphere Partner Gateway が文書をパートナーに送信するために使用するトランスポート固有の宛先を含めます)。
- WSDL ファイル、変換マップ、RosettaNet パッケージ、またはその他の文書定義手段をインポートして、交換する文書タイプの文書定義が WebSphere Partner Gateway のコミュニティー・コンソールの「文書フロー定義」ページに表示されるようにします。
- ハブが (WebSphere Process Server または外部パートナーから) 受信する文書のタイプと、ハブが (WebSphere Process Server または外部パートナーに) 送信する文書のタイプの間のインタラクションを作成します。
- 内部パートナーおよび外部パートナーのプロファイルに B2B 機能を作成し、送受信可能な文書タイプを指定します。
- 内部パートナーと外部パートナーの間のパートナー接続を作成して、ソース・パートナー (文書の送信側)、ターゲット・パートナー (文書の受信側)、およびハブが文書を変換するために実行するアクション (ある場合) を指定します。

## WebSphere Process Server システム側

サービスをアセンブルおよびデプロイするために使用する WebSphere Process Server の成果物がモジュールになります。最初のステップは、WebSphere Integration Developer を使用したモジュールの作成です。

モジュールを作成したら、コンポーネントおよびインターフェースを作成し、使用するバインディングを指定します。

1. コンポーネントのインターフェースを指定します。インターフェースをインポートするか (例えば既存の WSDL ファイルをインポートできます)、インターフェースを作成できます。

インターフェースを作成するときには、コンポーネントが実行する 1 つ以上の操作を定義し、コンポーネントが想定する入力および出力を定義します。

コンポーネントのインターフェースを WSDL にするか、インターフェースの Java 実装を使用することができます。WSDL インターフェースを作成する場合、および Java インターフェースを作成する場合の詳細については、WebSphere Process Server の資料を参照してください。

2. コンポーネントの実装を指定します。実装をインポートするか (例えば既存の Java プログラムをインポートできます)、実装を作成できます。
3. WebSphere Integration Developer のアセンブリ・エディターを使用してアプリケーションを構成します。サービス・コンポーネントを作成し、コンポーネント (ステップ 1 で作成またはインポートしたもの) のインターフェースを指定します。実装 (ステップ 2 で作成またはインポートしたもの) も指定します。
4. WebSphere Partner Gateway が文書をサービスに送信できるようにするには、エクスポート・バインディングを作成します。WebSphere Process Server が文書を WebSphere Partner Gateway に送信できるようにするには、インポート・バインディングを作成します。

バインディングを作成するときには、文書を送受信するために必要な情報を指定します。例えば、JMS バインディング定義では、JMS キューおよびバスのほか、ビジネス文書とビジネス・オブジェクトとの相互変換に使用するデータ・バインディングも指定します。(データ・バインディングの使用および要件について詳しくは、『バックエンド統合パッケージ化メッセージの処理』を参照してください。) サービス・バインディングに対するトランスポート固有の要件については、後述の章で説明します。

5. モジュールを構成するコンポーネントのアセンブルが完了したら、モジュールをデプロイします。

## バックエンド統合パッケージ化メッセージの処理

RosettaNet などの一部のプロトコルの場合、WebSphere Partner Gateway はバックエンド統合パッケージ化を使用するためにバックエンド・アプリケーションを必要とします。WebSphere Partner Gateway は、HTTP プロトコルおよび JMS プロトコルを介したバックエンド統合パッケージ化をサポートします。このためには、WebSphere Process Server サービスがバックエンド統合パッケージ化メッセージを処理できる必要があります。WebSphere Process Server サービスはビジネス・オブジェクトを使用します。したがって、バックエンド統合パッケージ化メッセージを WebSphere Partner Gateway に送信するためには、WebSphere Process Server サービスがビジネス・オブジェクトをバックエンド統合パッケージ化メッセージに直列化する必要があります。同様に、WebSphere Partner Gateway からバックエンド統合パッケージ化メッセージを受信するためには、WebSphere Process Server サービスがバックエンド統合パッケージ化メッセージをビジネス・オブジェクトに非直列化する必要があります。

このセクションでは、WebSphere Partner Gateway が提供するデータ・バインディング・ユーティリティー API および JMS データ・バインディングについて説明します。WebSphere Process Server サービスは、これらを使用してバックエンド統合パッケージ化メッセージを処理できます。

データ・バインディングとは、以下の目的に使用する機構を指します。

- WebSphere Process Server から送信されたビジネス・オブジェクトを、WebSphere Partner Gateway が処理可能なビジネス文書に変換します。
- WebSphere Partner Gateway から送信されたビジネス文書を、WebSphere Process Server が処理可能なビジネス・オブジェクトに変換します。

WebSphere Partner Gateway 提供のバックエンド統合パッケージ化データ・バインディング・ユーティリティー API および JMS データ・バインディングにより、ビジネス・オブジェクトからバックエンド統合パッケージ化メッセージを作成したり、バックエンド統合パッケージ化メッセージをビジネス・オブジェクトに変換したりすることができます。WebSphere Partner Gateway 提供のデータ・バインディングは、バックエンド統合パッケージ化と文書を送受信するときそのまま使用できます。また、処理をカスタマイズする場合は、WebSphere Partner Gateway によって提供されるデータ・バインディング・ユーティリティー API を使用して独自のデータ・バインディングを作成できます。

## JMS トランスポート

トランスポート・プロトコルとして JMS を使用し、バックエンド統合パッケージ化を持たない (つまりパッケージ化なしが指定されている) XML 文書を送信または受信する場合は、WebSphere Process Server が提供するデフォルトの JMS バインディングを使用するか、WebSphere Process Server 提供のビジネス・オブジェクト API を使用して独自のデータ・バインディングを作成できます。バックエンド統合パッケージ化を持たない (つまりパッケージ化なしが指定されている) XML 以外の文書を送信または受信する場合は、独自のデータ・バインディングを作成する必要があります。SCA (Service Component Architecture) JMS エクスポート・バインディングまたはインポート・バインディングを構成するときには、JMS バインディングを選択します。デフォルトの JMS バインディングの使用について詳しくは、WebSphere Process Server のインフォメーション・センターを参照してください。

トランスポート・プロトコルとして JMS を使用し、バックエンド統合パッケージ化を持つ文書を送信または受信する場合は、BCGBackEndIntegrationJMSDataBindingImpl データ・バインディングをそのまま使用できます。処理をカスタマイズする場合は、BCGBackEndIntegrationJMSDataBindingImpl を使用して独自のデータ・バインディングを作成できます。BCGBackEndIntegrationJMSDataBindingImpl を使用するには、SCA JMS エクスポート・バインディングおよびインポート・バインディングでデータ・バインディングとして構成します。

- 要求処理の場合、エクスポートは JMSDataBinding インターフェースの read メソッドを呼び出し、WebSphere Partner Gateway からのバックエンド統合パッケージ化 JMS メッセージをビジネス・オブジェクトに変換します。要求応答処理の場合、エクスポートは write メソッドも呼び出す場合があります。
- 要求処理の場合、インポートは JMSDataBinding インターフェースの write メソッドを呼び出し、ビジネス・オブジェクトからバックエンド統合パッケージ化 JMS メッセージを構成した後、そのメッセージを WebSphere Partner Gateway に送信します。要求応答処理の場合、インポートは read メソッドを呼び出し、サービスによって提供される応答を読み取ることもできます。

データ・バインディングのメソッドの呼び出し方法およびその時点については、WebSphere Process Server のインフォメーション・センターを参照してください。

## HTTP トランスポート

トランスポート・プロトコルとして HTTP を使用する場合は、汎用の 82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』を拡張および指定変更する HTTP データ・バインディング・クラスを作成できます。

- HTTP サブレットは BCGBackendIntegrationDataBindingUtil のメソッドを呼び出し、WebSphere Partner Gateway から送信される HTTP ストリームからビジネス・オブジェクトを構成できます。その後、サブレットは、このビジネス・オブジェクトを使用してサービスを呼び出すことができます。
- WebSphere Partner Gateway に対する HTTP POST を実行するコンポーネントを作成できます。サービスは、ビジネス・オブジェクトとともにこのコンポーネントを呼び出すことができます。コンポーネントがバックエンド統合パッケージ化

メッセージを必要とする場合は、BCGBackendIntegrationDataBindingUtil クラスを使用してビジネス・オブジェクトからメッセージを構成できます。

**注:** WebSphere Partner Gateway は、バックエンド統合パッケージ化に対する要求専用呼び出しをサポートします。ただし、(WebSphere Partner Gateway ユーザー出口を経由する) バックエンド統合パッケージ化データ・バインディングを持つ要求応答フローを開発することもできます。このユーティリティ・クラスは、応答ビジネス・オブジェクトを応答ストリームに変換したり、応答ストリームを応答ビジネス・オブジェクトに変換したりするための基礎として使用できます。

## トップレベルおよび子ビジネス・オブジェクト

このセクションでは、バックエンド統合データ・バインディングが使用するトップレベル・ビジネス・オブジェクトおよび子ビジネス・オブジェクトについて説明します。

### トップレベル・ビジネス・オブジェクト:

バックエンド統合パッケージ化データ・バインディングを使用するには、以下の 3 つの子属性を持つトップレベル・ビジネス・オブジェクトを作成します。

表 30. トップレベル・ビジネス・オブジェクト

属性	タイプ
ペイロード	この属性のタイプはペイロード・コンテナ・ビジネス・オブジェクトです。任意のタイプにすることができますが、必ず 71 ページの表 31 または 72 ページの表 32 で指定されるプロパティを持つ必要があります。
添付ファイル	この属性のタイプは添付ファイル・コンテナ・ビジネス・オブジェクトです。任意のタイプにすることができますが、必ず 73 ページの表 33 で指定されるプロパティを持つ必要があります。
packagingHeaders	この属性のタイプはパッケージ化ヘッダー・ビジネス・オブジェクトです。任意のタイプにすることができますが、必ず 73 ページの表 34 で指定されるプロパティを持つ必要があります。

バックエンド統合パッケージ化メッセージをトップレベル・オブジェクトに変換すると、トップレベル・オブジェクトにメッセージからのデータが取り込まれます。トップレベル・オブジェクトをバックエンド統合パッケージ化メッセージに変換する場合は、そのトップレベル・オブジェクトがデータ・バインディングに対する入力になります。

以下の図に、トップレベル・ビジネス・オブジェクトとその子オブジェクトを示します。

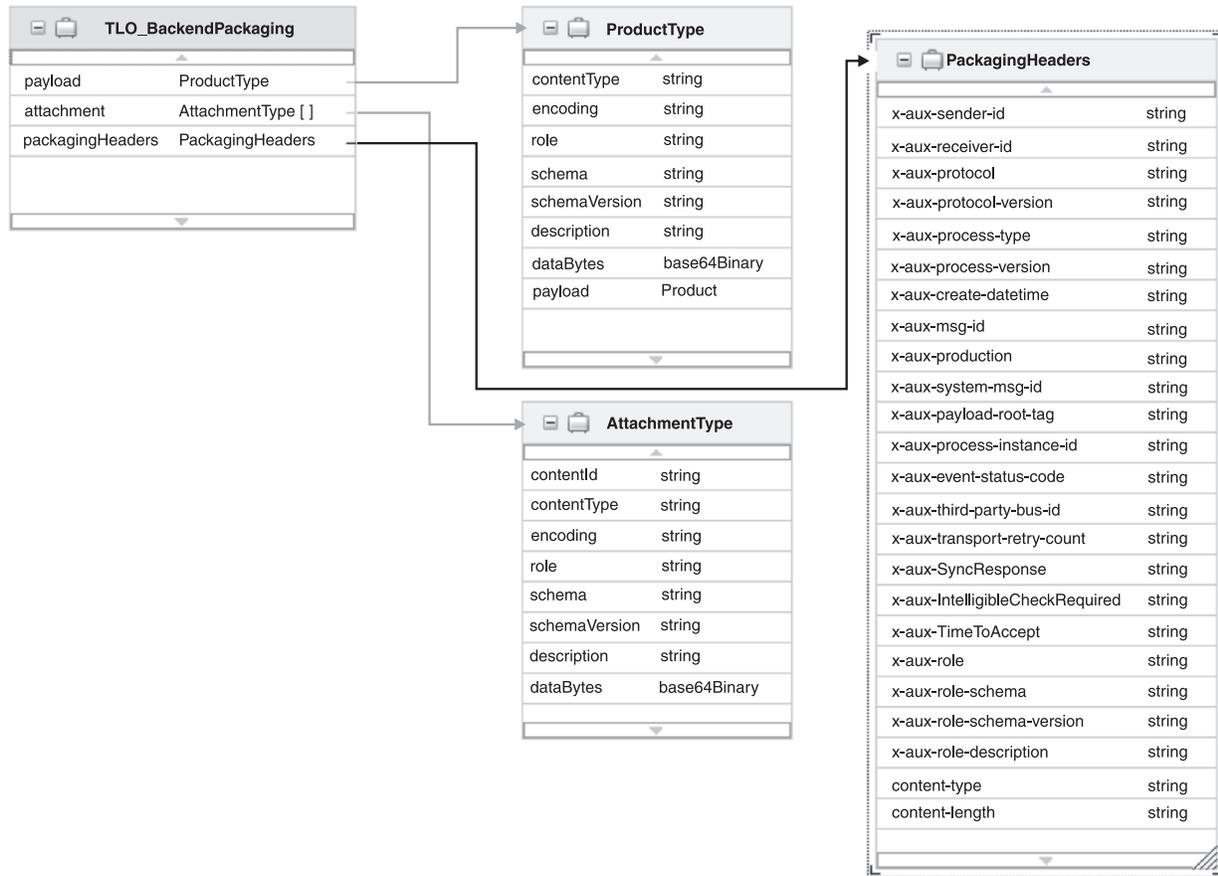


図 18. トップレベル・ビジネス・オブジェクト

**ペイロード・コンテナ・ビジネス・オブジェクト:**

ペイロード・コンテナ・ビジネス・オブジェクトの構造は、ペイロードが XML であるか非 XML であるかによって異なります。

**XML ペイロード:**

ペイロードが XML の場合、ペイロード・コンテナは以下のプロパティを持っています。

表 31. XML ペイロードの場合のペイロード・コンテナ・オブジェクト

属性	タイプ
contentType	xsd:string
エンコード	xsd:string
role	xsd:string
schema	xsd:string
schemaVersion	xsd:string
説明	xsd:string
dataBytes	xsd:base64Binary

表 31. XML ペイロードの場合のペイロード・コンテナ・オブジェクト (続き)

属性	タイプ
ペイロード	ペイロード・ビジネス・オブジェクトに対する参照。例えばペイロード自体が Pip3A4PurchaseOrderRequest オブジェクトに含まれる場合、ペイロード属性のタイプは PIP3A4PurchaseOrderRequest です。

PIP に対するビジネス・オブジェクトを作成する方法については、74 ページの『RosettaNet サービス・コンテンツ用ビジネス・オブジェクトの作成』で説明します。

#### 非 XML ペイロード:

ペイロードが XML でない場合、ペイロード・コンテナは以下のプロパティを持ちます。

表 32. 非 XML ペイロードの場合のペイロード・コンテナ・オブジェクト

属性	タイプ
contentType	xsd:string
エンコード	xsd:string
role	xsd:string
schema	xsd:string
schemaVersion	xsd:string
説明	xsd:string
dataBytes	xsd:base64Binary
dataString	xsd:string

ペイロードは dataBytes または dataString の値です。

- ペイロードが XML でない場合は、バックエンド統合パッケージ化メッセージをトップレベル・オブジェクトに変換するときに、バックエンド統合パッケージ化データ・バインディングはメッセージの <payload> エレメントの値をペイロード・ビジネス・オブジェクトに変換しません。代わりに、ペイロード・コンテナ・ビジネス・オブジェクトの dataBytes (または dataString) 属性の値をペイロードのバイト (またはストリング) 値として設定します。
- 同様に、トップレベル・オブジェクトからバックエンド統合パッケージ化メッセージを構成するときに、バックエンド統合パッケージ化データ・バインディングは、ペイロード・コンテナ・ビジネス・オブジェクトの dataBytes (または dataString) 属性の内容をバックエンド統合パッケージ化メッセージの <payload> エレメントの値として設定します。

属性 Role、Schema、SchemaVersion、および Description は、wbipackaging 1.2 で追加されました。これは ebXML フローでのみ使用されます。ペイロードのこれらの属性の値は、そのペイロードの ebXML Soap メッセージ内の対応する属性の値として設定されます。

#### 添付ファイル・コンテナ・ビジネス・オブジェクト:

このビジネス・オブジェクトは添付ファイルを表します。任意の複合タイプにすることができますが、必ず以下の属性を持つ必要があります。

表 33. 添付ファイル・コンテナ・ビジネス・オブジェクトの属性

属性	タイプ
contentID	xsd:string
contentType	xsd:string
エンコード	xsd:string
role	xsd:string
schema	xsd:string
schemaVersion	xsd:string
説明	xsd:string
dataBytes	xsd:base64Binary
dataString	xsd:string

添付ファイルは dataBytes または dataString の値です。

#### パッケージ化ヘッダー・ビジネス・オブジェクト:

このビジネス・オブジェクトはトランスポート・レベルのヘッダーを保持します。x-aux トランスポート・ヘッダーについては、29 ページの『トランスポート・レベル・ヘッダーの内容』で説明します。このビジネス・オブジェクトには以下の属性が必要です。

表 34. パッケージ化ヘッダー・ビジネス・オブジェクトの属性

属性	タイプ
x-aux-sender-id	string
x-aux-receiver-id	string
x-aux-protocol	string
x-aux-protocol-version	string
x-aux-process-type	string
x-aux-process-version	string
x-aux-create-datetime	string
x-aux-msg-id	string
x-aux-production	string
x-aux-system-msg-id	string
x-aux-payload-root-tag	string
x-aux-process-instance-id	string
x-aux-event-status-code	string
x-aux-third-party-bus-id	string
x-aux-transport-retry-count	string
x-aux-SyncResponse	string
x-aux-IntelligibleCheckRequired	string
x-aux-TimeToAccept	string
x-aux-role	string

表 34. パッケージ化ヘッダー・ビジネス・オブジェクトの属性 (続き)

属性	タイプ
x-aux-role-schema	string
x-aux-role-schema-version	string
x-aux-description	string
content-type	string
content-length	string

**RosettaNet サービス・コンテンツ用ビジネス・オブジェクトの作成:** バックエンド統合パッケージ化データ・バインディングはトップレベル・オブジェクトを必要とします。トップレベル・オブジェクトの作成については、70 ページの『トップレベル・ビジネス・オブジェクト』を参照してください。WebSphere Partner Gateway は、RosettaNet サービス・コンテンツ XML をバックエンド統合パッケージ化メッセージのペイロードとして受け入れます。ペイロード・コンテナ・オブジェクトを作成するには、RosettaNet サービス・コンテンツ XML を表すペイロード・ビジネス・オブジェクトが必要です。RosettaNet サービス・コンテンツ・ビジネス・オブジェクトは、以下のいずれかの方法を使用して作成できます。

- RosettaNet によって提供される PIP サービス・コンテンツの構造が XML スキーマ形式である場合は、その XML スキーマをペイロード・ビジネス・オブジェクトとして使用できます。
- RosettaNet によって提供される PIP サービス・コンテンツの構造が DTD 形式である場合は、その DTD を XML スキーマに変換する必要があります。PIP サービス・コンテンツ DTD を XML スキーマに変換するには、「*WebSphere Partner Gateway* ハブ構成ガイド」の手順に従ってください。この XML スキーマは、ペイロード・ビジネス・オブジェクトとして使用できます。
- PIP が DTD 形式であり、PIP に WebSphere Partner Gateway 提供の PIP パッケージが存在する場合は、代わりに以下の手順に従って PIP サービス・コンテンツ・ビジネス・オブジェクトを作成してください。
  1. 対象とする RNIF バージョンの RosettaNet サービス・コンテンツ PIP パッケージにナビゲートします。例えば、PIP3A4PurchaseOrderRequest ビジネス・オブジェクトを作成する場合は、WebSphere Partner Gateway 製品ディレクトリ内の BCG\_Package\_RNSC1.0\_RNIFV02.02.zip ファイルを使用できます。
  2. WebSphere Integration Developer を使用して、PIP 用にセットアップした WebSphere Process Server モジュールに ZIP ファイルをインポートします。
  3. Data Types フォルダを展開し、サービス・コンテンツのルート・エレメントに対応するビジネス・オブジェクトを右クリックします。例えば、PIP 3A4 要求の場合は Data Types フォルダで Pip3A4PurchaseOrder を右クリックし、「**テキスト・エディターで開く (Open with the Text Editor)**」を選択します。
  4. テキスト・エディターで、includes を以下のように変更します。
 

```
../../../../common/
```

 変更後:
 

```
./
```
  5. 変更内容を保管してテキスト・エディターを閉じます。

## バックエンド統合パッケージ化データ・バインディングの仕組み

このセクションでは、バックエンド統合データ・バインディングがビジネス・オブジェクトからバックエンド統合パッケージ化メッセージを作成したり、着信メッセージからビジネス・オブジェクトを作成したりする方法について説明します。

**バックエンド統合パッケージ化メッセージからトップレベル・オブジェクトへの変換:** このセクションでは、WebSphere Process Server 上のサービスが使用できるビジネス・オブジェクトにバックエンド統合パッケージ化メッセージを変換する方法について説明します。

70 ページの『トップレベルおよび子ビジネス・オブジェクト』で説明したように、バックエンド統合パッケージ化データ・バインディングは特定のタイプのトップレベル・ビジネス・オブジェクトを処理します。このビジネス・オブジェクトを作成しなければ、データ・バインディングを使用することはできません。

トップレベル・ビジネス・オブジェクトについては 70 ページの『トップレベル・ビジネス・オブジェクト』を参照してください。

**メッセージのタイプ:** WebSphere Partner Gateway からのバックエンド統合パッケージ化メッセージを変換する方法は、メッセージのタイプによって異なります。メッセージには以下の 3 つのタイプがあります。

- ルート・タグに <transport envelope> を持ち、URI が「http://www.ibm.com/websphere/bcg/2003/v1.1/wbipackaging」または「http://www.ibm.com/websphere/bcg/2003/v1.0/wbipackaging」である XML メッセージ。

トランスポート・エンベロープ内のメッセージの例を以下に示します。

```
<transport-envelope
  xmlns="http://www.ibm.com/websphere/bcg/2003/v1.1/wbipackaging">
  <payload encoding="base64" contentType="application/xml"
  contentId="111111111111">
    ...base64 encoded XML message...
  </payload>
  <attachment encoding="base64" contentType="text/xml"
  contentId="222222222222">
    ...base64 encoded XML attachment...
  </attachment>
  <attachment encoding="base64" contentType="application/pdf"
  contentId="333333333333">
    ...base64 encoded PDF attachment...
  </attachment>
</transport-envelope>
```

- その他の XML メッセージ
- 非 XML メッセージ

**メッセージの変換に使用するメソッド:** バックエンド統合パッケージ化 JMS メッセージをトップレベル・オブジェクトに変換するには、BCGBackEndIntegrationJMSDataBindingImpl の read メソッドを使用します。

バックエンド統合パッケージ化メッセージをトップレベル・オブジェクトに変換するために、BCGBackEndIntegrationDataBindingUtil の以下のメソッドを使用できます。

- read():

バックエンド統合パッケージ化メッセージが入カストリームに含まれる場合は、このメソッドを使用できます。

- `setFromArray`:

バックエンド統合パッケージ化メッセージがバイト配列に含まれる場合は、このメソッドを使用できます。

- `setFromString`

バックエンド統合パッケージ化がストリングに含まれる場合は、このメソッドを使用できます。

これらのメソッドを呼び出す前に、以下のメソッドを呼び出すことによってデータ・バインディングの処理をカスタマイズできます。

- `setTLOTypeName()`

オブジェクトに対して `setTLOTypeName` を使用すると、命名と URI の指定が可能です。

- `setBOPrefix()`

`setBOPrefix` を使用すると、ビジネス・オブジェクト・プレフィックスを指定できます。これを行わない場合は、デフォルトの `TLO_` プレフィックスが使用されません。

- ペイロード・ビジネス・オブジェクトのタイプを判別できる場合、トップレベル・オブジェクトは `<BOPrefix><PayloadBOTypeName>` です。
- ペイロード・ビジネス・オブジェクトを判別できない場合 (非 XML ペイロードの場合)、名前は `BCG_TLO_BackendPackaging` (デフォルト) です。

`setTLOTypeName` と `setBOPrefix` の両方を指定した場合は、`setBOPrefix` が無視されます。

非 XML ペイロードの場合、`read` メソッドを呼び出す前に `setTLOTypeName` を呼び出してトップレベル・オブジェクト名を指定しなければ、デフォルトのトップレベル・オブジェクトが使用されます。ペイロードはペイロード・ビジネス・オブジェクトに変換されません。代わりに、データ・バイトがそのままデフォルト・トップレベル・オブジェクトに設定されます。デフォルト・トップレベル・オブジェクトを 77 ページの図 19に示します。

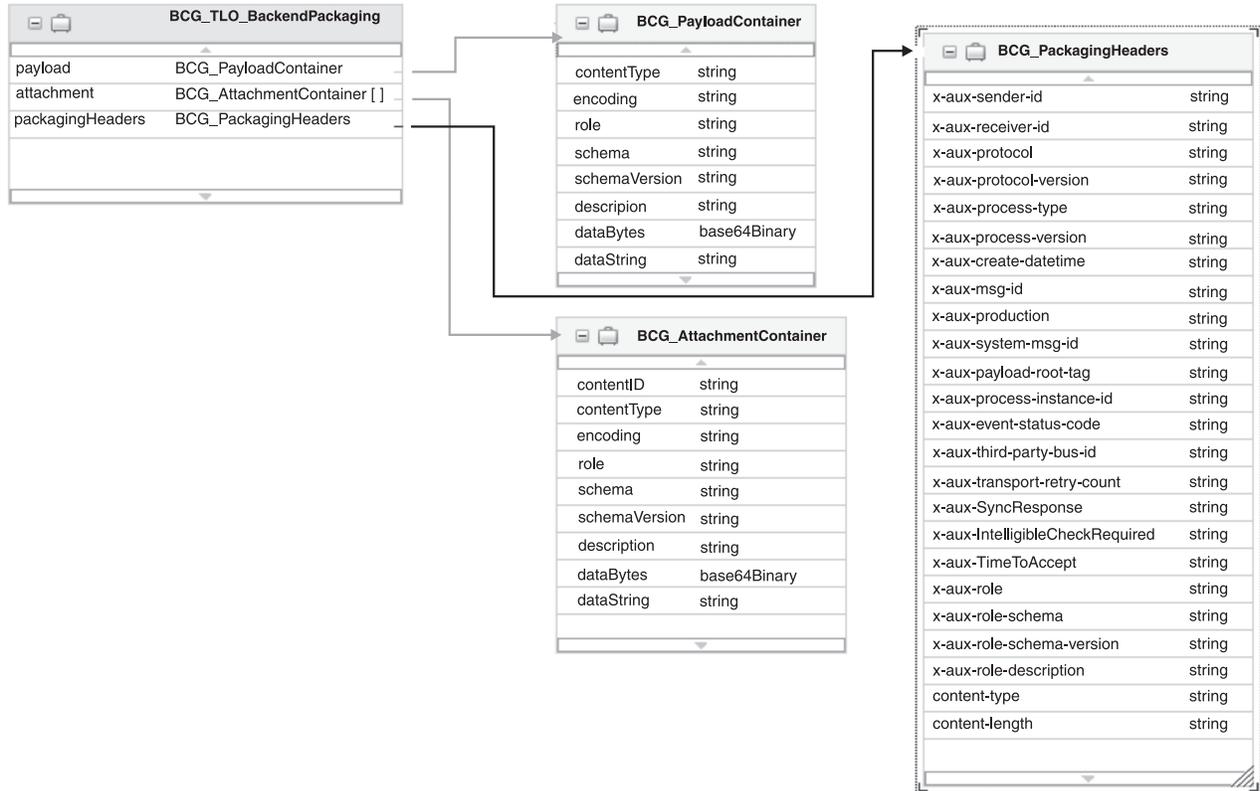


図 19. デフォルト・トップレベル・ビジネス・オブジェクト

メッセージがトランスポート・ヘッダーを持つ場合は、それらのトランスポート・ヘッダーを読み取り、setxAuxHeaders() を使用して設定する必要があります。setxAuxHeaders メソッドは、必ず read() を呼び出す前に呼び出してください。JMS トランスポート・ヘッダーは read メソッドで指定された JMS メッセージから読み取られるため、setxAuxHeaders メソッドは BCGBackEndIntegrationJMSDataBindingImpl データ・バインディングには適用されません。

**メッセージの変換方法:** 以下のセクションでは、

BCGBackEndIntegrationDataBindingUtil の read メソッド、setFromByteArray メソッド、および setFromString メソッドと、BCGBackEndIntegrationJMSDataBindingImpl の read メソッドの動作の仕組みについて説明します。これらのメソッドは、

1. Content-type ヘッダーがメッセージで使用可能な場合は、そのヘッダーから XML コンテンツ・タイプを判別します。例を次に示します。

Content-Type: application/xml

contentType が使用可能でない場合、メソッドは、メッセージの先頭数バイトを検査してタイプを判別します。

メソッドは、以下のようにコンテンツ・タイプに基づいてペイロードを取得します。

- XML メッセージの場合は、メッセージを非直列化してペイロード・ビジネス・オブジェクトを取得します。

- ルート・タグが <transport envelope> である XML メッセージの場合は、ペイロードを Base64 デコードしてペイロード・バイトを取得し、メッセージを解析して XML であるか非 XML であるかを判別します。
  - メッセージが XML である場合は、ペイロード・バイトを非直列化してペイロード・ビジネス・オブジェクトを取得します。
  - メッセージが XML でない場合は、ペイロード・バイトを非直列化しません。ペイロード・バイトはそのまま使用されます。

トランスポート・エンベロープ内に含まれる添付ファイルごとに、添付ファイルを Base64 デコードして添付ファイル・バイトを取得します。添付ファイル・バイトはそのまま使用され、非直列化は行われません。

- ルート・タグが <EventNotification> である XML メッセージの場合は、メッセージを非直列化してイベント通知ビジネス・オブジェクトを取得します。
- XML でないメッセージの場合は、メッセージを非直列化しません。バイトはそのまま使用されます。

## 2. トップレベル・ビジネス・オブジェクトの名前を判別します。

- ペイロードが XML である場合、
  - read、setFromByteArray、または setFromString が呼び出される前に setTLOTypeName が呼び出された場合は、そのメソッドで指定された名前を直接使用してトップレベル名を作成します。
  - setBOPrefix を使用した場合、トップレベル・オブジェクト名は、指定のプレフィックスにペイロードの XML ルート・エレメントを付加して構成されます。URI もペイロードから取得されます。
  - setTLOTypeName も setBOPrefix も使用しない場合、トップレベル・オブジェクト名は、デフォルト・プレフィックス TLO\_ にペイロードの XML ルート・エレメントを付加して構成されます。
- ペイロードが XML でない場合、
  - デフォルト・トップレベル・オブジェクトを使用できます。
  - setTLOTypeName メソッドを使用できますが、作成したトップレベル・オブジェクトまたはデフォルト・トップレベル・オブジェクト (BCG\_TLO\_BackendPackaging) に適合する必要があります。

## 3. トップレベル・オブジェクトをインスタンス化します。

## 4. ペイロード・コンテナ・ビジネス・オブジェクトをインスタンス化し、値を設定します。値の設定方法は、メッセージのタイプによって異なります。

- ルート・タグが <transport envelope> である XML メッセージの場合は、以下のように属性を設定します。
  - contentType
    - <payload>タグの contentType 属性の値。
  - encoding
    - <payload>タグの encoding 属性の値。
  - payload
    - ペイロード・ビジネス・オブジェクトの名前。

- XML メッセージまたはルート・タグが <event notification> である XML メッセージの場合は、以下のように属性を設定します。
    - contentType は設定しません
    - encoding は設定しません
    - payload
      - ペイロード・ビジネス・オブジェクトの名前。
  - 非 XML メッセージの場合は、以下のように属性を設定します。
    - contentType は設定しません
    - encoding は設定しません
    - dataBytes
      - 着信メッセージをバイトとして受信した場合は、バイトのセット全体をこの属性の値として設定します。
    - dataString
      - 着信メッセージをストリングとして受信した場合は、ストリング全体をこの属性の値として設定します。
    - payload は設定しません。
5. 添付ファイル・ビジネス・オブジェクトをインスタンス化し、以下のように値を設定します。
- contentType
    - <attachment>タグの contentType 属性の値。
  - encoding
    - <attachment>タグの encoding 属性の値。
  - contentId
    - <attachment>タグの contentId 属性の値。
  - dataBytes
    - Base64 デコードされた添付ファイル・バイト。
6. トランスポート・ヘッダー・ビジネス・オブジェクトをインスタンス化し、メッセージから読み取ったトランスポート・ヘッダーを設定します。このビジネス・オブジェクトの名前は、トップレベル・オブジェクトの packagingHeaders 属性の値として設定されます。

87 ページの『コード例』に、メソッドの使用方法を示します。

**トップレベル・オブジェクトからバックエンド統合パッケージ化メッセージへの変換:** このセクションでは、WebSphere Process Server からのビジネス・オブジェクトをバックエンド統合パッケージ化メッセージに直列化する方法について説明します。データ・バインディングは、内容およびすべての添付ファイルを Base64 形式でトランスポート・エンベロープに格納します。また、トップレベル・オブジェクトのトランスポート・ヘッダーをストリングとしてエンベロープに格納します。

バックエンド統合パッケージ化データ・バインディングを呼び出す前に、サービスがトップレベル・オブジェクトを作成する必要があります (詳しくは 70 ページの『トップレベル・ビジネス・オブジェクト』を参照してください)。

**オブジェクトの変換に使用するメソッド:** WebSphere Process Server サービスは、バックエンド統合パッケージ化データ・バインディングを呼び出してトップレベル・オブジェクトを送信します。オブジェクトをバックエンド統合パッケージ化 JMS メッセージに変換するには、BCGBackendIntegrationJMSDataBindingImpl の write メソッドを使用します。オブジェクトをバックエンド統合パッケージ化メッセージに変換するには、BCGBackendIntegrationDataBindingUtil の write メソッド、getAsByteArray メソッド、または getAsString メソッドを使用します。

これらのいずれかのメソッドを呼び出す前に、以下のものを呼び出すことができます。

- 86 ページの『 setPackagingSchema メソッド』

このメソッドは、バックエンド統合パッケージ化メッセージの構成に使用するパッケージ化スキーマを指定します。

- 85 ページの『 setOptions メソッド』

このメソッドは、メッセージを直列化するために使用できる情報 (ペイロードのルート・タグなど) を提供します。設定可能なオプションについては、85 ページの『 setOptions メソッド』を参照してください。

**オブジェクトの変換方法:** 以下のセクションでは、BCGBackendIntegrationDataBindingUtil の write メソッド、getAsByteArray メソッド、および getAsString メソッドと、BCGBackendIntegrationJMSDataBindingImpl の write メソッドの動作の仕組みについて説明します。これらのメソッドは、

1. ペイロードのコンテンツ・タイプを判別します。この情報は、ペイロード・コンテナ・ビジネス・オブジェクトから取得します。ペイロードの処理方法は、コンテンツ・タイプに応じて以下のように異なります。

- XML ペイロード

ペイロード・コンテナが存在する場合、その dataBytes 属性または dataString 属性はヌルでなければならず、ペイロード・コンテナ・ビジネス・オブジェクトに少なくとも 1 つの複合タイプの属性が必要です。これは、ペイロード・ビジネス・オブジェクトに対する参照です。ビジネス・オブジェクトは直列化され、XML ペイロードとして使用されます。複数の複合タイプの属性が設定されている場合は、最初の複合タイプ属性が XML ペイロードであると見なされます。

- EventNotification

最初のヌル以外の属性のタイプは EventNotification でなければなりません。これは、イベント通知ビジネス・オブジェクトに対する参照です。ビジネス・オブジェクトは直列化され、XML ペイロードとして使用されます。

- 非 XML

dataBytes 属性がヌル以外の場合は、その属性の値をペイロードとして使用します。

dataString 属性がヌル以外の場合は、その属性の値をペイロードとして使用します。

2. タイプが XML または EventNotification の場合は、ペイロード・ビジネス・オブジェクトを直列化します。
3. 以下のように、タイプに基づいてペイロードを Base64 エンコードします。

ペイロード・コンテナ・ビジネス・オブジェクトの dataByte 属性の値がヌル以外の場合は、その値 (ペイロード・バイト) を Base64 エンコードします。

ペイロード・コンテナ・ビジネス・オブジェクトの dataString 属性の値がヌル以外の場合は、ビジネス・オブジェクトの contentType 属性に基づいてバイトを抽出します。そのバイトを Base64 エンコードします。

ペイロードがビジネス・オブジェクトから直列化された場合は、ビジネス・オブジェクトの contentType 属性に基づいてバイトを抽出します。そのバイトを Base64 エンコードします。

4. 指定されたパッケージ化スキーマに基づいて XML 文書を構成します。
  - a. <transport envelope> ルート・タグを追加します。
  - b. <payload> タグ (<transport envelope> タグの子エレメント) に、ステップ 3 で base64 エンコードしたストリングを設定します。
5. 添付ファイル进行处理します。

添付ファイル・コンテナ・オブジェクトの dataByte 属性がヌル以外の場合は、バイトを base64 エンコードし、そのストリングを <attachment> タグの値として設定します。

添付ファイル・コンテナ・オブジェクトの dataString 属性がヌル以外の場合は、ビジネス・オブジェクトの contentType 属性に基づいてバイトを抽出します。次にそのバイトを base64 エンコードし、そのストリングを <attachment> タグの値として設定します。

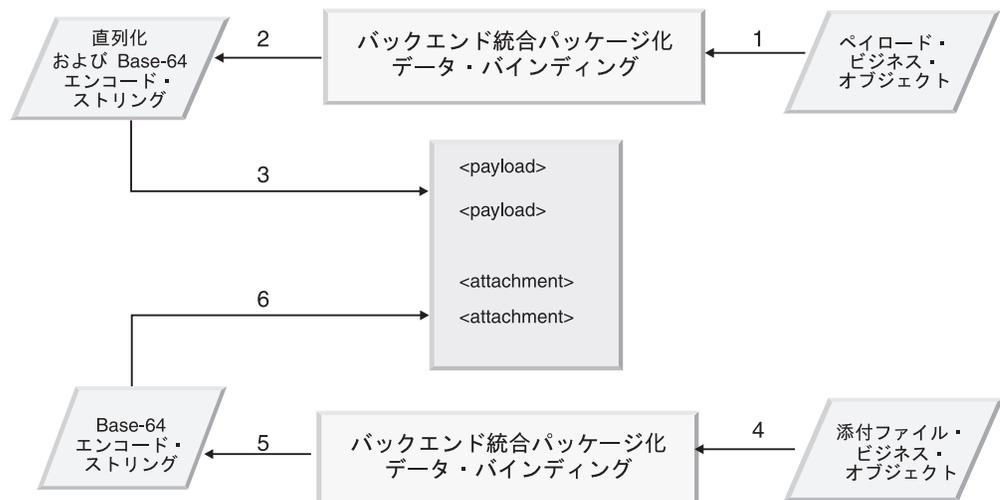


図 20. データ・バインディングが変換後のビジネス・オブジェクトをバックエンド統合パッケージ化メッセージに追加する方法

87 ページの『コード例』に、メソッドの使用方法を示します。

## BCGBackEndIntegrationDataBindingUtil クラス

このセクションでは、BCGBackEndIntegrationDataBindingUtil クラスおよびこのクラスのメソッドについて説明します。このクラスはそのまま使用できます。また、データ・バインディングの処理をカスタマイズする場合は、新しいデータ・バインディング・クラスを作成することもできます。

### DataBinding インターフェース

BCGBackEndIntegrationDataBindingUtil は以下の DataBinding インターフェースを実装します。

```
DataBinding
public interface commonj.connector.runtime.DataBinding extends Serializable {
    public DataObject getDataObject() throws
commonj.connector.runtime.DataBindingException;
    public void setDataObject(DataObject dataObject) throws
commonj.connector.runtime.DataBindingException;
}
```

### メソッド

BCGBackEndIntegrationDataBindingUtil メソッドについては、このセクションで説明します。

BCGBackEndIntegrationDataBindingUtil クラスには、以下の 2 種類のメソッドがあります。

- バイト配列、ストリング、またはデータ・ストリームからデータ・オブジェクトを作成するもの。
  - read(InputStream)
  - setFromArray(byte[ ])
  - setFromString(String)
  - setAuxHeaders(HashMap)
  - setBOPrefix(String)
  - setTLOType(String, String)
  - getDataObject()

#### 注記:

1. 以下のいずれかのメソッドを使用すると、データ・オブジェクトを作成できます。

- read(InputStream)
- setFromArray(byte[ ])
- setFromString(String)

データ・オブジェクトを作成するメソッドを呼び出す前に、プログラムは以下の任意のメソッドを呼び出すことができます。

- setAuxHeaders
- setBOPrefix
- setTLOType

- 2. `setTLOType` メソッドを使用する場合、`setBOPrefix` メソッドは意味を持たず、無視されます。
- 3. `getDataObject` メソッドは、トップレベル・ビジネス・オブジェクトを取得するために使用します。
- データ・オブジェクトをバイト、ストリング、またはデータ・ストリームに変換するもの。
  - `setDataObject(DataObject)`
  - `setPackagingSchema(String)`
  - `setOptions(HashMap)`
  - `getAsString()`
  - `getAsByteArray()`
  - `write(OutputStream)`
  - `getxAuxHeaders()`

**getAsByteArray** メソッド: このメソッドは、トップレベル・オブジェクトから取得した `byte[ ]` を戻します。

**構文:**

```
public byte[ ] getAsByteArray()
```

**getAsString** メソッド: このメソッドは、ストリング形式のトップレベル・オブジェクトを戻します。

**構文:**

```
protected java.lang.String getAsString()
```

**getDataObject** メソッド: このメソッドは、データ・オブジェクトが既にインスタンス化されている場合に、そのデータ・オブジェクトを戻します。

**構文:**

```
public commonj.sdo.DataObject getDataObject()
```

**getxAuxHeader** メソッド: このメソッドは、トランスポート・エンベロープの `x-aux` ヘッダーを戻します。`x-aux` ヘッダーについては、29 ページの『トランスポート・レベル・ヘッダーの内容』を参照してください。

**構文:**

```
public java.util.HashMap getxAuxHeaders()
```

**read** メソッド: このメソッドは、入力ストリームを受け取り、そのストリームを読み取ってトップレベル・ビジネス・オブジェクトに変換します。`read` メソッドによって実行される操作については、75 ページの『バックエンド統合パッケージ化メッセージからトップレベル・オブジェクトへの変換』を参照してください。

**構文:**

```
public void read(javax.jms.Message jmsMessage)
    throws javax.jms.JMSException
```

**パラメーター:**

## **jmsMessage**

ペイロードおよびトランスポート・ヘッダーを含むメッセージ。

**setBOPrefix** メソッド: トップレベル・オブジェクトが指定されなかった場合は、`setBOPrefix ()` を呼び出してプレフィックスを指定できます。プレフィックスを指定しない場合は、デフォルト・プレフィックスとして `TLO_` が使用されます。

### **注記:**

1. XML ペイロードの場合は、XML ペイロードのネーム・スペースがトップレベル・オブジェクトのネーム・スペースとして使用されます。
2. 非 XML ペイロードの場合は、デフォルトのネーム・スペース「`http://ibm.com/websphere/bcg/2005/wbi/bo`」がトップレベル・オブジェクトのネーム・スペースとして使用されます。

### **構文:**

```
public void setBOPrefix(java.lang.String prefix)
```

**setDataObject** メソッド: このメソッドはデータ・オブジェクトを設定します。

### **構文:**

```
public void setDataObject(commonj.sdo.DataObject topLevelbo)
    throws commonj.connector.runtime.DataBindingException
```

### **パラメーター:**

#### **topLevelbo**

トップレベル・ビジネス・オブジェクト。70 ページの『トップレベル・ビジネス・オブジェクト』を参照してください。

**setDebugLevel** メソッド: デフォルトでは、デバッグ・レベルはエラーに設定されます。レベルを変更するには、このメソッドを使用します。

### **構文:**

```
public static void setDebugLevel(int debugLevel)
```

### **パラメーター:**

#### **debugLevel**

デバッグ・レベルに指定可能な値は以下のとおりです。

- `BCG_LOG_DEBUG`
- `BCG_LOG_ERROR`
- `BCG_LOG_WARNING`
- `BCG_LOG_INFO`

**setFromArray** メソッド: このメソッドは、ペイロード・データ・バイトまたはトランスポート・エンベロープ・データ・バイトを受け取ってトップレベル・オブジェクトを作成します。このメソッドを呼び出す前に、以下の 1 つ以上のメソッドを呼び出すことができます。

- 86 ページの『 `setTLOTypeName` メソッド』
- 『 `setBOPrefix` メソッド』
- 87 ページの『 `setxAuxHeader` メソッド』

### 構文:

```
public void setFromByteArray(byte[ ] dataBytes)
    throws commonj.connector.runtime.DataBindingException
```

### パラメーター:

#### dataBytes

バイトの形式のペイロード。

**setFromString メソッド:** このメソッドは、パラメーターとして文字列を受け取り、トップレベル・オブジェクトに変換します。このメソッドを呼び出す前に、以下の 1 つ以上のメソッドを呼び出すことができます。

- 86 ページの『 setTLOTypeName メソッド』
- 84 ページの『 setBOPrefix メソッド』
- 『 setOptions メソッド』

### 構文:

```
public void setFromString(java.lang.String fromString)
    throws commonj.connector.runtime.DataBindingException
```

### パラメーター:

#### fromString

文字列形式のトップレベル・データ・オブジェクトまたはペイロード。

**setOptions メソッド:** このメソッドは、必要なオプションを設定します。

### 構文:

```
public void setOptions(java.util.HashMap options)
    throws commonj.connector.runtime.DataBindingException
```

### パラメーター:

#### HashMap

HashMap は以下のキー値を持つことができます。

- BCG\_ROOT\_NODE\_NAME

これは、ペイロードの生成に使用するルート名です。

- BCG\_APPEND\_DTD

この属性は、ペイロードに DTD タグを付加するかどうかを示します。値は BCG\_APPEND\_DTD\_TRUE または BCG\_APPEND\_DTD\_FALSE にすることができます。デフォルトは BCG\_APPEND\_DTD\_FALSE です。

- BCG\_SYSTEM\_ID

これは DTD のシステム ID です。BCG\_APPEND\_DTD の値が BCG\_APPEND\_DTD\_TRUE の場合は、この値を設定できます。

BCG\_APPEND\_DTD の値が BCG\_APPEND\_DTD\_FALSE の場合は、この値は無視されます。

- BCG\_PUBLIC\_ID

これは DTD の公開 ID です。BCG\_APPEND\_DTD の値が BCG\_APPEND\_DTD\_TRUE の場合は、この値はオプションです。BCG\_APPEND\_DTD の値が BCG\_APPEND\_DTD\_FALSE の場合は、このオプションは無視されます。

**setPackagingSchema メソッド:** このメソッドは、バックエンド統合パッケージ化を生成するときに使用するパッケージ化スキーマを定義します。以下のいずれかのパッケージ化スキーマを指定できます。

- <http://www.ibm.com/websphere/bcg/2003/v1.0/wbipackaging>
- <http://www.ibm.com/websphere/bcg/2003/v1.1/wbipackaging>
- <http://www.ibm.com/websphere/bcg/2004/v1.2/wbipackaging>

パッケージ化スキーマが指定されない場合は、<http://www.ibm.com/websphere/bcg/2003/v1.1/wbipackaging> が使用されます。

パッケージ化スキーマは、write メソッドを呼び出すときに使用されます。

#### 構文:

```
public void setPackagingSchema(java.lang.String packagingSchema)
```

#### パラメーター:

##### packagingSchema

トップレベル・ビジネス・オブジェクトのスキーマ名。

**setTLOTypeName メソッド:** このメソッドは、トップレベル・オブジェクト名とそのネーム・スペースを設定します。

- ペイロードが XML でないときに setTLOTypeName を使用しない場合は、デフォルト・トップレベル・オブジェクト名が使用されます。
- ペイロードが XML 文書である場合、トップレベル・オブジェクト名は以下のように生成されます。

`boPrefix + < RootTag of the XML Payload >`.

例えば、setBOPrefix("IBM") を呼び出し、ペイロードのルート・エレメントが「ABC」である場合のトップレベル名は「IBM\_ABC」です。

プレフィックスは、84 ページの『 setBOPrefix メソッド』を使用して設定できます。トップレベル・オブジェクト名が設定されておらず、ペイロードが XML 文書である場合は、タイプが `boPrefix + < RootTag >` の DataType を作成する必要があります。

#### 構文:

```
public void setTLOTypeName(java.lang.String tns,  
                           java.lang.String typeName)
```

#### パラメーター:

##### tns

トップレベル・ビジネス・オブジェクトのネーム・スペース。

##### typeName

トップレベル・オブジェクトのタイプ。

**setxAuxHeader メソッド:** このメソッドはトランスポート・ヘッダーを設定します。トランスポート・ヘッダーを設定しなければ、トランスポート・ヘッダーは読み取られません。x-aux ヘッダーについて詳しくは、29 ページの『トランスポート・レベル・ヘッダーの内容』を参照してください。

**構文:**

```
public void setxAuxHeaders(java.util.HashMap xAuxHeaders)
```

**write メソッド:** このメソッドは、トップレベル・オブジェクトを指定の出カストリームに書き込みます。write メソッドによって実行される操作について詳しくは、79 ページの『トップレベル・オブジェクトからバックエンド統合パッケージ化メッセージへの変換』を参照してください。

**構文:**

```
public void write(javax.jms.Message jmsMessage)
    throws javax.jms.JMSEException
```

**パラメーター:**

**outputStream**

データ・オブジェクトの書き込み先出カストリーム。

## コード例

以下のコードに、入カストリームからトップレベル・オブジェクトを入力する方法を示します。

```
BCGBackEndIntegrationDataBindingUtil util = BCGBackEndIntegrationDataBindingUtil ();
util.setTLOTypeName ("TLO_URIName","TLOName");
//util.setBOPrefix ("BO_prefix");
//setBoPrefix is commented because setTLOTypeName () is being used
util.read (inputStream);
DataObject tlo = util.getDataObject ();
```

以下のコードに、データ・オブジェクトからストリームを取得する方法を示します。

```
BCGBackEndIntegrationDataBindingUtil util = BCGBackEndIntegrationDataBindingUtil ();
util.setOptions (options);
util.setDataObject (tlo);
byte [ ] tlo_bytes = util.getAsByteArray ();
```

## BCGBackEndIntegrationJMSDataBindingImpl クラス

このクラスは、ペイロードを含む JMS メッセージからビジネス・オブジェクトを作成したり、ビジネス・オブジェクトを JMS メッセージに書き込んだりします。JMS メッセージからビジネス・オブジェクトを読み取る際には、JMS トランスポート・ヘッダーを読み取ってからビジネス・オブジェクトを作成します。ビジネス・オブジェクトを JMS メッセージに書き込むとき、ビジネス・オブジェクトがパッケージ化ヘッダー子ビジネス・オブジェクトを持つ場合は、このクラスは JMS トランスポート・ヘッダーを書き込みます。

BCGBackEndIntegrationJMSDataBindingImpl は BCGBackEndIntegrationDataBindingUtil を拡張します。後者について詳しくは、82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』を参照してください。

## JMSDataBinding インターフェース

BCGBackEndIntegrationJMSDataBindingImpl は、以下のように JMSDataBinding インターフェースを実装します。

```
public interface com.ibm.websphere.sca.jms.data.JMSDataBinding extends DataBinding {  
  
    public void read(javax.jms.Message message) throws javax.jms.JMSEException;  
    public void write(javax.jms.Message message) throws javax.jms.JMSEException;  
    public int getMessageType();  
  
    static public int OBJECT_MESSAGE = 0;  
    static public int TEXT_MESSAGE = 1;  
    static public int BYTES_MESSAGE = 2;  
    static public int STREAM_MESSAGE = 3;  
    static public int MAP_MESSAGE = 4;  
  
}
```

### メソッド

このセクションでは BCGBackEndIntegrationJMSDataBindingImpl メソッドについて説明します。

**getMessageType:** このメソッドはメッセージのタイプを戻します。

**構文:**

```
public int getMessageType()
```

**isBusinessException:** このメソッドは、ビジネス例外が発生したかどうかの標識を戻します。

**構文:**

```
public boolean isBusinessException()
```

**read:** このメソッドはトランスポート・ヘッダーを読み取り、メッセージ・タイプによってトップレベル・ビジネス・オブジェクトを作成します。read メソッドによって実行される操作について詳しくは、75 ページの『バックエンド統合パッケージ化メッセージからトップレベル・オブジェクトへの変換』を参照してください。

このメソッドでは、AuxHeader を設定したりメッセージ自体から読み取ったりする必要はありません。

**構文:**

```
public void read(javax.jms.Message jmsMessage)  
    throws javax.jms.JMSEException
```

**パラメーター:**

**jmsMessage**

ペイロードおよびトランスポート・ヘッダーを含むメッセージ。

**setBusinessException:** このメソッドは、ビジネス例外が発生したかどうかの標識を設定します。

**構文:**

```
public void setBusinessException(boolean arg0)
```

**write:** このメソッドはデータ・オブジェクトをメッセージに書き込み、メッセージ・タイプによってメッセージのヘッダーを設定します。write メソッドによって実行される操作について詳しくは、79 ページの『トップレベル・オブジェクトからバックエンド統合パッケージ化メッセージへの変換』を参照してください。

**構文:**

```
public void write(javax.jms.Message jmsMessage)
    throws javax.jms.JMSEException
```

## コンポーネント実装へのデータ・バインディング・クラスの組み込み

WebSphere Integration Developer を使用して WebSphere Process Server コンポーネントを開発するときには、WebSphere Partner Gateway 提供の BCGBackEndIntegrationJMSDataBindingImpl クラスおよび BCGBackEndIntegrationDataBindingUtil クラスを使用できます。これらのクラスは databinding.jar に格納されています。このファイルは、製品イメージの Integration¥WBI¥WebSphereProcessServer¥DataBinding ディレクトリーにあります。

ビジネス統合プロジェクトでこれらのクラスを使用するには、WebSphere Process Server にデプロイするビジネス統合プロジェクトの EAR ファイルで databinding.jar を使用可能にしてください。生成された EAR ファイルに databinding.jar ファイルを組み込むには、WebSphere Process Server および WebSphere Integration Developer のインフォメーション・センターを参照するか、このセクションで説明する手順に従ってください。この手順を行う前に、必ずビジネス統合パースペクティブに移動し、この jar ファイルをインポートするビジネス統合プロジェクトを作成しておく必要があります。

1. WebSphere Integration Developer から、databinding.jar をプロジェクトにインポートします。databinding.jar は「J2EE Utility jars」としてインポートする必要があります。正常にインポートされると、WebSphere Integration Developer が databinding.jar 用のプロジェクトを暗黙に作成します。
2. このプロジェクトを依存関係としてビジネス統合プロジェクトに追加します。プロジェクトを依存関係として追加するには、以下のようにします。

**注:** 以下に示す方法は、プロジェクトを依存関係として追加するための 1 つの例です。他の方法については、WebSphere Process Server の資料を参照してください。

- a. ビジネス統合プロジェクトをダブルクリックします。

このステップにより、依存関係エディターが開きます。

- b. 「Java」フォルダーを展開します。
- c. 「追加」ボタンをクリックします。

プロジェクトのリストが表示されます。

- d. ステップ 1 で databinding.jar に対して WebSphere Integration Developer によって暗黙に作成されたプロジェクトを選択します。

databinding.jar に対して WebSphere Integration Developer によって作成されたプロジェクトが、依存関係として「Java」の下に表示されます。

- e. そのプロジェクトを選択します。
  - f. 「モジュールとともにデプロイ (Deploy with Module)」チェック・ボックスを選択します。
3. 依存関係エディターを閉じてワークスペースを保管します。
- これらのステップを完了すると、生成された EAR ファイルで databindng.jar が使用可能になります。

---

## トランスポートとしての HTTP と WebSphere Process Server の統合

この章では、WebSphere Partner Gateway と WebSphere Process Server を HTTP トランスポート・プロトコルを通じて統合する方法について説明します。この章で説明する内容は、次のとおりです。

- 『WebSphere Process Server へのメッセージの送信方法』
- 92 ページの 『WebSphere Process Server からのメッセージの送信方法』
- 94 ページの 『WebSphere Process Server への文書の送信』
- 96 ページの 『WebSphere Process Server からの文書の送信』

この章では、HTTP を介して送信された SOAP 要求が処理される仕組みについては説明しません。HTTP を介して SOAP 要求を送受信する方法については、110 ページの 『WebSphere Process Server と SOAP/HTTP の統合』を参照してください。

### WebSphere Process Server へのメッセージの送信方法

このセクションでは、WebSphere Partner Gateway が外部パートナーからメッセージを受信し、HTTP を介してそのメッセージを WebSphere Process Server に送信する方法について説明します。

#### 注記:

1. WebSphere Partner Gateway から WebSphere Process Server に送信する場合、RosettaNet 以外の文書タイプは、いずれもパッケージ化なしにすることができます。RosettaNet 文書には、必ずバックエンド統合パッケージ化が必要です。
2. WebSphere Partner Gateway から WebSphere Process Server に送信する場合、SOAP (Web サービス要求) 以外の文書タイプは、いずれもバックエンド統合のパッケージ化を持つことができます。SOAP 要求の場合はパッケージ化なしを指定する必要があります。

HTTP を介した SOAP 文書の送受信については、110 ページの 『WebSphere Process Server と SOAP/HTTP の統合』を参照してください。

WebSphere Partner Gateway から送信されたメッセージを WebSphere Process Server が受信するためには、メッセージを取得してビジネス・オブジェクトに変換するサーブレットを作成する必要があります。

WebSphere Partner Gateway は、パートナー接続で内部パートナーの「レシーバー」HTTP 宛先で構成された URL にメッセージを送信します。サーブレットは、この URL で listen してメッセージを受信します。変換後のビジネス文書を受信するサービスは SCA エクスポート・バインディングを持ちます。外部パートナーから送信

されたメッセージが WebSphere Partner Gateway によって処理されて HTTP サブレットに送信され、さらにエクスポート・バインディングを介してビジネス・オブジェクトがサービスを呼び出す仕組みを 図 21 に示します。

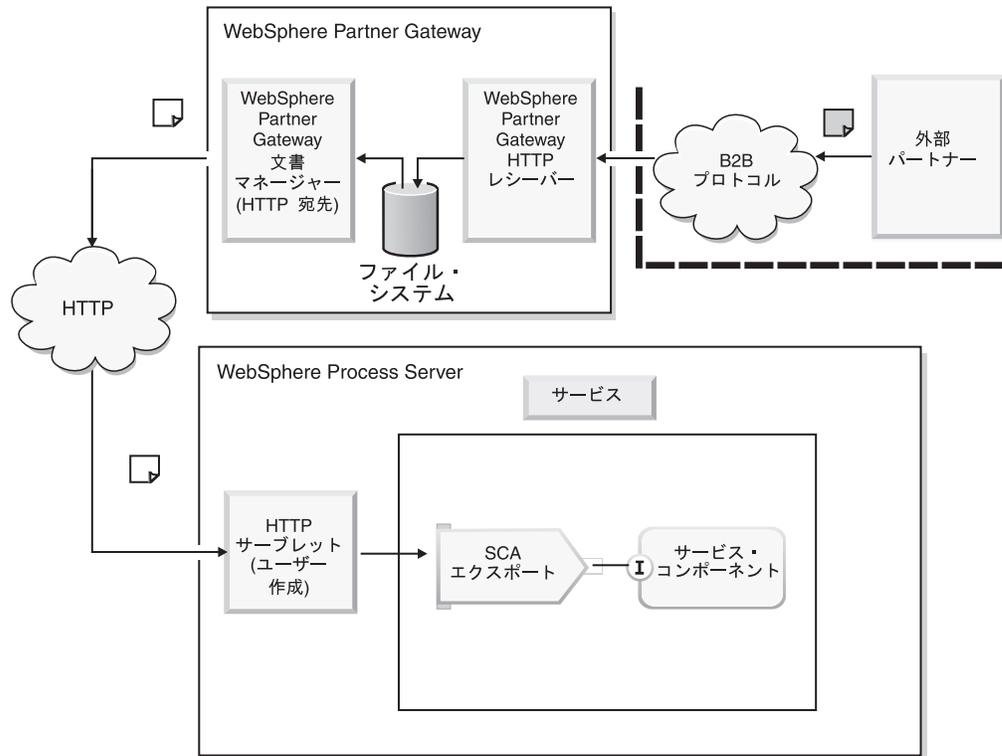


図 21. WebSphere Partner Gateway からのメッセージが HTTP サブレットに送信される仕組み

HTTP サブレットは以下のタスクを行います。

1. サブレットが、受信した要求メッセージから、起動する必要があるサービスとそのサービスのメソッドを判別します。
2. HTTP サブレットが着信メッセージからビジネス・オブジェクトを構成します。詳しくは、95 ページの『サブレットの作成』を参照してください。
3. HTTP サブレットが、以下のように適切な SCA クライアント API を使用してサービスを呼び出します。
  - a. SCA サービス・メソッドが要求専用である場合は、ビジネス応答は不要です。HTTP サブレットが SCA サービスを呼び出すことができる場合、サブレットは、HTTP 200 の状況コードを WebSphere Partner Gateway に戻します。HTTP サブレットが SCA サービスを呼び出すことができない場合は、適切な HTTP エラー状況コードを戻します。
  - b. SCA サービス・メソッドが要求応答である場合、SCA サービスは応答ビジネス・オブジェクトを戻します。HTTP サブレットは、このビジネス・オブジェクトをメッセージに直列化します。HTTP サブレットは、HTTP 応答でこのメッセージを WebSphere Partner Gateway に戻します。HTTP サブレットは、何らかの理由で SCA サービスを正常に呼び出すことができない場合には、該当する HTTP エラー状況コードを戻します。

例えば、パートナーから cXML 同期メッセージを受信した場合、WebSphere Partner Gateway は cXML メッセージを HTTP で WebSphere Process Server に送信します。WebSphere Partner Gateway は、WebSphere Process Server が同じ HTTP 接続で同期 cXML 応答を提供すると想定します。

**注:** WebSphere Partner Gateway は、バックエンド統合パッケージ化に対する要求専用呼び出しをサポートします。ただし、(WebSphere Partner Gateway ユーザー出口を経由する) バックエンド統合パッケージ化データ・バインディングを持つ要求応答フローを開発することもできます。82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』は、応答ビジネス・オブジェクトを応答ストリームに変換するための基礎として使用できます。

## WebSphere Process Server からのメッセージの送信方法

このセクションでは、WebSphere Partner Gateway が HTTP を介して WebSphere Process Server からメッセージを受信し、外部パートナーに送信する方法について説明します。

### 注記:

1. WebSphere Process Server から WebSphere Partner Gateway に送信する場合、RosettaNet およびバイナリー以外の文書タイプは、いずれもパッケージ化なしにすることができます。RosettaNet およびバイナリーの文書には、必ずバックエンド統合パッケージ化が必要です。
2. WebSphere Process Server から WebSphere Partner Gateway に送信する場合、SOAP (Web サービス要求) 以外の文書タイプは、いずれもバックエンド統合のパッケージ化にすることができます。SOAP 要求の場合はパッケージ化なしを指定する必要があります。

HTTP を介した SOAP 文書の送受信について詳しくは、110 ページの『WebSphere Process Server と SOAP/HTTP の統合』を参照してください。

WebSphere Process Server は、WebSphere Partner Gateway にメッセージを送信するときに、WebSphere Partner Gateway HTTP レシーバーに対する HTTP POST を実行するコンポーネントを使用して HTTP POST を WebSphere Partner Gateway HTTP レシーバーに送信します。そのコンポーネントを開発し、サービスとして公開してください。

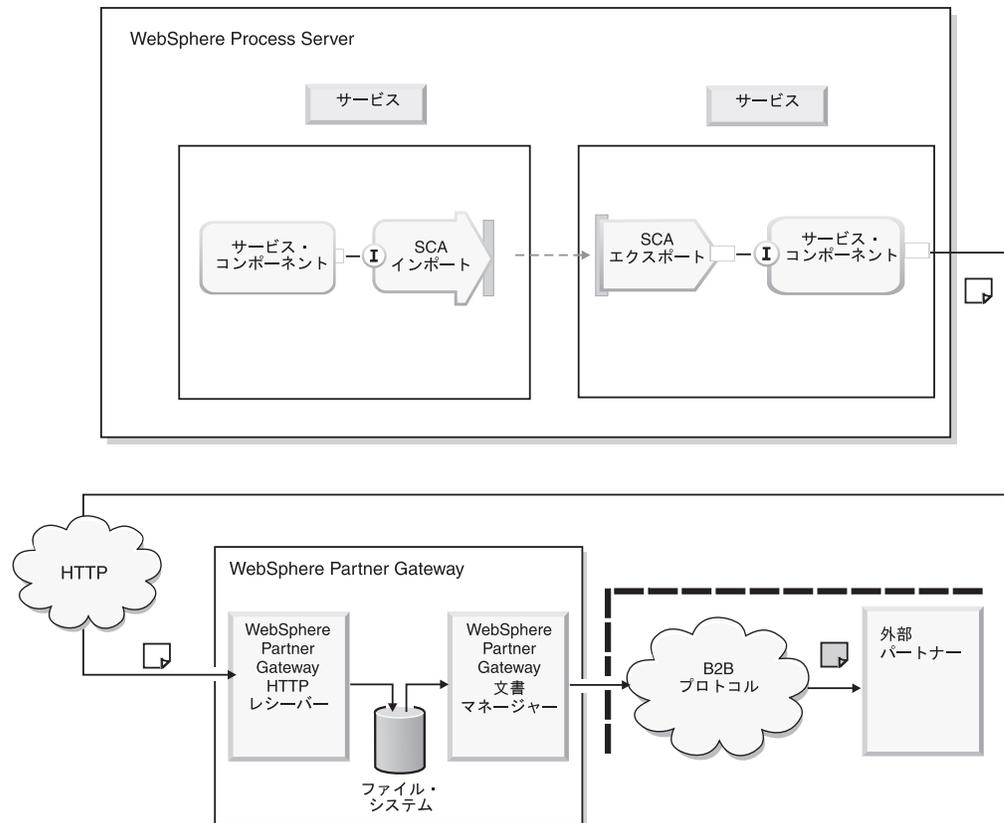


図 22. WebSphere Process Server から WebSphere Partner Gateway 上の HTTP レシーバーにメッセージを送信する仕組み

WebSphere Process Server サービスは、以下のようにメッセージを WebSphere Partner Gateway に送信します。

1. HTTP を介してビジネス文書を WebSphere Partner Gateway に送信する必要がある WebSphere Process Server SCA サービスが、それ自体の SCA インポート・バインディングを使用して HTTP POST を実行するコンポーネントを呼び出します。サービスの SCA インポート・バインディングが、ビジネス・オブジェクトとともにコンポーネントを呼び出します。
2. コンポーネントが、エクスポートでビジネス・オブジェクトを受信します。コンポーネントがビジネス・オブジェクトをビジネス文書に直列化します。97 ページの『メッセージを変換および送信するためのコンポーネントの作成』を参照してください。
3. コンポーネントが、HTTP POST を使用して WebSphere Partner Gateway HTTP レシーバー URL にメッセージを送信します。
4. WebSphere Partner Gateway が、HTTP レシーバーでこのメッセージを受信します。

WebSphere Partner Gateway の応答方法は、メッセージが同期応答を必要とするかどうか、あるいは HTTP トランスポート・レベル応答のみを必要とするかどうかによって異なります。HTTP レシーバーには、この判別を行う SyncCheck 構成ポイントがあります。レシーバーの作成時に SyncCheck を構成すること

も、レシーバーを変更して指定することもできます。SyncCheck 構成ポイントについて詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

- a. WebSphere Process Server がトランスポート・レベル応答のみを要求していると WebSphere Partner Gateway が判断した場合は、WebSphere Partner Gateway レシーバーが適切な HTTP 状況コードで応答します。次に、外部パートナーにビジネス文書を送付します。
  - b. WebSphere Process Server から受信したメッセージが応答ビジネス文書を要求していると WebSphere Partner Gateway が判断した場合は、HTTP トランスポート接続を開いたままにします。WebSphere Partner Gateway が、ビジネス文書を外部パートナーに送付します。外部パートナーが、要求文書を受信して応答文書を WebSphere Partner Gateway に送信します。WebSphere Partner Gateway が、WebSphere Process Server に対する HTTP 応答として応答ビジネス文書を戻します。
5. コンポーネントが HTTP 応答を受信します。以下のように、応答が HTTP 状況コードのみであるかビジネス文書でもあるかを判別します。
- a. 応答が正常終了を示す HTTP 状況コードである場合は、コンポーネントが呼び出しを戻します。WebSphere Process Server SCA サービス (コンポーネントを起動したサービス) は、処理を続行します。しかし、HTTP 状況コードが失敗を示している場合、コンポーネントは該当する障害を戻します。WebSphere Process Server SCA サービスは、適切な障害処理を実行します。
  - b. 応答がビジネス文書である場合は、コンポーネントがそのビジネス文書をビジネス・オブジェクトに変換します。次に、そのビジネス・オブジェクトを WebSphere Process Server SCA サービスに戻し、サービスが応答ビジネス・オブジェクトを処理します。

例えば、WebSphere Process Server から cXML 同期メッセージを受信し、対象が外部パートナーである場合には、WebSphere Partner Gateway はそのメッセージをパートナーに送信し、同期的に応答を受信します。次に WebSphere Partner Gateway が、最初に WebSphere Process Server から送信された要求と同じ HTTP 要求の HTTP 応答として WebSphere Process Server に応答を戻します。

**注:** WebSphere Partner Gateway は、バックエンド統合パッケージ化に対する要求専用呼び出しをサポートします。ただし、(WebSphere Partner Gateway ユーザー出口を経由する) バックエンド統合パッケージ化データ・バインディングを持つ要求応答フローを開発することもできます。82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』は、応答ビジネス・オブジェクトを応答ストリームに変換するための基礎として使用できます。

## WebSphere Process Server への文書の送信

このセクションでは、WebSphere Partner Gateway から送信された文書を WebSphere Process Server が受け入れることができるようにするための手順について説明します。

## WebSphere Partner Gateway のセットアップ

「*WebSphere Partner Gateway* ハブ構成ガイド」の説明に従って WebSphere Partner Gateway を構成します。WebSphere Partner Gateway の構成時は、以下の点に注意してください。

1. `com.ibm.bcg.server.sync.SoapSyncHdlr` ハンドラーを HTTP レシーバーの `syncCheck` 構成ポイントに追加します (両方向要求を処理する場合)。
2. 内部パートナーの HTTP 宛先が、WebSphere Partner Gateway による WebSphere Process Server へのメッセージ送信先の URL を示していることを確認します。

宛先は、WebSphere Process Server にデプロイした HTTP サブレットを指している必要があります。

## WebSphere Process Server のセットアップ

このセクションでは、WebSphere Partner Gateway からメッセージを受信するための WebSphere Process Server のセットアップ方法について説明します。

**ビジネス・オブジェクトの作成:** `BCGBackendIntegrationDataBindingUtil` クラスを使用してメッセージからビジネス・オブジェクトへの変換を行う場合は、トップレベル・ビジネス・オブジェクトを使用する必要があります (70 ページの『トップレベルおよび子ビジネス・オブジェクト』を参照してください)。BO エディターを使用してビジネス・オブジェクトを作成するか、インストール・メディアに収録されている `xsd` ファイル (`BCG_TLO_BackendPackaging.xsd`) をインポートすることができます。

バックエンド統合パッケージ化を使用しない (つまり、文書のパッケージ化がなしと指定された) 場合は、文書交換が要求されるごとにビジネス・オブジェクトを作成します。

**サブレットの作成:** このセクションでは、WebSphere Partner Gateway が HTTP トランスポートを介して WebSphere Process Server にメッセージを送信できるようにするために作成する必要があるサブレットについて説明します。サブレットは以下を行うように設計します。

- WebSphere Partner Gateway からメッセージを受信する
- ビジネス文書をビジネス・オブジェクトに変換する
- WebSphere Process Server 上の適切なサービスを呼び出す

バックエンド統合パッケージ化を使用する場合は、82 ページの『`BCGBackEndIntegrationDataBindingUtil` クラス』を使用して、サービスが想定する要求ビジネス・オブジェクトに HTTP 要求メッセージを変換できます。また、`BCGBackEndIntegrationDataBindingUtil` を使用して、サービスによって戻された応答ビジネス・オブジェクトから HTTP 応答メッセージを作成することもできます。`BCGBackEndIntegrationDataBindingUtil` クラスを使用して HTTP 要求メッセージを要求ビジネス・オブジェクトに変換する方法の例を以下に示します。この例では、83 ページの『`read` メソッド』および 83 ページの『`getDataObject` メソッド』を使用します。事前にトップレベル・オブジェクトの名前が判明している場合は、86 ページの『`setTLOTypeName` メソッド』を呼び出すこともできます。

```
try{
    DataObject rootBO = null;
    BCGBackEndIntegrationDataBindingUtil dataBinding = new
```

```

BCGBackendIntegrationDataBindingUtil();

    // request.getInputStream() gives the
    // backend integration packaging input stream received by servlet
    dataBinding.read(request.getInputStream());
    rootBO = dataBinding.getDataObject();

}
catch(Exception exp){
    System.out.println("Error occurred while creating request business
object: " + exp);
}

```

サーブレットを作成したらデプロイします。このサーブレットが受信する URL は、WebSphere Partner Gateway のパートナー接続用に構成した受信側宛先の URL として指定してください。

## WebSphere Process Server からの文書の送信

このセクションでは、WebSphere Process Server が WebSphere Partner Gateway に文書を送信できるようにするための手順について説明します。

### WebSphere Partner Gateway のセットアップ

「*WebSphere Partner Gateway* ハブ構成ガイド」の説明に従って WebSphere Partner Gateway を構成します。WebSphere Partner Gateway の構成時は、以下の点に注意してください。

1. WebSphere Process Server から送信された文書を受信するための HTTP/S レシーバーがまだ存在しない場合は、ハブに作成します。このレシーバーは、WebSphere Partner Gateway のレシーバー・コンポーネントが WebSphere Process Server からの文書を listen する URL を示します。
2. com.ibm.bcg.server.sync.SoapSyncHdlr ハンドラーを HTTP レシーバーの syncCheck 構成ポイントに追加します (両方向要求を処理する場合)。

### WebSphere Process Server のセットアップ

このセクションでは、WebSphere Partner Gateway にメッセージを送信するための WebSphere Process Server のセットアップ方法について説明します。WebSphere Process Service が HTTP を介して WebSphere Partner Gateway にメッセージを送信するには、WebSphere Partner Gateway HTTP レシーバーが文書を受信する URL に対して HTTP POST を実行できるコンポーネントが必要です。WebSphere Process Server サービスは、ビジネス文書を外部パートナーに送信するときに必ずこのコンポーネントを呼び出します。

**ビジネス・オブジェクトの作成:** BCGBackendIntegrationDataBindingUtil クラスを使用してビジネス文書からビジネス・オブジェクトへの変換を行う場合は、トップレベル・ビジネス・オブジェクトを使用する必要があります (70 ページの『トップレベルおよびビジネス・オブジェクト』を参照してください)。BO エディターを使用してビジネス・オブジェクトを作成するか、インストール・メディアに収録されている xsd ファイル (BCG\_TLO\_BackendPackaging.xsd) をインポートすることができます。

バックエンド統合パッケージ化を使用しない (つまり、パッケージ化がなしと指定された) 場合は、文書交換が要求されるごとにビジネス・オブジェクトを作成します。

**メッセージを変換および送信するためのコンポーネントの作成:** ビジネス・オブジェクトを入力ストリームに変換し、HTTP メッセージを WebSphere Partner Gateway に送信するコンポーネントを作成します。このコンポーネントは、サービスから取得したビジネス・オブジェクトをビジネス文書に直列化します。次に、JAVA API `java.net.HTTPURLConnection` を使用して、WebSphere Partner Gateway HTTP レシーバーが文書を受信する URL に対する HTTP POST を実行します。

パッケージ化を使用しない場合に、このコンポーネントが WebSphere Partner Gateway にメッセージを POST する方法を以下の例に示します。

```
String wpgHTTPTargetURL;    // URL of WebSphere Partner Gateway HTTP Target
byte payload[];            // message that needs to be sent to WebSphere Partner Gateway

// Set wpgHTTPTargetURL
// Set payload bytes. Convert business object received from service into bytes.
...

try{

    java.net.HttpURLConnection uc = (java.net.HttpURLConnection)
        new URL(wpgHTTPTargetURL).openConnection();

    uc.setDoInput(true);
    uc.setDoOutput(true);
    uc.setRequestMethod("POST");

    // Set the content type you want to send
    uc.setRequestProperty("Content-Type", "text/xml");

    uc.connect();
    BufferedOutputStream os = new BufferedOutputStream (uc.getOutputStream());
    os.write( payload);
    os.close();

    // If you are expecting response business document from WebSphere Partner
    // Gateway, then you have to read the response
    InputStream is = uc.getInputStream();
    if(is != null){
        BufferedInputStream bis = new BufferedInputStream
(uc.getInputStream());
        // Read bytes from BufferedInputStream obtained above
        // Convert response business document received from WebSphere
        // Partner Gateway into business object
        ...
        bis.close();
    }
}
catch(Throwable ex){
    System.out.println("Cannot post: " + ex.getMessage());
    ex.printStackTrace();
}
}
```

**重要:** バックエンド統合パッケージ化を使用して文書を WebSphere Process Server に送信する場合は、82 ページの『BCGBackEndIntegrationDataBindingUtil クラス』を使用して、サービス (またはコンポーネント) から受信したビジネス・オブジェクトを WebSphere Partner Gateway が要求するビジネス文書に変換できます。WebSphere Partner Gateway からの応答ビジネス文書が必要な場合は、

BCGBackEndIntegrationDataBindingUtil を使用して、サービスによって戻された応答ビジネス・オブジェクトから HTTP 応答メッセージを作成することもできます。

---

## トランスポートとしての JMS と WebSphere Process Server の統合

この章では、WebSphere Partner Gateway と WebSphere Process Server を JMS トランスポート・プロトコルを通じて統合する方法について説明します。ここでは、WebSphere Platform Messaging (デフォルトのメッセージング・プロバイダー) およびサービス統合バスを JMS プロバイダーとして使用し、WebSphere Partner Gateway を WebSphere Application Server にインストールする場合に、JMS 用に WebSphere Partner Gateway を構成する手順について説明しています。

この章では、JMS トランスポート・プロトコルを使用して WebSphere Partner Gateway と WebSphere Process Server との間で文書を送受信する方法を、以下のトピックにより説明します。

- 『 JMS トランスポート・プロトコルを使用した文書の送信』
- 99 ページの 『 JMS トランスポート・プロトコルを使用した文書の受信』
- 101 ページの 『WebSphere Application Server での WebSphere Process Server 環境のセットアップ』
- 104 ページの 『WebSphere Application Server 上に WebSphere Partner Gateway をインストール中の JMS の構成』

### JMS トランスポート・プロトコルを使用した文書の送信

WebSphere Partner Gateway が JMS トランスポート・プロトコルを使用して文書を WebSphere Process Server に送信するためには、JMS エクスポート・バインディングを使用し、JMS を介して WebSphere Process Server コンポーネントを呼び出す必要があります。SCA (Service Component Architecture) コンポーネントは、JMS エクスポート・バインディングで構成された JMS キューから JMS メッセージを受信できます。WebSphere Partner Gateway が JMS トランスポート・プロトコルを介して WebSphere Process Server に文書を送信する方法の概要を 99 ページの図 23 に示します。

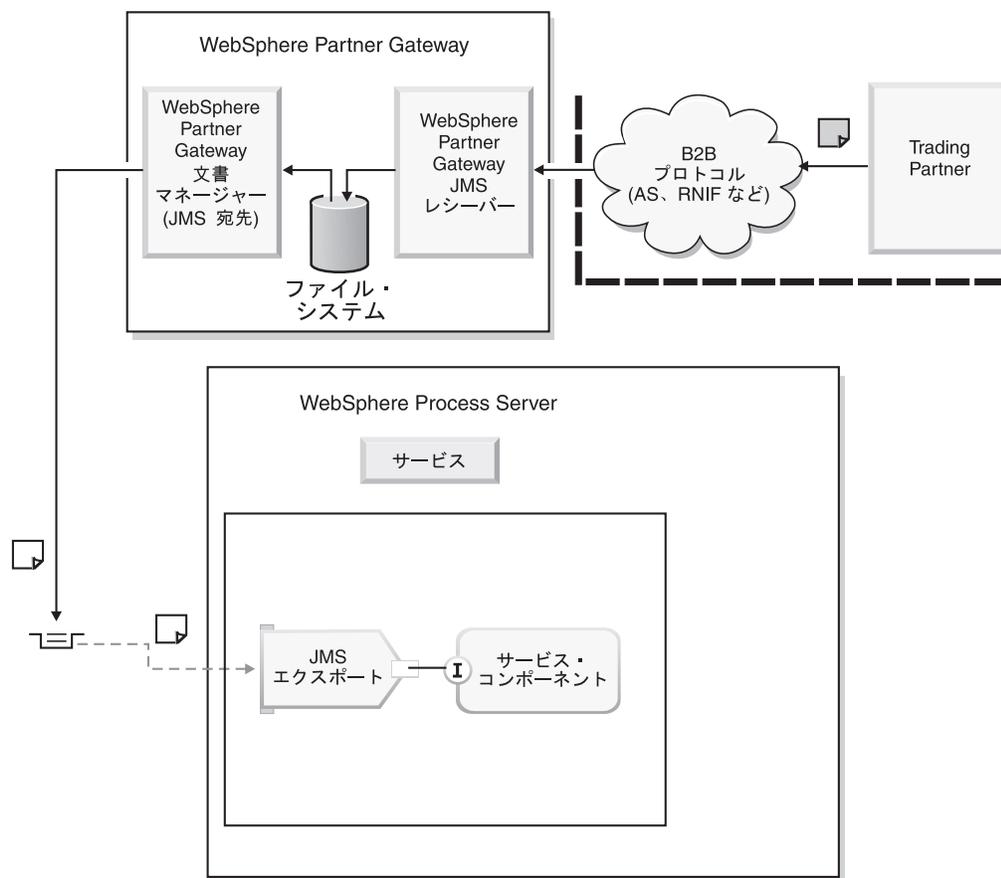


図 23. JMS を介して SCA サービスを呼び出す JMS エクスポート

以下の手順は、JMS を介して SCA サービスを呼び出すための SCA JMS エクスポート・バインディングの使用方法を説明しています。

1. 取引パートナーが、B2B プロトコルを使用して WebSphere Partner Gateway にビジネス文書を送信します。
2. WebSphere Partner Gateway が取引パートナーからビジネス文書を受信します。
3. WebSphere Partner Gateway が、このビジネス文書に対する構成済みパートナー接続を使用して WebSphere Process Server に文書を送付します。

JMS ベースの統合の場合は、パートナー接続の「受信側」宛先が JMS 宛先である必要があります。

この JMS 宛先は、SI バス宛先に JMS メッセージを送信するように構成します。

SI バス宛先は、SCA コンポーネントの SCA JMS エクスポート・バインディングで構成された JMS キューです。

## JMS トランスポート・プロトコルを使用した文書の受信

WebSphere Partner Gateway が JMS トランスポート・プロトコルを使用して WebSphere Process Server から文書を受信するためには、JMS インポート・バインディングを使用し、JMS を介して外部サービス呼び出す必要があります。SCA (Service Component Architecture) コンポーネントは、JMS インポート・バインディ

ングで構成された JMS キューに JMS メッセージを送信できます。WebSphere Partner Gateway が JMS トランスポート・プロトコルを介して WebSphere Process Server から文書を受信する方法の概要を図 24 に示します。

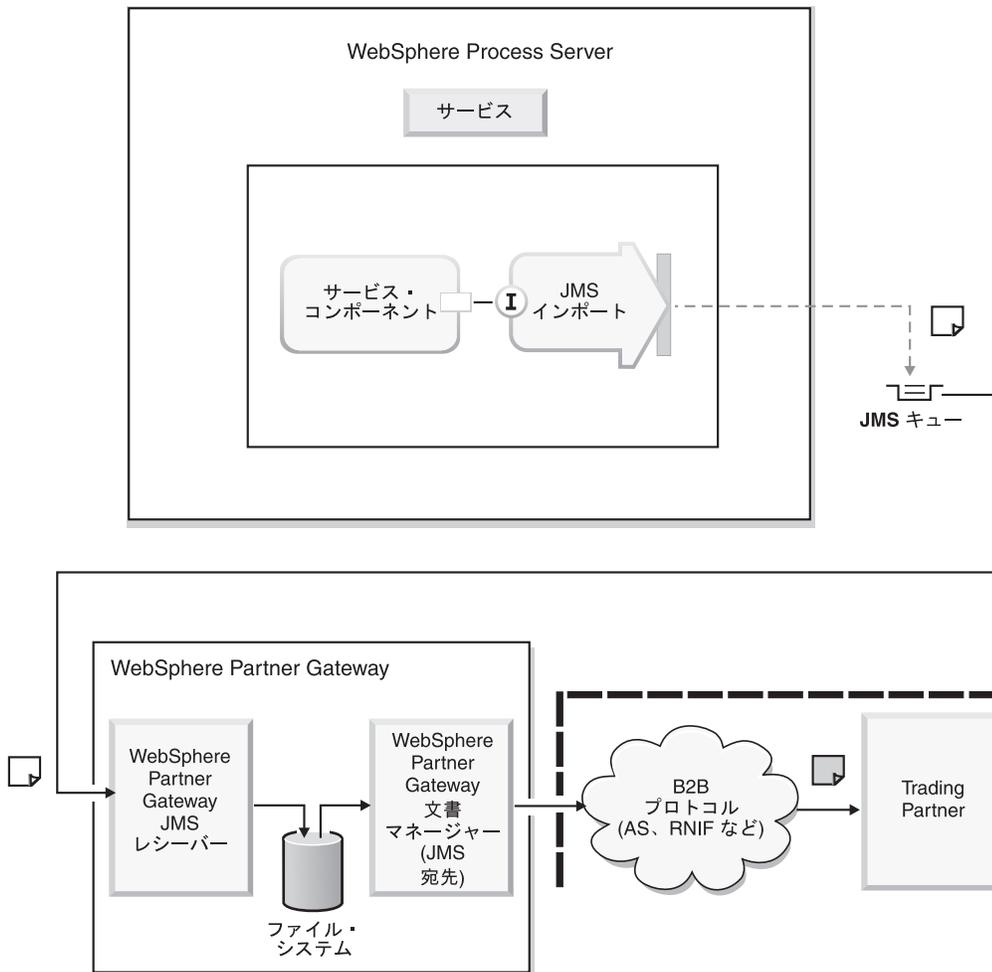


図 24. JMS を介して文書を受信する JMS インポート

以下の手順は、WebSphere Process Server から文書を受信するための SCA JMS インポート・バインディングの使用方を説明しています。

1. WebSphere Partner Gateway JMS レシーバーが WebSphere Process Server からビジネス文書を受信します。
2. WebSphere Partner Gateway が、このビジネス文書に対する構成済みパートナー接続を使用してこの文書を取引パートナーに送付します。
3. 取引パートナーが、相互に合意している B2B プロトコルを介して WebSphere Partner Gateway からビジネス文書を受信します。

JMS ベースの統合の場合、WebSphere Partner Gateway レシーバーの JMS レシーバーで構成する JMS キューは、WebSphere Process Server サービスが JMS メッセージを送信する JMS 宛先である必要があります。

この JMS キュー宛先は、SCA コンポーネントの SCA JMS インポート・バインディングで構成された JMS キュー宛先です。

## WebSphere Application Server での WebSphere Process Server 環境のセットアップ

このセクションでは、WebSphere Application Server で JMS トランスポート用に WebSphere Process Server 環境をセットアップするための手順について説明します。ここでは、WebSphere Process Server 成果物の作成および構成についても説明しています。

このセクションで説明する内容は、次のとおりです。

- 『WSDL での SCA サービスの作成』
- 『JMS インポートおよびエクスポート・バインディングのカスタマイズ』
- 102 ページの『JMS データ・バインディングの実装』
- 103 ページの『機能セレクターのカスタマイズ』

### WSDL での SCA サービスの作成

SCA サービスは、WebSphere Partner Gateway の B2B および取引パートナーとの対話機能を活用するためにエクスポート・バインディングおよびインポート・バインディングと組み合わせて使用します。Web サービス記述言語 (WSDL) を使用して SCA サービスを作成および定義する方法については、WebSphere Process Server のインフォメーション・センターを参照してください。

### JMS インポートおよびエクスポート・バインディングのカスタマイズ

コンポーネントの JMS インポート・バインディングは、JMS を介して外部サービスを呼び出すために使用します。SCA コンポーネントは、JMS インポート・バインディングで構成された JMS キューに JMS メッセージを送信できます。WebSphere Partner Gateway JMS レシーバーで構成されたキューに SCA コンポーネントがメッセージを送信できるようにするには、以下のようにします。

1. コンポーネントの SCA JMS インポートが必要とする JMS データ・バインディングを提供します。102 ページの『JMS データ・バインディングの実装』を参照してください。
2. 使用する環境に固有の JMS インポート・バインディング属性を指定します。コンポーネントの JMS インポートは、JMS インポート・バインディングで指定された JMS 宛先にメッセージを配置します。JMS インポート・バインディングで指定する JMS 宛先は、WebSphere Partner Gateway JMS レシーバーで構成されている JMS キューを参照する必要があります。JMS インポート・バインディング属性について詳しくは、WebSphere Process Server のインフォメーション・センターを参照してください。

コンポーネントの JMS エクスポート・バインディングは、JMS を介して WebSphere Process Server コンポーネントを呼び出すために使用されます。SCA コンポーネントが、JMS エクスポート・バインディングで構成された JMS 宛先から JMS メッセージを受信できるようにするには、以下のようにします。

1. コンポーネントの SCA JMS エクスポート・バインディングが必要とする JMS データ・バインディングを提供します。102 ページの『JMS データ・バインディングの実装』を参照してください。

2. コンポーネントの SCA JMS エクスポート・バインディングが必要とする機能セクターを提供します。103 ページの『機能セクターのカスタマイズ』を参照してください。
3. 使用する環境に固有の JMS エクスポート・バインディング属性を指定します。コンポーネントの JMS エクスポートは、JMS エクスポート・バインディングで指定された JMS 宛先から JMS メッセージを取得します。JMS エクスポート・バインディングで指定する JMS 宛先は、WebSphere Partner Gateway JMS 宛先で構成されている JMS キューを参照する必要があります。JMS インポート・バインディング属性について詳しくは、WebSphere Process Server のインフォメーション・センターを参照してください。

## JMS データ・バインディングの実装

WebSphere Process Server SCA JMS インポートおよびエクスポート・コンポーネントは、JMS データ・バインディングを指定するための構成情報を提供します。JMS データ・バインディングは、以下のように SCA JMS インポートおよびエクスポートによってビジネス・オブジェクトから JMS メッセージへの変換、および JMS メッセージからビジネス・オブジェクトへの変換に使用されます。

- SCA JMS エクスポート: サービスの SCA JMS エクスポートは、構成されている JMS データ・バインディングを活用して、WebSphere Partner Gateway から受信した JMS メッセージを、サービスのメソッドが想定するビジネス・オブジェクトに変換します。このビジネス・オブジェクトは、サービスのメソッドを呼び出すために使用されます。
- SCA JMS インポート: サービスの SCA JMS インポートは、構成されている JMS データ・バインディングを活用して、ビジネス・オブジェクトを WebSphere Partner Gateway が想定する JMS メッセージに変換します。

JMS データ・バインディングを開発および実装するには、WebSphere Process Server のインフォメーション・センターを参照してください。

バックエンド統合に JMS を使用する場合、WebSphere Partner Gateway はパッケージ化なしとバックエンド統合パッケージ化をサポートします。

パッケージ化なしの場合、WebSphere Process Server 提供のデフォルト JMS データ・バインディングを活用したり、独自のデータ・バインディングを実装したりすることができます。通常、WebSphere Partner Gateway と WebSphere Process Server の間で交換する JMS メッセージの形式が XML 以外である場合は、独自のデータ・バインディングを作成する必要があります。

バックエンド統合パッケージ化を使用する場合は、WebSphere Partner Gateway 提供の `com.ibm.bcg.dataBinding.BCGBackendIntegrationJMSDataBindingImpl` を利用できます。処理をカスタマイズするために、WebSphere Partner Gateway によって提供される `com.ibm.bcg.dataBinding.BCGBackendIntegrationDataBindingUtil` または `com.ibm.bcg.dataBinding.BCGBackendIntegrationJMSDataBindingImpl` を使用してデータ・バインディングを実装できます。これらのクラスについて詳しくは、87 ページの『`BCGBackendIntegrationJMSDataBindingImpl` クラス』および 82 ページの『`BCGBackendIntegrationDataBindingUtil` クラス』を参照してください。

バックエンド統合パッケージ化用の JMS データ・バインディング・インターフェースを実装するには、ユーザー指定の read メソッドおよび write メソッドを作成します。

## 機能セレクトラーのカスタマイズ

SCA JMS エクスポート・バインディングには機能セレクトラーが必要です。これは、JMS 宛先で受信されるビジネス文書に対して呼び出すサービス・メソッドを判別するために使用されます。

機能セレクトラーは、以下のいずれかの手法を使用して実装します。

- パッケージ化なし: WebSphere Partner Gateway が、パッケージ化なしで WebSphere Process Server に JMS メッセージを送信する場合には、呼び出す SCA サービスのメソッドをユーザーが判別する必要があります。パッケージ化なしの場合、ビジネス文書のタイプを判別する方法は、文書の解析です。文書を解析するためには、以下のいずれかの手法を使用できます。
    - ディスパッチャー: パッケージ化なしの場合、IBM では、文書の解析に機能セレクトラーを使用しないことをお勧めします。データ・バインディングでも文書を解析する必要があるため、この処理は冗長で負荷が高くなるだけです。IBM では、ビジネス文書に使用される実際の WebSphere Process Server コンポーネントに対するフロントエンドになるディスパッチャーまたは機能セレクトラー・コンポーネントを開発することをお勧めします。このディスパッチャー・コンポーネントの JMS エクスポート・バインディングは、WebSphere Partner Gateway からビジネス文書を受信します。このディスパッチャー・コンポーネントは、受信する可能性があるすべてのビジネス文書を受信できる 1 つのメソッドを持ちます。JMS エクスポート・バインディングは、JMS メッセージをビジネス・オブジェクトにアンマーシャルします。次に、そのビジネス・オブジェクトの対象となっているサービスのコンポーネントをディスパッチャー・モジュールが判別し、そのビジネス・オブジェクトとともにコンポーネントを呼び出します。
    - 最小限の解析: SCA JMS エクスポート・バインディングが複数のメソッド・バインディングを持つ場合は、ビジネス文書の最小限の解析を実行してビジネス文書のタイプ (PO、POConfirm、Invoice など) を判別する機能セレクトラーを実装できます。機能セレクトラーは、ビジネス文書のタイプをネイティブ・メソッド名として戻すことができます。構成時には、ビジネス文書タイプとして SCA JMS エクスポート・バインディングのメソッド・バインディングのネイティブ・メソッド名を指定できます。SCA JMS エクスポートは、メソッド・バインディングを使用して、呼び出す必要があるサービスのメソッドを解決できます。
- 注:** データ・バインディングは、ビジネス・オブジェクトを構成するためにビジネス文書を完全に解析します。したがって、機能セレクトラーで行うビジネス文書の解析はごく最小限にしてください。機能セレクトラーでの解析が高くつきすぎる場合は、この手法を使用しないでください。
- 簡易機能セレクトラー: ユーザーが 1 種類のビジネス文書しか想定していない場合は、文書を解析せずに常に同じネイティブ・メソッド名を戻す簡易機能セレクトラーを開発できます。構成時には、このメソッド名として SCA JMS エクスポート・バインディングのメソッド・バインディングのネイティブ・メソッド名を指定できます。SCA JMS エクスポートは、メソッド・バインディングを

使用して、呼び出す必要があるサービスのメソッドを判別できます。この手法の利点は、着信ビジネス文書を解析する必要がない点です。ただし、この手法には、JMS キューに 1 種類のビジネス文書しか存在できないという制限があります。

- バックエンド統合パッケージ化: WebSphere Partner Gateway がバックエンド統合パッケージ化を使用して JMS メッセージを WebSphere Process Server に送信する場合には、ユーザーは、バックエンド統合パッケージ化 JMS ヘッダーを利用して機能セクターを開発できます。JMS プロパティは、機能セクターによって入力 JMS メッセージから読み取ることができます。機能セクターは、バックエンド統合パッケージ化 JMS ヘッダーのいずれかの値を使用してネイティブ・メソッド名を構成します。例えば、RosettaNet メッセージの場合には、x-aux-payload ルート・タグ JMS プロパティを読み取ってビジネス文書のタイプを判別できます。それを利用して、機能セクターは、JMS エクスポートのメソッド・バインディングのネイティブ・メソッド名に一致するネイティブ・メソッド名を構成できます。

WebSphere Integration Developer を使用し、`commonj.connector.runtime.FunctionSelector` インターフェースを実装することによってカスタム機能セクターを作成します。例を次に示します。

```
public class PurchaseOrderSelector implements FunctionSelector {
    public String generateEISFunctionName(Object[] arg0)
        throws SelectorException {
        return "receiveMessage";
    }
}
```

## WebSphere Application Server 上に WebSphere Partner Gateway をインストール中の JMS の構成

このセクションでは、メッセージング・プロバイダーとして WebSphere Platform Messaging を使用する場合に、WebSphere Application Server で JMS トランスポート用に WebSphere Partner Gateway 環境をセットアップするための手順について説明します。このセクションでは、WebSphere Partner Gateway を WebSphere Application Server にインストールしていると想定します。

このセクションでは、ユーザーが WebSphere Platform Messaging およびサービス統合バスに精通していることを前提としています。

このセクションで説明する内容は、次のとおりです。

- 105 ページの『バス、JMS キュー、および接続ファクトリーの作成と構成』
- 108 ページの『JMS レシーバーの作成』
- 108 ページの『JMS 宛先の作成』
- 109 ページの『宛先キューの作成』

WebSphere Partner Gateway コンポーネントのアプリケーション・サーバーのメッセージ・エンジン (ME) および JNDI を使用すると、お客様のバス・トポロジーが制限される場合があります。したがって、IBM では、WebSphere Partner Gateway コンポーネント (JMS レシーバーおよび JMS 宛先) を別のアプリケーション・サーバー上の ME に接続することをお勧めします。この ME では、WebSphere Process Server を実行してもしなくても構いません。このトポロジーの概要については、

図 25 を参照してください。

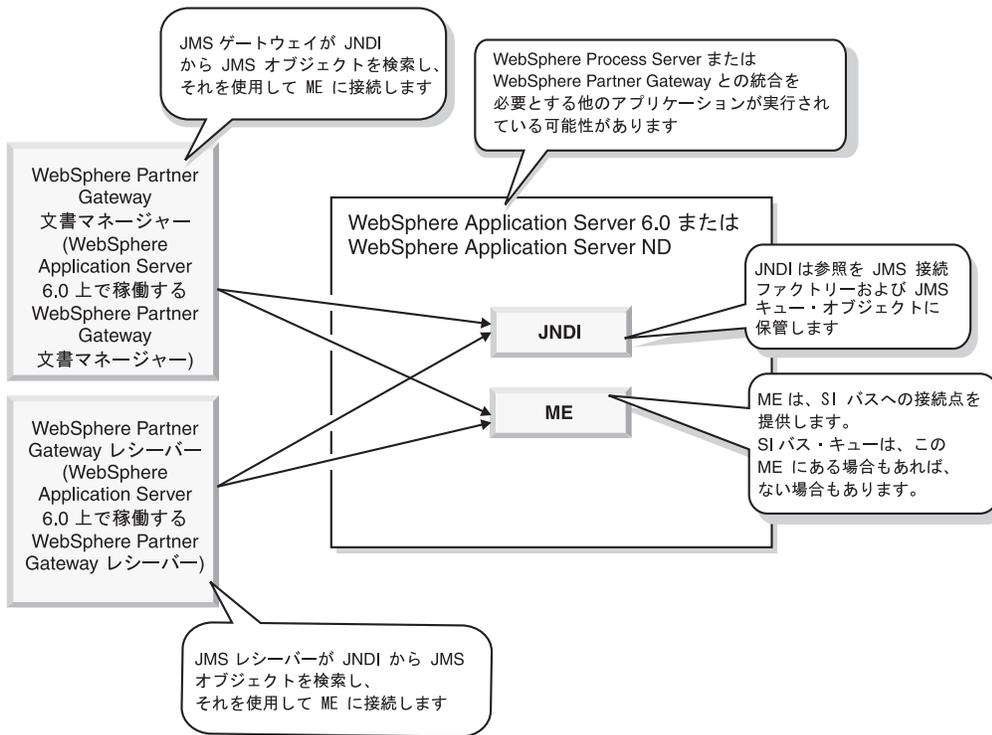


図 25. 別の WebSphere Application Server インスタンスからの ME および JNDI へのアクセス

## バス、JMS キュー、および接続ファクトリーの作成と構成

このセクションでは、WebSphere Partner Gateway がメッセージの送受信に使用するバス、JMS キュー、および接続ファクトリーを作成および構成する方法について説明します。

### サービス統合バスの作成:

#### このタスクについて

サービス統合バスは、メッセージ・ベースおよびサービス指向のアーキテクチャーを使用してアプリケーションをサポートします。バスとは、バスのメンバーとして追加された、相互接続されたサーバーおよびクラスターのグループのことです。アプリケーションは、そのバス・メンバーに関連付けられたいずれかのメッセージング・エンジンでバスに接続します。

### WebSphere 管理コンソールの使用:

1. バスを作成して名前を付けます。
  - a. 「サービス統合」 > 「バス」をクリックします。
  - b. 「新規」をクリックしてバス名を指定します。例として SIBUS とします。
  - c. 「適用」をクリックします。

- d. 「メッセージ」ウィンドウが表示されたら、「**保管**」をクリックして構成を保管します。このメッセージは、変更内容をマスター構成に適用することを確認するためのものです。
  - e. 現在の変更内容でマスター・リポジトリを更新するかどうか確認するメッセージが表示された場合は、再度「**保管**」をクリックします。
2. バス・メンバーをバスに追加します。
    - a. 新規作成したバスの名前をクリックします。
    - b. 「追加プロパティ」ペインで、「**バス・メンバー**」をクリックします。
    - c. 「**追加**」をクリックして、追加するサーバーまたはクラスターを選択します。
    - d. 「**次へ**」をクリックしてから「**終了**」をクリックし、新規バス・メンバーの追加を確認します。
    - e. 「メッセージ」ウィンドウが表示されたら、「**保管**」をクリックして構成を保管します。このメッセージは、変更内容をマスター構成に適用することを確認するためのものです。
    - f. 現在の変更内容でマスター・リポジトリを更新するかどうか確認するメッセージが表示された場合は、再度「**保管**」をクリックします。
  3. 以下のようにして宛先キューを作成します。
    - a. WebSphere Application Server のデフォルト・コンソールで、左のパネルにある「**システム統合 (System Integration)**」をクリックします。
    - b. 「バス」 < 「**SIBUS**」 (またはステップ 1 で作成したバスの名前) をクリックします。
    - c. 「追加プロパティ」ペインで、「**宛先**」をクリックします。「**新規**」をクリックします。
    - d. 宛先タイプに対応する「**キュー**」ラジオ・ボタンを選択し、「**次へ**」をクリックします。
    - e. 「**ID**」を入力します。例として Request とします。これにより、バスに宛先キューが作成されます。
  4. 「メッセージ」ウィンドウが表示されたら、「**保管**」をクリックして構成を保管します。このメッセージは、変更内容をマスター構成に適用することを確認するためのものです。
  5. 現在の変更内容でマスター・リポジトリを更新するかどうか確認するメッセージが表示された場合は、再度「**保管**」をクリックします。

### JMS キュー接続ファクトリーの作成:

#### このタスクについて

JMS キュー接続ファクトリーは、Point-to-Point メッセージングのために JMS キューの関連 JMS プロバイダーへの接続を作成するために使用されます。

#### WebSphere 管理コンソールの使用:

1. 以下の構文を使用して、「名前」フィールドおよび「**JNDI 名**」フィールドを入力し、キュー接続ファクトリーを作成します。
  - **名前:** SIBUS.JMSTargetQCF
  - **JNDI 名:** SIBUS/JMSTargetQCF

ここで、SIBUS は以前のステップで作成したバスの名前です。

2. バスを選択します。例として **SIBUS** とします。
3. 表示されるウィンドウで、作成したキュー接続ファクトリーをクリックし、「**プロバイダー・エンドポイント**」に以下を入力します。

```
IPAddress/Name:7276:BootstrapBasicMessaging
```

ここで、IPAddress は、WebSphere Application Server が稼働しているマシンの IP アドレスまたは名前です。このサービス統合バスのメッセージ・エンジンは、このマシンで稼働していると想定されます。7276 は、WebSphere インスタンスの SIB\_ENDPOINT\_ADDRESS に指定したポート番号です。IP アドレスが 9.26.234.100 のシステムでメッセージング・エンジンを実行し、このサーバーで稼働する WebSphere インスタンスの SIB\_ENDPOINT\_ADDRESS を 7276 と指定している場合は、以下のようにプロバイダー・エンドポイントを指定します。

```
9.26.234.100:7276:BootstrapBasicMessaging
```

4. 「メッセージ」ウィンドウが表示されたら、「**保管**」をクリックして構成を保管します。このメッセージは、変更内容をマスター構成に適用することを確認するためのものです。
5. 現在の変更内容でマスター・リポジトリを更新するかどうか確認するメッセージが表示された場合は、再度「**保管**」をクリックします。

### JMS キューの作成:

#### このタスクについて

JMS キューは、Point-to-Point メッセージングの宛先として使用されます。

WebSphere 管理コンソールの使用:

1. 「リソース」メニューを展開し、「**JMS プロバイダー**」 > 「**デフォルト・メッセージング**」をクリックします。
2. 表示されるページの「宛先」セクションで、「**JMS キュー**」をクリックします。
3. 「**新規**」をクリックします。
4. 以下の構文を使用して、「**名前**」フィールドおよび「**JNDI 名**」フィールドの両方にキュー名を入力します。
  - **名前:** Request.JMSTarget
  - **JNDI 名:** Request/JMSTarget
5. ドロップダウン・リストから**バス名** (例: SIBUS) および**キュー名**を選択します。
6. 「**OK**」をクリックします。
7. 「メッセージ」ウィンドウが表示されたら、「**保管**」をクリックして構成を保管します。このメッセージは、変更内容をマスター構成に適用することを確認するためのものです。
8. 現在の変更内容でマスター・リポジトリを更新するかどうか確認するメッセージが表示された場合は、再度「**保管**」をクリックします。

## JMS レシーバーの作成

### このタスクについて

このセクションでは、WebSphere Partner Gateway のレシーバーの作成について説明します。

WebSphere Partner Gateway コミュニティー・コンソールを使用します。

1. 「ハブ管理」 > 「ハブ構成」 > 「レシーバー」をクリックし、「レシーバー・リスト」を表示させます。
2. 「レシーバー・リスト」ページで、「レシーバーの作成」をクリックします。
3. 「レシーバーの詳細」セクションで、以下のステップを行います。
  - a. レシーバーの名前を入力します。例として、レシーバーを `JMSTarget` と呼ぶことにします。このフィールドは必須です。ここに入力した名前が「レシーバー」リストに表示されます。
  - b. (オプション) レシーバーの状況を指定します。デフォルトは「使用可能」です。使用可能なレシーバーは、文書を受け入れる準備ができています。使用不可のレシーバーは、文書を受け入れることができません。
  - c. (オプション) レシーバーの説明を入力します。
4. 「トランスポート」リストから、「**JMS**」を選択します。
5. JMS プロバイダーの URL を入力します。例を次に示します。

```
iiop://systemip:2809/
```

ここで、`systemip` は WebSphere Platform Messaging が稼働しているシステムの IP アドレスであり、`2809` は BOOTSTRAP サーバーが稼働しているデフォルト・ポートです。

6. JMS キュー名の値を入力します。このフィールドは必須です。この名前は、前のセクションで作成した JMS キューの JNDI 名に一致する必要があります。例えば、`Request/JMSTarget`。
7. JMS ファクトリー名の値を入力します。このフィールドは必須です。この名前は、以前に作成したキュー接続ファクトリー名に一致する必要があります。例えば、`SIBUS/JMSTargetQCF`。
8. 以下の JNDI ファクトリー名を入力します。

```
com.ibm.websphere.naming.WsnInitialContextFactory
```
9. 「保管」をクリックします。

## JMS 宛先の作成

### このタスクについて

このセクションでは、WebSphere Partner Gateway の宛先の作成について説明します。

以下のように、WebSphere Partner Gateway コミュニティー・コンソールを使用して JMS 宛先を作成します。

1. 「アカウント管理」 > 「プロファイル」 > 「パートナー」をクリックします。
2. 検索基準を入力して「検索」をクリックします。検索基準を入力せずに「検索」をクリックすると、すべてのパートナーのリストが表示されます。

3. 「詳細の表示」アイコンをクリックし、パートナーのプロファイルを表示させます。
4. 「宛先」をクリックします。
5. 「作成」をクリックします。
6. 「宛先リスト」ページに、宛先を識別する名前を入力します。このフィールドは必須です。
7. 「トランスポート」タイプ・リストから「JMS」を選択します。
8. 「アドレス」フィールドに、文書の配信先となる URI を入力します。例を次に示します。

```
iiop//systemip:2809
```

ここで、systemip は、WebSphere Platform Messaging が稼働しているシステムの IP アドレスです。

9. 「JMS ファクトリー名」フィールドに、JMS プロバイダーが JMS キューに接続するために使用する Java クラスの名前を入力します。これは、以前に作成した JMS ファクトリーの JNDI 名です。例を次に示します。

```
SIBUS/JMSTargetQCF
```

10. 「JMS キュー名」フィールドに、文書を送信する JMS キューの名前を入力します。これは、メッセージを格納するキューの JNDI 名です。例を次に示します。

```
Response/JMSTarget
```

11. 「JMS JNDI ファクトリー名」フィールドに、以下のファクトリーを入力します。

```
com.ibm.websphere.naming.WsnInitialContextFactory
```

12. 「保管」をクリックします。

## 宛先キューの作成

### このタスクについて

以下のように、WebSphere 管理コンソールを使用して WebSphere Application Server に宛先キューを作成します。

1. 左のパネルで「システム統合 (System Integration)」をクリックします。
2. 「バス」 < 「SIBUS」 (または以前のセクションで作成したバスの名前) をクリックします。
3. 「追加プロパティ」ペインで、「宛先」をクリックします。「新規」をクリックします。
4. 宛先タイプに対応する「キュー」ラジオ・ボタンを選択し、「次へ」をクリックします。
5. 「ID」を入力します。例として Response とします。これにより、バスに宛先キューが作成されます。

## WebSphere Process Server と SOAP/HTTP の統合

この章では、SOAP を使用して WebSphere Partner Gateway と WebSphere Process Server を HTTP トランスポート・プロトコルを使用して統合する方法について説明します。以下の情報を扱います。

- 『HTTP プロトコルを使用した SOAP メッセージの交換方法』
- 112 ページの『WebSphere Process Server によってホストされる Web サービスの起動』
- 113 ページの『外部パートナーによってホストされる Web サービスの起動』

### HTTP プロトコルを使用した SOAP メッセージの交換方法

外部パートナーが WebSphere Process Server から Web サービスを要求する場合、または WebSphere Process Server が外部パートナーから Web サービスを要求する場合、関連した SOAP メッセージが WebSphere Partner Gateway 上のレシーバーに送信されます。WebSphere Partner Gateway はプロキシとして機能し、サービスが置かれている URL に要求を渡します。

### WebSphere Process Server によってホストされる Web サービスの起動方法

外部パートナーが WebSphere Process Server 上でホストされる Web サービスを起動すると、以下のフローが発生します。

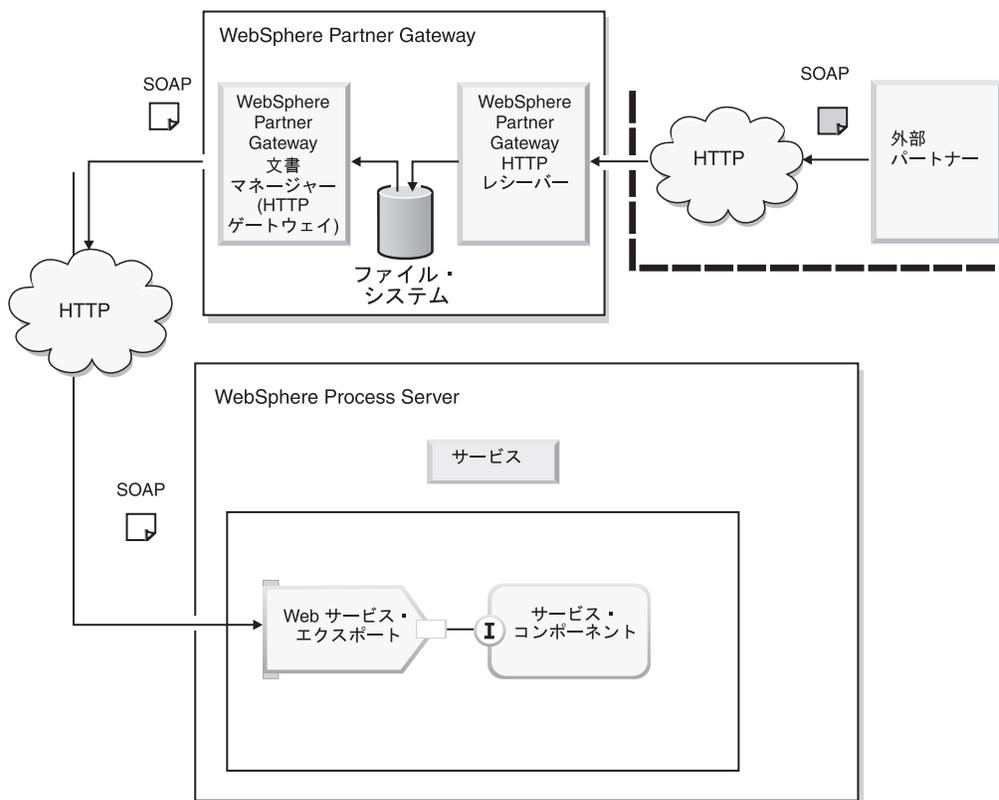


図 26. Web サービスを起動する外部パートナー

WebSphere Partner Gateway は、Web サービス要求を WebSphere Process Server に送付し、その後 Web サービス・エクスポート・バインディングを使用してサービスが起動されます。

外部パートナーに公開 WSDL (エンドポイントとして指定された WebSphere Partner Gateway の URL を持つ) を提供します。この URL の指定方法については、113 ページの『WebSphere Process Server の構成』を参照してください。WebSphere Partner Gateway はプロキシーとして機能します。パートナーから SOAP メッセージを受信し、対応するプライベート Web サービスを見つけ出します。次に、同じ SOAP メッセージを使用して、WebSphere Process Server 上の専用 Web サービスを起動します。次に、WebSphere Process Server によって返される HTTP 応答 (トランスポート・レベルの応答と Web サービス応答の両方) がパートナーに返されます。

### パートナーによってホストされる Web サービスの起動方法

WebSphere Process Server が外部パートナーから Web サービスを起動すると、以下のフローが発生します。

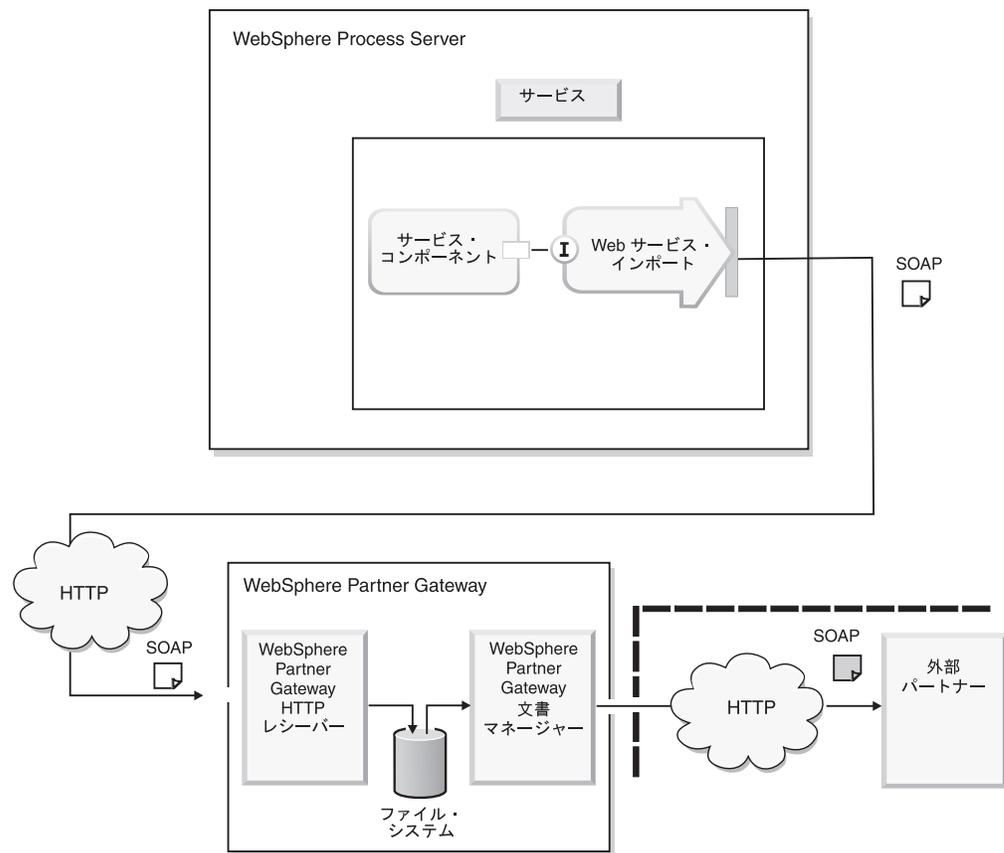


図 27. Web サービスを起動する WebSphere Process Server 上のサービス

WebSphere Process Server は、Web サービス・インポート・バインディングを使用して Web サービスを起動します。WebSphere Partner Gateway は、WebSphere Process Server から該当する外部パートナーに Web サービス要求を送付します。

WebSphere Partner Gateway は、コミュニティー・コンソールでの Web サービスのアップロード時に指定される Web サービス URL で、WebSphere Process Server が Web サービスを利用できるようにします。さらに、WebSphere Process Server 上のサービスは、「受信側パートナー」を識別するための URL パラメーターを指定する必要があります。詳細については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。WebSphere Partner Gateway はプロキシとして機能します。WebSphere Process Server から SOAP メッセージを受信し、対応する Web サービスおよび「宛先パートナー」を見つけ出します。次に、同じ SOAP メッセージを使用して、外部パートナーが提供する Web サービスを呼び出します。外部パートナーによって返される HTTP 応答 (トランスポート・レベルの応答と Web サービス応答の両方) が WebSphere Process Server に返されます。

外部パートナーから受信される応答は、要求と同じ HTTP 接続で WebSphere Process Server に返されます。要求のみの Web サービスの場合も、要求/応答 Web サービスの場合も、動作は同じです。

## WebSphere Process Server によってホストされる Web サービスの起動

このセクションでは、外部パートナーが WebSphere Process Server 上でホストされる Web サービスを起動する方法について説明します。

外部パートナーが、WebSphere Partner Gateway ハブ上の HTTP レシーバーに要求を送信します。WebSphere Partner Gateway が、ユーザーを認証し、外部パートナーと内部パートナー間の接続を検索し、内部パートナーで定義される宛先に要求を転送します。

**注:** 外部パートナーが SOAP メッセージを送信すると、外部パートナーは WebSphere Partner Gateway に対して自らを認証する必要があります。外部パートナーは、パートナーのビジネス ID、コンソール名、およびコンソール・パスワードを指定して、HTTP 基本認証を使用できます。あるいは、パートナーは、WebSphere Partner Gateway で事前に設定してある SSL クライアント証明書を提示できます。証明書について詳しくは、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

### WebSphere Partner Gateway の構成

「*WebSphere Partner Gateway* ハブ構成ガイド」の説明に従って、WebSphere Partner Gateway を Web サービス用に構成します。WebSphere Partner Gateway の構成時は、以下の点に注意してください。

1. `com.ibm.bcg.server.sync.SoapSyncHdlr` ハンドラーを HTTP レシーバーの `syncCheck` 構成ポイントに追加します (両方向要求を処理する場合)。
2. WebSphere Process Server から、コンポーネントの Web サービス・エクスポート・バインディングによって生成される WSDL を取得します。
3. 内部パートナーの HTTP 宛先が、WebSphere Partner Gateway による WebSphere Process Server へのメッセージ送信先の URL を示していることを確認します。この宛先は、ステップ 2 でアップロードした WSDL で指定されている `EndPointURL` を指します。

## WebSphere Process Server の構成

WebSphere Process Server インフォメーション・センターの資料に従って、WebSphere Process Server 上のコンポーネントをセットアップします。コンポーネントのエクスポートの作成時は、必ず「**Web サービス・バインディング (Web Service Binding)**」を選択してください。

## 外部パートナーによってホストされる Web サービスの起動

WebSphere Process Server 上のサービスが、WebSphere Partner Gateway 上の URL に対して SOAP 要求を行うと、WebSphere Partner Gateway は SOAP 要求を処理できるパートナーを識別します。その際、パートナー・プロファイルに対して、WebSphere Process Server で提供される基本認証を使用します。SOAP 要求が両方向の場合、WebSphere Partner Gateway は応答を外部パートナーから取得し、その応答を WebSphere Process Server に送信します。

## WebSphere Partner Gateway の構成

「*WebSphere Partner Gateway* ハブ構成ガイド」の説明に従って、WebSphere Partner Gateway を Web サービス用に構成します。WebSphere Partner Gateway の構成時は、以下の点に注意してください。

1. `com.ibm.bcg.server.sync.SoapSyncHdlr` ハンドラーを HTTP レシーバーの `syncCheck` 構成ポイントに追加します (両方向要求を処理する場合)。
2. 外部パートナーの HTTP 宛先が、WebSphere Partner Gateway によるメッセージ送信先の URL を示していることを確認します。この宛先は、構成プロセスの一環としてアップロードする WSDL で指定されている `EndPointURL` を指します。

## WebSphere Process Server の構成

WebSphere Process Server インフォメーション・センターの資料に従って、WebSphere Process Server 上のコンポーネントをセットアップします。サービスのプロバイダーから、Web サービスについて記述する WSDL ファイルを取得する必要があります。コンポーネントのモジュールに WSDL ファイルをインポートします。

WebSphere Partner Gateway は、外部パートナーが提供する Web サービスを URL で使用できるようにします。したがって、WebSphere Process Server は、WebSphere Partner Gateway HTTP レシーバーの URL でサービスを起動する必要があります。また、WebSphere Partner Gateway には、HTTP 基本認証の一環としてユーザー名およびパスワードを指定するための内部パートナーも必要です。ユーザー名およびパスワードについては、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。WebSphere Process Server が HTTP 基本認証を提供したり、WebSphere Process Server によって起動される Web サービスのエンドポイントを変更できるようにするには、以下のステップを実行します。

エンドポイント・アドレスを変更するには、以下のようにします。

1. WebSphere Process Server 管理コンソールにログインします。
2. WebSphere Partner Gateway から起動するサービスの Web サービス・クライアント・バインディングにナビゲートします。これは、次の場所にあります。「エ

「エンタープライズ・アプリケーション」 > <your\_application> > 「EJB モジュール」 > <your\_JAR\_file> > 「Web サービス・クライアント・バインディング」

3. WebSphere Partner Gateway から起動するサービス・ポートに「オーバーライド・エンドポイント URL」を指定します。WebSphere Process Server から Web サービス要求を受信する WebSphere Partner Gateway HTTP レシーバーの URL を指定します。エンドポイント URL の形式は、次のとおりです。

<IP\_address:port\_number>bcgreceiver/<target\_name>

4. 「適用」をクリックしてから、「OK」をクリックします。

基本認証に必要なユーザー名およびパスワードを指定するには、以下のステップを実行します。

1. WebSphere Process Server 管理コンソールにログインします。
2. WebSphere Partner Gateway から起動するサービスの Web サービス・クライアント・バインディングにナビゲートします。これは、次の場所にあります。「エンタープライズ・アプリケーション」 > <your\_application> > 「EJB モジュール」 > <your\_JAR\_files> > 「Web サービス・クライアント・セキュリティー・バインディング」 > 「HTTP 基本認証」
3. 基本認証 ID および基本認証パスワードを指定します。

認証 ID の形式は、次のとおりです。

*BusinessID/username*

例えば、内部パートナーのビジネス ID が 987654321 で、内部パートナーのユーザー名が admin の場合、認証 ID は次のようになります。

987654321/admin

4. 「適用」をクリックしてから、「OK」をクリックします。

---

## トランスポートとしてのファイル・システムと WebSphere Process Server の統合

この章では、ファイル・システム・プロトコルを使用して WebSphere Partner Gateway と WebSphere Process Server を統合する方法について説明します。

ファイル・システム・プロトコルによって、WebSphere Partner Gateway はメッセージを定義済みディレクトリー構造に配置して、メッセージを送信することができます。WebSphere Partner Gateway はディレクトリー構造からメッセージを読み込むことにより、メッセージを受信します。ファイル・システム・プロトコルは、パッケージ化なしのタイプをサポートします。

この章で説明する内容は、次のとおりです。

- 115 ページの『ファイル・システム・プロトコルを使用した文書の送信』
- 115 ページの『ファイル・システム・プロトコルを使用した文書の受信』
- 115 ページの『WebSphere Process Server 環境の設定』
- 117 ページの『WebSphere Partner Gateway 環境の設定』

## ファイル・システム・プロトコルを使用した文書の送信

WebSphere Process Server コンポーネントは、Service Component Architecture (SCA) J2C インポート・バインディングを利用して、ビジネス・データを WebSphere Adapter for Flat Files に送信します。アダプターは、ビジネス・データをファイル・システム内のディレクトリーに書き込みます。WebSphere Partner Gateway のファイル・システム・レシーバーは、このビジネス文書をファイル・システム上のディレクトリーから読み取り、適切な取引パートナーに送付します。

## ファイル・システム・プロトコルを使用した文書の受信

WebSphere Partner Gateway が、ビジネス文書を取引パートナーから受信します。WebSphere Partner Gateway の文書マネージャーが、文書をファイル・システム・ディレクトリーに書き込みます。WebSphere Adapter for Flat Files が、このディレクトリーでイベントをポーリングし、ビジネス文書を読み取ります。次いで、アダプターからのイベントを待っている Service Component Architecture (SCA) サービス・エクスポートを呼び出します。SCA サービスがビジネス文書を受け取ります。

**注:** WebSphere Adapter for Flat Files は、ファイルからの読み取りおよびファイルへの書き込みが可能なストリングまたは生データを予期します。したがって、WebSphere Process Server 環境では、ビジネス・オブジェクトをストリングまたは生データに直列化することが必要です。WebSphere Process Server サービスは、ビジネス・オブジェクトをビジネス文書に (またビジネス文書をビジネス・オブジェクトに) 直列化するコンポーネントを作成する必要があります。

## WebSphere Process Server 環境の設定

このセクションでは、WebSphere Application Server 上のファイル・システム・トランスポート用に WebSphere Process Server 環境をセットアップするための手順について説明します。ここでは、WebSphere Process Server 成果物の作成および構成についても説明しています。

このセクションで説明する内容は、次のとおりです。

- 『WebSphere Adapter for Flat Files のデプロイおよび構成』
- 116 ページの『SCA コンポーネントの作成』

### WebSphere Adapter for Flat Files のデプロイおよび構成

ファイル・システム・ベースの統合を実現するために、WebSphere Process Server は WebSphere Adapter for Flat Files (ファイル・システムとの間でビジネス・データの読み取りおよび書き込みを行える双方向アダプター) を利用します。

インバウンド通信の場合、アダプターは読み取り機能をサポートします。

アウトバウンド通信の場合、アダプターは以下の機能をサポートします。

- 作成
- 付加
- 削除
- 上書き
- 取り出し

- リスト
- 存在

WebSphere Adapter for Flat Files インストール済み環境には、WebSphere Process Server 上にデプロイ可能なリソース・アダプター・アーカイブ (RAR) ファイルがあります。RAR ファイルには、アダプターに付属のファイルが含まれています。WebSphere Adapter for Flat Files をデプロイする前に、アダプター RAR ファイルをインポートして、WebSphere Integration Developer 内にプロジェクトを作成する必要があります。

以下のセクションでは、WebSphere Adapter for Flat Files をデプロイおよび構成するために実行する必要がある手順の概要を示します。詳しくは、WebSphere Integration Developer の資料を参照してください。

### RAR ファイルのインポート

1. J2EE パースペクティブに切り替えて、RAR ファイルをインポートします。
2. RAR ファイルのインポート元 (インストール時にアダプターをコピーしたのと同じ場所) を指定して、プロジェクト名を指定します。

これで、ワークスペース内に新しい J2EE コネクター・プロジェクトが作成されます。

### プロジェクトへの外部依存関係の追加

外部依存関係を WebSphere Integration Developer プロジェクトの connectorModule にコピーして、内部ライブラリーとしてプロジェクトに追加します。これは依存関係を EAR ファイルに組み込むために必要であり、このファイルがエクスポートの対象となります。詳しくは、WebSphere Integration Developer の資料を参照してください。

### アダプターの構成

構成プロセスは、WebSphere Integration Developer のエンタープライズ・サービス・ディスカバリー・ウィザードを使用して実行します。このプロセスでは、初めてアダプターを構成するために必要なすべての情報を入力できます。エンタープライズ・サービス・ディスカバリー・ウィザードからの出力はビジネス統合モジュール (ビジネス・オブジェクトを含む)、インポート・ファイル (ActivationSpec で定義されるアウトバウンド処理を記述する)、エクスポート・ファイル (InteractionSpec で定義されるインバウンド・イベント処理を記述する)、および Web サービス記述言語 (WSDL) ファイルに保管されます。ファイル・システムからの読み取りおよびファイル・システムへの書き込みで使用できるモジュールにコンポーネントが作成されます。

### SCA コンポーネントの作成

SCA コンポーネントが、ファイル・システム・プロトコルを使用してビジネス文書を WebSphere Partner Gateway から受信できるようにするには、エンタープライズ・サービス・ディスカバリー・ツールを利用して、フラット・ファイル・アダプター用の必要なインポート・バインディングを生成できます。必要なインポート・バインディングを生成するには、エンタープライズ・サービス・ディスカバリー・ウィザードでサービス・タイプをインバウンドとして指定する必要があります。イ

ンバウンドの場合、メソッド名は READ です。SCA コンポーネント・エクスポートを、エンタープライズ・サービス・ディスカバリー・ツールで生成したインポート・バインディングにワイヤーします。

SCA コンポーネントが、ファイル・システム・プロトコルを使用してビジネス文書を WebSphere Partner Gateway に送信できるようにする場合は、エンタープライズ・サービス・ディスカバリー・ツールを利用して、フラット・ファイル・アダプター用の必要なエクスポート・バインディングを生成できます。必要なエクスポート・バインディングを生成するには、エンタープライズ・サービス・ディスカバリー・ウィザードでサービス・タイプをアウトバウンドとして指定する必要があります。エンタープライズ・サービス・ディスカバリー・ツールで生成したエクスポート・バインディングを SCA コンポーネントのインポート・バインディングにワイヤーします。

## WebSphere Partner Gateway 環境の設定

### このタスクについて

このセクションでは、WebSphere Application Server 上のファイル・システム・トランスポート用に WebSphere Partner Gateway 環境をセットアップするための手順について説明します。

WebSphere Partner Gateway コミュニティー・コンソールを使用します。

1. WebSphere Process Server または外部パートナーからハブに送信される文書を受信するファイル・システム・レシーバーをハブに作成します。
2. 内部パートナー・プロファイル (WebSphere Partner Gateway が文書を WebSphere Process Server に送信するために使用するファイル・システム宛先も含む) を作成します (既存のものがない場合)。
3. 外部パートナー・プロファイル (WebSphere Partner Gateway が文書をパートナーに送信するために使用するファイル・システム宛先も含む) を作成します。
4. WSDL ファイル、変換マップ、RosettaNet パッケージ、またはその他の文書定義手段をインポートして、交換する文書タイプの文書定義が WebSphere Partner Gateway のコミュニティ・コンソールの「文書フロー定義」ページに表示されるようにします。
5. ハブが (WebSphere Process Server または外部パートナーから) 受信する文書タイプと、ハブが (WebSphere Process Server または外部パートナー) に送信する文書タイプとの間のインタラクションを作成します。
6. 内部パートナーおよび外部パートナーのプロファイルに B2B 機能を作成し、送受信可能な文書タイプを指定します。
7. ハブが (WebSphere Process Server または外部パートナーから) 受信する文書タイプと、ハブが (WebSphere Process Server または外部パートナー) に送信する文書タイプとの間のインタラクションを作成します。
8. 内部パートナーと外部パートナーの間のパートナー接続を作成して、ソース・パートナー (文書の送信側)、ターゲット・パートナー (文書の受信側)、およびハブが文書を変換するために実行するアクション (ある場合) を指定します。



---

## 第 4 章 WebSphere InterChange Server との統合

以下の章では、WebSphere Partner Gateway を WebSphere InterChange Server と統合する手順について詳しく説明します。

---

### InterChange Server 統合の概要

この章では、WebSphere Partner Gateway を WebSphere InterChange Server と統合する方法について説明します。

#### 注記:

1. WebSphere Partner Gateway とバックエンド・システムの統合に使用する一般的なプロセスについては、11 ページの『バックエンド統合の計画』を参照してください。
2. この章では、読者が WebSphere InterChange Server と、コラボレーション、ビジネス・オブジェクト、およびアダプターなどの関連コンポーネントに精通していることを前提にしています。

WebSphere Partner Gateway をバックエンド・システムと統合する作業は、たいていの場合、異なる 2 人の担当者、つまり役割が行います。各役割は、その専門技術に応じて、特定のコンポーネントを構成します。したがって、この章では、WebSphere InterChange Server との統合は WebSphere Partner Gateway の構成と InterChange Server の構成に分割されています。表 35 に、これらの構成の役割と、関連する構成情報が記載されている本章内のセクションを示します。

表 35. InterChange Server を統合するための役割

構成の役割	詳細の参照先
WebSphere Partner Gateway の構成	<ol style="list-style-type: none"><li>1. 120 ページの『InterChange Server との統合の計画』</li><li>2. 123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』</li></ol>
WebSphere InterChange Server の構成	<ol style="list-style-type: none"><li>1. 120 ページの『InterChange Server との統合の計画』</li><li>2. 126 ページの『InterChange Server の構成』</li></ol>

注: 上記の各構成の役割は別々に実行できますが、各構成では、2 つのコンポーネント間で通信できるようにするための共通情報も必要になります。

この章で説明する内容は、次のとおりです。

- 120 ページの『InterChange Server との統合の計画』
- 123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』
- 126 ページの『InterChange Server の構成』

- 131 ページの『添付ファイルのある文書の処理』

## InterChange Server との統合の計画

WebSphere InterChange Server との統合を計画するには、11 ページの『バックエンド統合の計画』で説明されている手順を実行してください。表 36 には、WebSphere Partner Gateway と InterChange Server (ICS) とを統合するための統合手順が要約されています。

表 36. WebSphere InterChange Server との統合の計画

統合の手順	詳細の参照先
1. サポートされているバージョンの WebSphere InterChange Server がインストール済みで、これを WebSphere Partner Gateway が使用可能であることを確認します。	61 ページの『 WebSphere Process Server 統合の概要』 『WebSphere Partner Gateway がサポートする InterChange Server のバージョン』
2. WebSphere Partner Gateway 文書のビジネス・プロトコルを決定します。	11 ページの『バックエンド統合の計画』 : 11 ページの『ビジネス・プロトコルの選択』
3. 文書のパッケージ化タイプ (「なし」またはバックエンド統合) を決定します。	11 ページの『バックエンド統合の計画』 : 28 ページの『使用するパッケージ化』
4. WebSphere Partner Gateway と WebSphere InterChange Server との間で使用するトランスポート・プロトコルを決定します。	61 ページの『 WebSphere Process Server 統合の概要』 : 121 ページの『InterChange Server がサポートするメッセージ・トランスポート』
5. WebSphere Partner Gateway を構成します。	61 ページの『 WebSphere Process Server 統合の概要』 : 123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』
6. 選択したトランスポート・プロトコルを通じて使用するために、WebSphere InterChange Server コンポーネントを構成します。	61 ページの『 WebSphere Process Server 統合の概要』 : 126 ページの『InterChange Server の構成』

## WebSphere Partner Gateway がサポートする InterChange Server のバージョン

WebSphere Partner Gateway バージョン 6.2 では、InterChange Server の以下のバージョンとの統合がサポートされます。

- 4.3.0

InterChange Server は、Windows 2000 や UNIX ベースのプラットフォームを含む、いくつかのプラットフォームで使用可能です。詳しくは、WebSphere InterChange Server 資料セット内の InterChange Server のインストール・ガイドを参照してください。

WBI Adapter を使用して、WebSphere Partner Gateway と Interchange Server を統合します。サポートされるアダプターは、以下のとおりです。

- WBIA HTTP Adapter v1.3

- WBI JMS Adapter v2.8.x
- WBI Web Services Adapter v3.4.x

アダプターでは、WBI Data Handler を使用します。XML メッセージを処理する場合は、必ず WebSphere Business Integration Data Handler for XML バージョン 2.3.1 以降を使用してください。cXML メッセージの場合は、Data Handler for XML バージョン 2.4.1 以降を使用してください。

## InterChange Server がサポートするメッセージ・トランスポート

WebSphere Partner Gateway は、特定のメッセージ・トランスポート・プロトコルを通じてメッセージを InterChange Server に送信する場合、このメッセージを適切なアダプターに送信します。このコンポーネントは、特定のトランスポート・プロトコルを理解し、メッセージを InterChange Server に転送します。同様に、InterChange Server は、メッセージを WebSphere Partner Gateway に送信する場合、適切なトランスポート・プロトコルを通じて WebSphere Partner Gateway へメッセージを転送するために、メッセージを適切なアダプターに送信します。

アダプターはデータ・ハンドラーと連携します。データ・ハンドラーは、直列化されたデータからビジネス・オブジェクトへの実際の変換、またはビジネス・オブジェクトから直列化されたデータへの実際の変換を処理します。ペイロード・タイプに合ったデータ・ハンドラーを使用して、このような変換が実行されます。

InterChange Server では、以下の 2 つのメッセージ・トランスポート・プロトコルがサポートされています。

- HTTP トランスポート・プロトコル

**注:** HTTP での Web サービスの交換は、別のセクションで扱います。Web サービスは HTTP で伝送される他の文書とは異なる方法で交換されるためです。167 ページの『HTTP/S を通じた SOAP 文書の送信』を参照してください。

- JMS トランスポート・プロトコル

**注:** InterChange Server は、ファイル・ベースの統合など、他のタイプの統合オプションを提供します。ファイル・ベース統合を使用して文書の交換を使用可能にするための詳細については、WebSphere InterChange Server 資料を参照してください。

ビジネスのニーズに最も適したトランスポート・プロトコルを使用します。以下の事項を検討してください。

- 第一に、外部パートナーと WebSphere Partner Gateway との間で使用するトランスポート・プロトコルが、使用する統合手段で使用可能であることを確認します。40 ページの『使用するメッセージ・トランスポート』を参照してください。
- WebSphere InterChange Server への SOAP 文書の送受信には、HTTP トランスポート・プロトコルを使用する必要があります。詳細については、167 ページの『HTTP/S を通じた SOAP 文書の送信』を参照してください。

### HTTP:

HTTP を通じて WebSphere Partner Gateway と InterChange Server との間で文書を送信および受信するために必要なアダプターは、次の条件によって異なります。

- 送信する文書のタイプ

- 統合の対象とする InterChange Server のバージョン

注: HTTP トランスポート・プロトコルに言及する部分はすべて、HTTPS にも同様に適用されます。

表 37 には、アダプターを InterChange Server と連携して使用するよう構成する方法についての情報の入手先が要約されています。

表 37. InterChange Server を使用する HTTP トランスポートの構成

条件	詳細の参照先
InterChange Server を用いて SOAP 以外の文書を転送する場合	152 ページの『ICS での HTTP トランスポート・プロトコルの使用』
SOAP 文書を送信する場合	167 ページの『HTTP/S を通じた SOAP 文書の送信』

### JMS:

JMS を通じて WebSphere Partner Gateway と InterChange Server との間で文書を送信および受信するために必要なコンポーネントは、171 ページの表 63 に要約されています。基本的に、JMS をサポートするには WebSphere Business Integration Adapter for JMS を使用することになります。Adapter for JMS は、InterChange Server 内部でコラボレーションを非同期的に呼び出します。

### 各トランスポートを使用する利点:

InterChange Server とともに使用するメッセージ・トランスポートを決定するとき、各トランスポートの以下のような利点を考慮してください。

- Adapter for HTTP は、同期通信をサポートします。同期トランザクションが必要な場合は、HTTP トランスポート・プロトコルを使用する必要があります。
- Adapter for JMS は、WebSphere Partner Gateway から WebSphere InterChange Server への「保証付きイベント送達」機能を備えています。

保証付きイベント送達によって、イベントが失われたり 2 回送信されたりしないことが保証されます。

## InterChange Server 統合のサポート

WebSphere Partner Gateway には、InterChange Server との統合プロセスで役立つサンプルがあります。これらのサンプルは、WebSphere Partner Gateway 製品ディレクトリー内の以下のサブディレクトリーにあります。

Integration/WBI/WICS/samples

123 ページの表 38 には、InterChange Server がサポートしているさまざまなトランスポート・プロトコルの samples ディレクトリーのサブディレクトリーを示します。

表 38. InterChange Server 統合のサンプル

トランスポート・プロトコル	InterChange Server のバージョン	Samples サブディレクトリー
HTTP	サポートされているすべてのバージョン	汎用のサンプル: HTTP RosettaNet 固有のサンプル: RosettaNet/HTTP
JMS	サポートされているすべてのバージョン	汎用のサンプル: JMS RosettaNet 固有のサンプル: RosettaNet/JMS

## WebSphere Partner Gateway を InterChange Server 用に構成する

バックエンド・システムと通信するように WebSphere Partner Gateway を構成する方法の一般的な概要は、51 ページの『WebSphere Partner Gateway の構成』に記載されています。この章では、InterChange Server と通信するように WebSphere Partner Gateway を構成するために必要な手順をまとめています。

WebSphere Partner Gateway を構成するには、以下の手順を実行します。

- 発信文書をサポートするための構成

WebSphere Partner Gateway から InterChange Server への文書の送信については、『発信文書のサポート』を参照してください。

- 着信文書用の構成

InterChange Server から WebSphere Partner Gateway への文書の送信については、124 ページの『着信文書のサポート』を参照してください。

### 発信文書のサポート

WebSphere Partner Gateway が任意のバックエンド・システムに文書を送信できるようにするには、53 ページの『パートナー文書の送信先の定義』で説明されている手順を実行する必要があります。バックエンド・システムが InterChange Server である場合、トランスポート・タイプが WebSphere Partner Gateway および InterChange Server 間のメッセージに使用されるトランスポート・プロトコルに一致する宛先を作成する必要があります。ハブは、文書を InterChange Server に送信する場合、文書の送付先を知る必要があります。このロケーションは、使用するトランスポート・プロトコルに準拠していなければなりません。トランスポート・プロトコルは、InterChange Server でサポートされているものでなければなりません (121 ページの『InterChange Server がサポートするメッセージ・トランスポート』参照)。

以下のセクションでは、InterChange Server がサポートするトランスポート・プロトコル用の宛先を作成する方法をまとめています。

- 『HTTP トランスポート・プロトコルで文書を ICS に送信するための構成』
- 124 ページの『JMS トランスポート・プロトコルで文書を ICS に送信するための構成』

**HTTP トランスポート・プロトコルで文書を ICS に送信するための構成:** ハブは、HTTP プロトコルを通じて文書を InterChange Server に送信する場合、ハブは

定義された宛先を経由してメッセージを送付します。この宛先は、InterChange Server による文書の受信が可能な URL を示します。InterChange Server が HTTP プロトコルを使用する場合、アダプターは、適切な URL で文書を受信し、次にその URL で文書を InterChange Server へ送信できます。

ハブが HTTP トランスポート・プロトコルを通じて宛先経由で文書を送信できるようにするには、コミュニティー・コンソールの「宛先の詳細」ページから宛先を作成する必要があります。HTTP 1.1 トランスポート・プロトコルを使用し、適切なアダプターの listen 先 URL に書き込むには、この宛先を構成する必要があります。

注: 宛先の作成方法の概要は、53 ページの『パートナー文書の送信先の定義』で説明されています。

**JMS トランスポート・プロトコルで文書を ICS に送信するための構成:** ハブが JMS プロトコルを通じて発信文書を InterChange Server に送信する場合、ハブは適切な JMS キューへ文書を送付します。InterChange Server は JMS キューから文書を取り出すことができます。ハブがこの JMS ロケーションを取得できるようにするには、WebSphere Partner Gateway 内に JMS トランスポート・プロトコルを使用する宛先を作成する必要があります。この宛先は、Adapter for JMS によるポーリング先キューに書き込むことができるように構成する必要があります。

注: 宛先の作成方法の概要については、53 ページの『パートナー文書の送信先の定義』を参照してください。

ハブが JMS トランスポート・プロトコルを通じて宛先経由で文書を送信できるようにするには、コミュニティー・コンソールの「宛先の詳細」ページから宛先を作成する必要があります。最新のフィックスパックが適用された WebSphere MQ パーバージョン 6.0 を JMS プロバイダーとして使用する場合の詳細な手順については、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。さらに、「宛先の詳細」ページで JMS プロトコルに表 39 で指定された情報を指定します。

表 39. ICS と通信するための「宛先の詳細」ページの JMS 値

「宛先の詳細」フィールド	値	注記および制約事項
JMS メッセージ・クラス	TextMessage、BytesMessage、または StreamMessage	
JMS キュー名	Adapter for JMS 用の入力キューと同じ JMS キュー名	このキューは、Adapter for JMS の入力キューのリスト内に含まれていなければなりません。すなわち、アダプターは着信イベントを取得するためにこのキューをポーリングする必要があります。詳細については、178 ページの『JMS キューの識別』を参照してください。

## 着信文書のサポート

WebSphere Partner Gateway が任意のバックエンド・システムから文書を受信できるようにするには、57 ページの『バックエンド文書を取り出す場所の定義』で説明さ

れている手順を実行する必要があります。バックエンド・システムが InterChange Server である場合には、ハブで以下の手順を実行する必要があります。

1. 内部パートナーのパートナー・プロファイルの一部として、宛先タイプを定義し、レシーバーが listen する関連 IP アドレスを指定します。
2. トランスポート・タイプが WebSphere Partner Gateway および InterChange Server 間の文書に使用されるトランスポート・プロトコルに一致する受信側を作成します。

ハブは、InterChange Server から文書を受信するには、メッセージを取り出すロケーションを知る必要があります。このロケーションは、使用するトランスポート・プロトコルに準拠していなければなりません。トランスポート・プロトコルは、InterChange Server でサポートされているものでなければなりません (121 ページの『InterChange Server がサポートするメッセージ・トランスポート』参照)。

以下のセクションでは、InterChange Server がサポートするトランスポート・プロトコル用の受信側を作成する方法をまとめています。

**HTTP トランスポート・プロトコルで文書を ICS から受信するための構成:** ハブが HTTP トランスポート・プロトコルを通じて文書を受信すると、受信側は定義された受信側から文書を取り出します。この受信側は、受信側が InterChange Server からの文書を取得するために listen する URL を示します。InterChange Server が HTTP トランスポート・プロトコルを使用する場合、アダプターが適切な URL へ文書を送信します。ハブはその URL で文書を受信できます。

ハブが HTTP トランスポート・プロトコルを通じて受信側経由で文書を受信できるようにするには、コミュニティー・コンソールの「受信側リスト」ページから受信側を作成する必要があります。この受信側は HTTP 1.1 トランスポート・プロトコルを使用する必要があります。ハブは、この URL を以下の情報の組み合わせと判断します。

- ホスト・コンピューターの IP アドレス。内部パートナーのプロファイル内部から取得します。
- 受信側 URL。受信側定義の URL フィールドから取得します。

**注:** 受信側の作成方法の概要は、57 ページの『バックエンド文書を取り出す場所の定義』で説明されています。

InterChange Server がこの受信側に文書を送信できるようにするには、そのアダプターを構成して、この URL に文書を送信できるようにする必要があります。したがって、InterChange Server の構成に対してこの受信側 URL が使用できることを確認する必要があります。

**JMS トランスポート・プロトコルで文書を ICS から受信するための構成:** ハブは、InterChange Server から JMS プロトコルを通じて文書を受信すると、InterChange Server による文書の送信先である該当の JMS キューから文書を取得します。ハブがこの JMS ロケーションを取得できるようにするには、WebSphere Partner Gateway 内に JMS トランスポート・プロトコルを使用する受信側を作成する必要があります。ハブは、その入力キュー上にある文書を、受信側を通じて listen し、取得します。

注: 受信側の作成方法の概要については、57 ページの『バックエンド文書を取り出す場所の定義』を参照してください。

ハブが JMS トランスポートを通じて受信側経由で文書を受信できるようにするには、コミュニティー・コンソールの「受信側リスト」ページから受信側を作成する必要があります。最新のフィックスパックが適用された WebSphere MQ バージョン 6.0 を JMS プロバイダーとして使用する場合の詳細な手順については、

「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。さらに、「レシーバーの詳細」ページで JMS プロトコルに表 40 で指定された情報を指定します。

表 40. ICS と通信するための「レシーバーの詳細」ページの JMS 値

「レシーバーの詳細」フィールド	値	注記および制約事項
JMS メッセージ・クラス	TextMessage	
JMS キュー名	Adapter for JMS 用の出力キューと同じ JMS キュー名	このキューは、Adapter for JMS の出力キューとしてリストされていなければなりません。すなわち、アダプターはこのキューに文書を送信する必要があります。詳細については、178 ページの『JMS キューの識別』を参照してください。

## InterChange Server の構成

WebSphere Partner Gateway と InterChange Server 間の対話用に、System Manager ツール内で統合コンポーネント・ライブラリー (ICL) を作成する必要があります。この ICL には、以下の成果物が組み込まれます。

- ビジネス・オブジェクト定義
- コネクター・オブジェクト
- コラボレーション・テンプレートおよびコラボレーション・オブジェクト

ユーザー・プロダクトを作成し、InterChange Server および WebSphere Partner Gateway 間の特定の対話に必要な成果物を ICL から選択する必要があります。

注: ICL の作成方法と InterChange Server の構成方法について詳しくは、WebSphere InterChange Server 資料セット内の「システム・インプリメンテーション・ガイド」を参照してください。

### ビジネス・オブジェクト定義の作成

WebSphere Partner Gateway がメッセージを アダプターに送信すると、アダプターは、このメッセージを 1 つ以上のビジネス・オブジェクト の形式で InterChange Server に送付します。InterChange Server がビジネス・オブジェクトを認識するためには、まずビジネス・オブジェクト定義 と呼ばれるテンプレートを探し出し、ビジネス・オブジェクト内の情報の構造を記述します。ビジネス・オブジェクト定義内の情報はそれぞれ、属性に保持されています。そこで、メッセージ内で情報を表現するためのビジネス・オブジェクト定義を作成する必要があります。ビジネス・オブジェクト定義を作成するには、Business Object Designer ツールを使用します。

注: Business Object Designer は、WebSphere InterChange Server 製品と WebSphere Business Integration Adapter 製品のどちらにも含まれています。このツールの使用方法について詳しくは、「ビジネス・オブジェクト開発ガイド」を参照してください。

InterChange Server は、以下の情報のためにビジネス・オブジェクトを使用します。

- 『文書用ビジネス・オブジェクト』
- 129 ページの『構成情報用ビジネス・オブジェクト』

**文書用ビジネス・オブジェクト:** WebSphere Partner Gateway 文書またはメッセージのペイロードを保持するには、ペイロード・ビジネス・オブジェクト を表現するビジネス・オブジェクト定義を定義する必要があります。このビジネス・オブジェクト定義は、アダプターが文書を InterChange Server に (またはその逆方向に) 転送するペイロード・ビジネス・オブジェクトの形式になっています。このセクションでは、ペイロード・ビジネス・オブジェクトに関する次の内容について説明します。

- 『ビジネス・オブジェクトの構造』
- 128 ページの『ビジネス・オブジェクトの変換』
- 129 ページの『InterChange Server の用語』

#### ビジネス・オブジェクトの構造:

ペイロード・ビジネス・オブジェクトは、転送する文書内の個々の情報に、その関連ペイロード・ビジネス・オブジェクト定義の属性が必ず存在するように設計する必要があります。表 41 に示すように、ペイロード・ビジネス・オブジェクトの内容は、文書の構造と、文書が使用するパッケージ化のタイプによって異なります。

表 41. パッケージ化とペイロード・ビジネス・オブジェクトの構造との関係

文書の構造	パッケージ化のタイプ	ペイロード・ビジネス・オブジェクト定義
ペイロードのみ	なし	文書のペイロード情報を保持します。
ペイロードのみ	バックエンド統合	次のデータを保持します。 <ul style="list-style-type: none"> <li>• 文書のペイロード情報</li> <li>• トランスポート・レベルのヘッダー</li> </ul>
ペイロードと添付ファイル	なし	適用されません。文書に添付ファイルが含まれる場合は、バックエンド統合パッケージ化を使用する必要があります。
ペイロードと添付ファイル  この文書には、トランスポート・エンベロープと呼ばれる XML ラッパーが格納されています。ここでは、ペイロードと添付ファイルの両方がラップされます。	バックエンド統合	次のデータを保持します。 <ul style="list-style-type: none"> <li>• 文書のペイロード情報</li> <li>• トランスポート・レベルのヘッダー</li> <li>• 添付ファイル・コンテナ。ここには、添付ファイルのデータと任意の添付ファイル・ビジネス・オブジェクトが保持されます。</li> </ul> トランスポート・エンベロープを処理するには、Attachment Data Handler と呼ばれる WebSphere Partner Gateway 付属のデータ・ハンドラーが必要です。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

ペイロード・ビジネス・オブジェクトは、WebSphere Partner Gateway との統合に使用される特定のアダプターの要件に従って設計する必要があります。表 42 では、特定のトランスポート・プロトコルを通じて転送するペイロード・ビジネス・オブジェクトについて、作成方法を詳細に説明している参照先を示します。

表 42. さまざまなトランスポート・プロトコルに応じたペイロード・ビジネス・オブジェクトの作成

トランスポート・プロトコル	注記および制約事項	詳細の参照先
HTTP		158 ページの『HTTP を使用する ICS のビジネス・オブジェクト定義の作成』
JMS	文書がバックエンド統合パッケージ化を使用する場合	179 ページの『JMS 用のビジネス・オブジェクト定義の作成』
すべて	文書に添付ファイルがある場合	145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』

### ビジネス・オブジェクトの変換:

通常、アダプターは、データ・ハンドラーを使用して文書の形式とそのビジネス・オブジェクト表現との変換を実行します。このデータ・ハンドラーのことをペイロード・データ・ハンドラーといいます。ペイロードのコンテンツ・タイプに対して適切なデータ・ハンドラーを呼び出すには、アダプターを構成する必要があります。通常、WebSphere Business Integration Data Handler for XML は XML メッセージとビジネス・オブジェクト間の変換を実行するため、これをペイロード・データ・ハンドラーとして構成します。ただし、対応するデータ・ハンドラーが WebSphere Business Integration Server によって提供されていないメッセージ・フォーマットに対しては、カスタムのデータ・ハンドラーを作成できます。

**注:** XML メッセージを処理する場合は、必ず WebSphere Business Integration Data Handler for XML バージョン 2.3.1 以降を使用してください。cXML メッセージの場合は、Data Handler for XML バージョン 2.4.1 以降を使用してください。

使用しているペイロード・データ・ハンドラーが、使用しているトランスポート・プロトコルが必要とする子メタオブジェクトを無視できることを確認する必要があります。データ・ハンドラー (WebSphere Business Integration に付属するものであってもカスタム・データ・ハンドラーであっても) を使用する前に、子メタオブジェクトをサポートしていることを確認してください。トランスポート・プロトコルの適切なセクションで、ビジネス・オブジェクトのアプリケーション固有情報に存在する `cw_mo_label` タグのセクションを参照してください (表 42 を参照)。

ペイロードの変換に使用するデータ・ハンドラーを指示するには、以下の手順を実行する必要があります。

1. データ・ハンドラーがペイロードを変換するためにサポートしなければならない MIME タイプを示し、この MIME タイプを処理可能なデータ・ハンドラーを探し出します。

WebSphere Business Integration Adapter 資料セット内の「データ・ハンドラー・ガイド」では、IBM が提供しているデータ・ハンドラーについて説明しています。これらのデータ・ハンドラーが機能しない場合は、カスタム・データ・ハンドラーを作成することもできます。

2. Business Object Designer 内で、使用する必要のあるデータ・ハンドラー用の子メタオブジェクトを作成します。IBM が提供するデータ・ハンドラーを使用する場合、子メタオブジェクトの構造については、「データ・ハンドラー・ガイド」を参照してください。
3. Business Object Designer 内で、サポートされる MIME タイプの属性を組み込むように、コネクタ用のトップレベル・データ・ハンドラー・メタオブジェクトを更新します。この属性の属性タイプは、データ・ハンドラーの子メタオブジェクトです。
4. Connector Configurator 内で、使用するデータ・ハンドラーを示す適切なコネクタ構成プロパティを設定します。
  - DataHandlerConfigM0 をトップレベル・データ・ハンドラー・メタオブジェクトの名前で設定し、DataHandlerMimeType プロパティをサポートされる MIME タイプで設定します。
  - DataHandlerClassName プロパティを、インスタンスを生成するデータ・ハンドラー・クラスの名称で設定します。

注: DataHandlerConfigM0 プロパティと DataHandlerMimeType プロパティの組み合わせ、または DataHandlerClassName プロパティのいずれかを設定します。

5. Connector Configurator 内で、サポートされるビジネス・オブジェクトのリストにトップレベル・データ・ハンドラー・メタオブジェクトを追加します。

#### **InterChange Server の用語:**

InterChange Server の場合、ペイロード・ビジネス・オブジェクトの名称は、次に示すように通信の方向によって決まります。

- WebSphere Partner Gateway は、文書を InterChange Server に送信 するときには、InterChange Server のイベント通知に参加しています。

この場合、データ・ビジネス・オブジェクトはイベント・ビジネス・オブジェクトと呼ばれ (単にイベントと呼ばれる場合もある)、ある外部パートナーで発生したイベントについて InterChange Server に通知します。

- WebSphere Partner Gateway は、文書を InterChange Server から受信 するときには、InterChange Server の要求処理に参加しています。

この場合、データ・ビジネス・オブジェクトは要求ビジネス・オブジェクトであり、ある外部パートナーからの情報を要求するために InterChange Server が送信したものです。InterChange Server が応答して、ハブ・コミュニティーへ応答ビジネス・オブジェクトを戻す場合があります。

#### **構成情報用ビジネス・オブジェクト:**

多くのアダプターに対しては、構成情報を保持するためにビジネス・オブジェクト定義を作成します。このようなビジネス・オブジェクトは多くの場合メタオブジェクトと呼ばれます。

表 43 では、特定のトランスポート・プロトコルを通じて転送するデータ・ビジネス・オブジェクトについて、作成方法を詳細に説明している参照先を示しています。

表 43. ビジネス・オブジェクトの作成について説明したセクション

トランスポート・プロトコル	関連コンポーネント	詳細の参照先
HTTP	Adapter for HTTP	163 ページの『ICS 用の HTTP トランスポート・レベル・ヘッダー情報の作成』
JMS	Adapter for JMS	180 ページの『JMS ヘッダー情報の作成』
すべて	Attachment Data Handler	140 ページの『Attachment 子メタオブジェクトの作成』

## コネクターの作成

使用するアダプター用のコネクタ・オブジェクトを作成する必要があります。このコネクタ・オブジェクトは、実行時におけるアダプターのインスタンスを表しています。コネクタ・オブジェクトは、InterChange Server の System Manager ツール内部で作成します。

注: コネクタ・オブジェクトの作成方法については、WebSphere InterChange Server 資料セット内の「システム・インプリメンテーション・ガイド」を参照してください。

表 44 には、コネクタ・オブジェクトの作成方法に関する情報の入手先が、使用しているトランスポート・プロトコルに基づいて要約されています。

表 44. さまざまなトランスポート・プロトコルに応じたコネクタ・オブジェクトの作成

トランスポート・プロトコル	アダプター	詳細の参照先
HTTP	Adapter for HTTP	166 ページの『HTTP コネクタ・オブジェクトの作成』
JMS	Adapter for JMS	184 ページの『JMS コネクタ・オブジェクトの作成』

## コラボレーションの作成

### このタスクについて

必要な実際のビジネス・プロセスを実行するのは InterChange Server 内部のコラボレーションです。そのため、WebSphere Partner Gateway 文書を正しく処理するには、InterChange Server 用の適切なコラボレーションが存在しなければなりません。実行時に適切なコラボレーションを使用可能にするには、以下の手順を実行してください。

1. 必要なビジネス・プロセスを提供するコラボレーション・テンプレートが存在することを確認します。
  - 現在そのようなコラボレーション・テンプレートが存在しない場合は、コラボレーション・テンプレートを作成してコンパイルします。
  - コラボレーション・テンプレートが存在する場合、コラボレーション・オブジェクトを構成するためには、テンプレートの使用方法を十分に理解していなければなりません。
2. コラボレーション・オブジェクトを作成し、以下のようにポートをバインドします。
  - 要求処理の場合、WebSphere Partner Gateway へ要求を送信する「送信先」ポートをアダプターに設定します。
  - イベント通知の場合、WebSphere Partner Gateway からイベント受信する「送信元」ポートをアダプターに設定します。

表 45 には、コネクタ・オブジェクトの作成方法に関する情報の入手先が、使用しているトランスポート・プロトコルに基づいて要約されています。

表 45. さまざまなトランスポート・プロトコルに応じたコラボレーションのバインディング

トランスポート・プロトコル	アダプター	詳細の参照先
HTTP	Adapter for HTTP	167 ページの『Adapter for HTTP と通信する場合のコラボレーションのバインド』
JMS	Adapter for JMS	184 ページの『Adapter for JMS と通信する場合のコラボレーションのバインド』

## プロジェクトの配置

ユーザー・プロジェクト内に、必要な実行時コンポーネントを定義する成果物が作成されると、ユーザー・プロジェクトを InterChange Server リポジトリに配置することが必要になります。System Manager 内でユーザー・プロジェクトを配置します。

## 添付ファイルのある文書の処理

WebSphere Partner Gateway には Attachment Data Handler が用意されており、これによって、WebSphere Partner Gateway と InterChange Server との間で送信される文書を処理できます。Attachment Data Handler は、XML トランスポート・エンベロープの内部で (添付ファイルのある場合もない場合も)、直列化形式とビジネス・オブジェクト表現との間で文書の形式を変換します。次のいずれかの場合は、Attachment Data Handler をペイロード・データ・ハンドラーとして構成してください。

- バックエンド統合パッケージ化の「エンベロープ・フラグ」が「はい」に設定されている場合。

このフラグを「はい」に設定すると、WebSphere Partner Gateway は、文書に添付ファイルがあるかどうかにかかわらず、その文書を必ず XML トランスポート・エンベロープでラップします。バックエンド統合パッケージ化のこのフラグを

「はい」に設定するのは、プロファイルの「B2B 機能」ページでの操作の一環としてです。詳細については、37 ページの『ペイロード』を参照してください。

- 処理の対象となる文書に添付ファイルが含まれる可能性がある場合。

文書に添付ファイルが含まれていると、WebSphere Partner Gateway は、この文書を XML トランスポート・エンベロープでラップします。文書フローには、ペイロードが 1 つあり、必要に応じて複数の添付ファイルが存在します。添付ファイルが含まれる文書を送信または受信する場合、ペイロード・ビジネス・オブジェクトには、添付ファイルの情報が格納されている必要があります。

**注:** Attachment Data Handler は、添付ファイルを含む SOAP 文書の場合には必要ありません。SOAP 文書の処理方法については、167 ページの『HTTP/S を通じた SOAP 文書の送信』を参照してください。

Attachment Data Handler が、WebSphere Business Integration Adapter から呼び出されます。

- WebSphere Partner Gateway および InterChange Server が HTTP トランスポート・プロトコルを使用している場合、Attachment Data Handler を呼び出すのは、Adapter for HTTP になります。
- WebSphere Partner Gateway および InterChange Server が JMS トランスポート・プロトコルを使用している場合、Attachment Data Handler を呼び出すのは、Adapter for JMS になります。

アダプターが XML トランスポート・エンベロープ内で文書を受信すると、Attachment Data Handler を呼び出して、この文書を適切なビジネス・オブジェクト表現に変換します。例えば、172 ページの図 35 には、文書の直列化形式を文書のビジネス・オブジェクト表現に変換するために Attachment Data Handler を呼び出している Adapter for JMS が示されています。反対に、アダプターは、XML トランスポート・エンベロープに格納された文書のビジネス・オブジェクト表現を受信すると、Attachment Data Handler を呼び出して、このビジネス・オブジェクト構造を、その適切な文書形式に変換します。例えば、174 ページの図 36 には、文書のビジネス・オブジェクト表現を文書の直列化形式に変換するために Attachment Data Handler を呼び出している Adapter for JMS が示されています。

このセクションでは、Attachment Data Handler に関する次の内容について説明します。

- 『Attachment Data Handler による変換の実行方法』
- 139 ページの『Attachment Data Handler の環境の設定』
- 140 ページの『Attachment Data Handler の構成』
- 145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』

## Attachment Data Handler による変換の実行方法

Attachment Data Handler は、XML トランスポート・エンベロープの構造を解釈して、内部に格納されているデータと、対応するビジネス・オブジェクト表現との間の変換を処理します。変換方法については、以下のセクションを参照してください。。

- 文書を InterChange Server に送信する場合は、133 ページの『文書からビジネス・オブジェクトへの変換方法』

- 文書を InterChange Server から受信する場合は、136 ページの『ビジネス・オブジェクトから文書への変換方法』

**文書からビジネス・オブジェクトへの変換方法:** WebSphere Partner Gateway は、文書を InterChange Server に送信する前に、その内容を XML トランスポート・エンベロープでラップするかどうかを判別する必要があります。WebSphere Partner Gateway がトランスポート・エンベロープを作成した場合、ペイロードおよび添付ファイルは Base64 方式でエンコードされます。WebSphere Partner Gateway は、次に、XML トランスポート・エンベロープを、適切なトランスポート・レベルのヘッダーが付いた適切なアダプターに変換します。このアダプターを構成して、Attachment Data Handler を呼び出すと、XML でラップされた文書内にあるペイロードや添付ファイルから、対応するビジネス・オブジェクト表現への変換を処理できます。

XML トランスポート・エンベロープでラップされた文書をそのビジネス・オブジェクト表現に変換するため、呼び出し側のエンティティは、Attachment Data Handler のインスタンスを生成して、それを (トランスポート・エンベロープ内にある) 文書に渡します。次に、Attachment Data Handler は、以下の手順を実行します。

1. データ・ハンドラーの子メタオブジェクト内で定義されたコンテンツ・タイプ・マップをロードします。

コンテンツ・タイプ・マップは、子メタオブジェクトの `ContentTypeMap_x` 構成プロパティで定義されます。子メタオブジェクトは、Attachment Data Handler の構成情報を含むビジネス・オブジェクトです。このビジネス・オブジェクト内の属性は、コンテンツ・タイプ・マップをコンテンツ・タイプに関連付けます。詳細については、140 ページの『Attachment 子メタオブジェクトの作成』を参照してください。

2. 文書を検査して、この文書が XML トランスポート・エンベロープでラップされているかどうかを調べます。

- Attachment Data Handler は、トランスポート・エンベロープを検出できなかった場合、このエンベロープ構造からペイロードを抽出する必要はありません。

文書には、ペイロードのみが格納されています。このペイロードは、Attachment Data Handler によって、その関連ビジネス・オブジェクト表現に変換する必要があります。詳細については、134 ページの『トランスポート・エンベロープのない文書の処理方法』を参照してください。

- Attachment Data Handler は、トランスポート・エンベロープを検出した場合、このエンベロープ構造からペイロードと添付ファイル (存在する場合) を抽出する必要があります。

文書には、ペイロードと、場合によりいくつかの添付ファイルが格納されています。したがって Attachment Data Handler は、ペイロードおよび添付ファイルを、その関連ビジネス・オブジェクト表現に変換する必要があります。詳細については、134 ページの『トランスポート・エンベロープ内の文書の処理方法』を参照してください。

3. 変換して得られたペイロード・ビジネス・オブジェクトを設定して、このビジネス・オブジェクトを呼び出し側のエンティティに戻します。

**トランスポート・エンベロープのない文書の処理方法:** Attachment Data Handler は、XML トランスポート・エンベロープ内に文書が格納されていないと判断した場合、このエンベロープ構造からペイロード・データを抽出する必要はありません。このため、Attachment Data Handler は、(その子メタオブジェクト内に定義されている) PayloadDataHandlerMimeType 構成プロパティを使用して、デフォルトのペイロード・データ・ハンドラーを示す MIME タイプを取得し、文書ペイロードのインスタンスを生成します。このデータ・ハンドラーは、ペイロード・データをこれに対応するペイロード・ビジネス・オブジェクトに変換し、得られたペイロード・ビジネス・オブジェクトをその呼び出し側のエンティティに戻します。

**トランスポート・エンベロープ内の文書の処理方法:**  
このタスクについて

Attachment Data Handler は、XML トランスポート・エンベロープ内に文書が格納されていると判断した場合、ペイロードと添付ファイルを処理するために、先にこのエンベロープ構造からペイロードと添付ファイルを抽出する必要があります。このため、Attachment Data Handler は、次の手順に従って文書の処理および変換を実行します。

1. トランスポート・エンベロープからペイロードおよび添付ファイルを抽出して、ペイロード・データをデコードします。

ペイロードは、<payload> XML タグ内に含まれています。各添付ファイルは、<attachment> XML タグで囲まれています。

2. コンテンツ・タイプ・マップを検索し、ペイロードのコンテンツ・タイプと一致するコンテンツ・タイプの有無を確認します。

Attachment Data Handler は、一致したコンテンツ・タイプ・マップに指定されている MIME タイプを使用して、データ・ハンドラーのインスタンスを生成します。このデータ・ハンドラーは、ペイロード・データをこれに対応するペイロード・ビジネス・オブジェクトに変換し、得られたビジネス・オブジェクトを Attachment Data Handler に戻します。

3. ペイロードに対してコンテンツ情報ビジネス・オブジェクトを作成します。

Attachment Data Handler は、ペイロード・ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報を調べ、属性名が cw\_mo\_bcg\_content\_info タグで指定される、コンテンツ情報ビジネス・オブジェクトの名前を決定します。次に、このコンテンツ情報ビジネス・オブジェクトのインスタンスを生成して、ペイロード・コンテンツ・タイプとエンコード方式の値を設定します。

4. ペイロードに対して添付ファイル・コンテナ・ビジネス・オブジェクトを作成します。

Attachment Data Handler は、ペイロード・ファイル・ビジネス・オブジェクトのビジネス・オブジェクト・レベルのアプリケーション固有情報を調べ、属性名が cw\_mo\_bcg\_attachment タグで指定される、添付ファイル・コンテナ・ビジネス・オブジェクトの名前を決定します。次に、添付ファイル・コンテナ・ビジネス・オブジェクトのインスタンスを生成して、ペイロード・ビジネス・オブジェクトの適切な属性に保管します。

`cw_mo_bcg_attachment` タグが存在しない (または空である) 場合は、文書に添付ファイルが格納されていないものとみなします。したがって、これ以上の処理ステップは必要ありません。Attachment Data Handler は、変換済みのペイロード・ビジネス・オブジェクトを戻します。

5. 添付ファイル・コンテナに対して、デフォルト添付ファイル・ビジネス・オブジェクトを作成します。

Attachment Data Handler は、添付ファイル・コンテナ・ビジネス・オブジェクトのビジネス・オブジェクト・レベルのアプリケーション固有情報を調べ、属性名が `cw_mo_bcg_default_attribute` タグで指定される、デフォルトの添付ファイル・ビジネス・オブジェクトの名前を決定します。次に、デフォルト添付ファイル・ビジネス・オブジェクトのインスタンスを生成して、添付ファイル・コンテナ・ビジネス・オブジェクトの適切な属性に保管します。

6. コンテンツ・タイプ・マップを検索し、添付ファイルのコンテンツ・タイプと一致するコンテンツ・タイプの有無を確認することにより、添付ファイルをビジネス・オブジェクトに変換する必要があるかどうかを調べます。

Attachment Data Handler は、添付ファイルからコンテンツ・タイプと文字セットのエンコード方式を調べて、コンテンツ・タイプ・マップ内に対応する項目があるかどうかを検査します。

- 対応するコンテンツ・タイプ・マップが見つからない場合、Attachment Data Handler は添付ファイル・データに対してビジネス・オブジェクトを作成しません。

このため Attachment Data Handler は、デフォルト添付ファイル・ビジネス・オブジェクトのインスタンスを生成し、コンテンツ情報ビジネス・オブジェクトの内部にコンテンツ・タイプとエンコード方式の値を設定して、Base64 方式でエンコードされた添付ファイル・データを (ストリングとして) `attachment` 属性に設定します。

Attachment Data Handler は、次に、デフォルト添付ファイル・ビジネス・オブジェクトのデータを添付ファイル・コンテナ・ビジネス・オブジェクトに取り込みます。

- コンテンツ・タイプ・マップが見つかった場合、Attachment Data Handler は、添付ファイルをビジネス・オブジェクトに変換する必要があるかどうかを次のように調べます。
  - 一致するコンテンツ・タイプ・マップの `ConvertAttachment` 構成プロパティが `false` の場合、Attachment Data Handler は、デフォルト添付ファイル・ビジネス・オブジェクトのインスタンスを生成し、コンテンツ情報ビジネス・オブジェクトの内部にコンテンツ・タイプとエンコード方式の値を設定して、Base64 方式でエンコードされた添付ファイル・データを (ストリングとして) `attachment` 属性に設定します。

Attachment Data Handler は、次に、デフォルト添付ファイル・ビジネス・オブジェクトのデータを添付ファイル・コンテナ・ビジネス・オブジェクトに取り込みます。

- 一致するコンテンツ・タイプ・マップの `ConvertAttachment` 構成プロパティが `true` の場合、Attachment Data Handler は、添付ファイル・データを

デコードし、データ・ハンドラーのインスタンスを生成して、添付ファイル・データを処理します。このデータ・ハンドラーは、デコードされたバイト・データを処理して、対応する添付ファイル・ビジネス・オブジェクトを戻します。

Attachment Data Handler は、次に添付ファイル・ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報を調べ、属性名が `cw_mo_bcg_content_info` タグで指定される、コンテンツ情報ビジネス・オブジェクトの名前を決定します。このタグが存在する場合、Attachment Data Handler は、添付ファイルに対してコンテンツ情報ビジネス・オブジェクトを作成し、添付ファイルのコンテンツ・タイプおよびエンコード方式の値を設定します。

最後に、Attachment Data Handler は、添付ファイル・ビジネス・オブジェクトのデータを添付ファイル・コンテナ・ビジネス・オブジェクトに取り込みます。

### ビジネス・オブジェクトから文書への変換方法: このタスクについて

WebSphere Partner Gateway が InterChange Server から文書を受信する前に、アダプターは、ペイロードのビジネス・オブジェクト表現と添付ファイルを XML トランスポート・エンベロープでラップするかどうかを調べる必要があります。

InterChange Server は、ビジネス・オブジェクトを適切なデータ・ハンドラーに送信し、ICS 互換コンポーネントが実際の変換に対処します。このデータ・ハンドラーは、Attachment Data Handler を呼び出し、ペイロードや添付ファイル・ビジネス・オブジェクトから、対応するペイロードや添付ファイルへの変換を処理するだけでなく、XML トランスポート・エンベロープの作成も処理するように構成できます。

添付ファイル付きのペイロード・ビジネス・オブジェクトを、そのトランスポート・エンベロープ表現に変換するため、呼び出し側のエンティティーは、Attachment Data Handler のインスタンスを生成して、それをペイロード・ビジネス・オブジェクトに渡します。Attachment Data Handler は以下の手順を実行します。

1. 構成メタオブジェクトで定義されたコンテンツ・タイプ・マップをロードします。

コンテンツ・タイプ・マップは、子メタオブジェクトの `ContentTypeMap_x` 構成プロパティーで定義されます。子メタオブジェクトは、Attachment Data Handler の構成情報を含むビジネス・オブジェクトです。このビジネス・オブジェクト内の属性は、コンテンツ・タイプ・マップをコンテンツ・タイプに関連付けます。詳細については、140 ページの『Attachment 子メタオブジェクトの作成』を参照してください。

2. ビジネス・オブジェクトを検査して、XML トランスポート・エンベロープを作成するかどうかを判別します。
  - Attachment Data Handler は、文書にトランスポート・エンベロープが必要であると判断しなかった場合、このエンベロープ構造でペイロードをラップする必要はありません。

文書には、ペイロードのみが格納されています。このペイロードは、Attachment Data Handler によって、その関連ビジネス・オブジェクト表現から作成する必要があります。Attachment Data Handler は、文書に対してトランスポート・エンベロープを作成する必要はありません。詳細については、『トランスポート・エンベロープのない文書の作成方法』を参照してください。

- Attachment Data Handler は、文書にトランスポート・エンベロープが必要であると判断した場合、このエンベロープ構造の内部にペイロードと添付ファイルをラップする必要があります。

文書には、ペイロードと、場合によりいくつかの添付ファイルが格納されています。したがって Attachment Data Handler は、ペイロード・ビジネス・オブジェクト表現をペイロードおよび 添付ファイルに変換し、これらのコンポーネントをトランスポート・エンベロープでラップする必要があります。詳細については、『トランスポート・エンベロープのある文書の作成方法』を参照してください。

3. 変換して得られたペイロードおよび添付ファイルのタグを WebSphere Partner Gateway 文書内に設定して、この文書呼び出し側のエンティティに戻します。

**トランスポート・エンベロープのない文書の作成方法:** Attachment Data Handler は、ペイロード・ビジネス・オブジェクトには XML トランスポート・エンベロープが必要ないと判断した場合、ペイロード・データをエンベロープ構造でラップする必要はありません。したがって、Attachment Data Handler は、デフォルトのペイロード・データ・ハンドラーを使用して、ペイロード・ビジネス・オブジェクトをこれに対応するペイロード文書に変換します。(Attachment Data Handler の子メタオブジェクト内に定義されている) PayloadDataHandlerMimeType 構成プロパティには、ペイロード・ビジネス・オブジェクトのインスタンスを生成するために、デフォルトのペイロード・データ・ハンドラーを示す MIME タイプが格納されています。このデータ・ハンドラーは、ペイロード・ビジネス・オブジェクトを引き数として受信し、結果として得られたペイロード文書をその呼び出し側のエンティティに戻します。

#### **トランスポート・エンベロープのある文書の作成方法: このタスクについて**

Attachment Data Handler は、ペイロード・ビジネス・オブジェクトに XML トランスポート・エンベロープが必要と判断した場合、ペイロードと添付ファイル文書をこのエンベロープ構造でラップする必要があります。このため、Attachment Data Handler は、次の手順に従ってビジネス・オブジェクトの処理および変換を実行します。

1. ペイロードのコンテンツ・タイプと文字セットのエンコード方式を取得します。

ペイロード・ビジネス・オブジェクトのビジネス・オブジェクト・レベルのアプリケーション固有情報にある `cw_mo_bcg_content_info` タグは、コンテンツ情報属性の名前を指定します。この属性には、ペイロードのコンテンツ・タイプとエンコード方式が格納されています。

注: コンテンツ情報属性が存在しない場合は、(Attachment Data Handler のメタオブジェクト内にある、PayloadDataHandlerMimeType 構成プロパティに格納されている MIME タイプで指定される) デフォルトのデータ・ハンドラーを使用して、ペイロード・ビジネス・オブジェクトを変換します。

2. コンテンツ・タイプ・マップを検索し、ペイロードのコンテンツ・タイプと一致するコンテンツ・タイプの有無を確認します。

Attachment Data Handler は、一致したコンテンツ・タイプ・マップに指定されている MIME タイプを使用して、ペイロード・データ・ハンドラーのインスタンスを生成します。このデータ・ハンドラーは、ペイロード・ビジネス・オブジェクトをこれに対応するペイロード文書に変換し、得られた文書を Attachment Data Handler に戻します。Attachment Data Handler は、ペイロード・データ・ハンドラーによって戻されたストリングを基に、Base64 方式を使用してバイト・データをエンコードし、結果を XML トランスポート・エンベロープのペイロード・タグに格納します。

3. ペイロード・ビジネス・オブジェクトから添付ファイル・コンテナを取得します。

添付ファイル・コンテナは、ペイロード・ビジネス・オブジェクトの添付ファイル・コンテナ属性に存在します。ペイロード・ビジネス・オブジェクトのビジネス・オブジェクト・レベルのアプリケーション固有情報には、`cw_mo_bcg_attachment` タグが格納されています。このタグは、添付ファイル・コンテナ属性を示しています。この属性には、添付ファイルが格納されていません。

`cw_mo_bcg_attachment` タグが存在しない (または空である) 場合は、文書に添付ファイルが格納されていないものとみなします。したがって、これ以上の処理ステップは必要ありません。Attachment Data Handler は、変換済みのペイロードをそのトランスポート・エンベロープに戻します。

4. 添付ファイルをビジネス・オブジェクトと単なるデータのどちらで表現するかを、添付ファイルごとに指定します。
  - 添付ファイルの内容が添付データの場合、添付ファイル・コンテナ・ビジネス・オブジェクトのビジネス・オブジェクト・レベルのアプリケーション固有情報には、`cw_mo_bcg_default_attribute` タグが格納されます。このタグは、デフォルトの添付ファイル属性を示しています。この属性には添付データが格納されています。このデータは、Attachment Data Handler によって検索され、Base64 方式でエンコードされたデータが抽出されて、結果が文書に格納されます。
  - 添付ファイルがビジネス・オブジェクトによって表現されると、その属性レベルのアプリケーション固有情報には `wbic_type` タグが格納され、ここには添付ファイル・ビジネス・オブジェクトが格納されていることが示されます。

Attachment Data Handler は、次の手順に従って、添付ファイル・ビジネス・オブジェクトを処理します。

- a. 添付ファイル属性の内容を検索して、添付ファイルのコンテンツ・タイプとエンコード方式を取得します。

添付ファイル・ビジネス・オブジェクトのビジネス・オブジェクト・レベルのアプリケーション固有情報には、`cw_mo_bcg_content_info` タグが格納されています。このタグは、コンテンツ情報属性を示しています。この属性には、この添付ファイルのコンテンツ・タイプとエンコード方式が格納されています。Attachment Data Handler は、このコンテンツ情報を文書の `attachment` タグに格納します。

- b. コンテンツ・タイプ・マップを検索し、添付ファイルのコンテンツ・タイプと一致するコンテンツ・タイプの有無を確認します。

Attachment Data Handler は、一致したコンテンツ・タイプ・マップの MIME タイプを使用して、データ・ハンドラーのインスタンスを生成します。このデータ・ハンドラーは、添付ファイル・ビジネス・オブジェクトをこれに対応する添付ファイル文書に変換し、得られた文書を (ストリングとして) Attachment Data Handler に戻します。

- c. エンコードした結果を文書の XML ラッパーの `attachment` タグに格納します。

Attachment Data Handler は、(文字セットが存在した場合は文字セットを使用して) 戻されたストリングからバイト・データを取得し、このバイト・データを Base64 方式でエンコードします。次に、この結果を `attachment` タグに格納します。

## Attachment Data Handler の環境の設定

WebSphere Partner Gateway に付属する Attachment Data Handler を使用するには、以下の手順を実行します。

- 『使用するスキーマの指定』
- 140 ページの『Attachment Data Handler の配置』
- 140 ページの『Attachment Data Handler の構成』

**使用するスキーマの指定:** Attachment Data Handler のデフォルト・スキーマを使用するか、バックエンド統合パッケージ化で `contentId` を渡すスキーマ (`wbipackaging_v1.1_ns.xsd`) を使用できます。

`wbipackaging_v1.1_ns.xsd` スキーマを使用するには、`bcg.properties` ファイルの `wbipackaging_version` プロパティ・ファイルを構成します。( `bcg.properties` ファイルについては、「管理者ガイド」を参照。) このプロパティは、以下のように指定されています。

```
wbipackaging_version=1.n
```

ここで、*n* は 0 または 1 です。このプロパティのデフォルト値は 1.0 です。

Attachment Data Handler のメタオブジェクトには、`wbipackaging_version` 属性 (値は 1.0 または 1.1) があります。1.1 を指定すると、Attachment Data Handler は、添付ファイルの `contentId` が含まれている XML メッセージを解析および生成します。

添付ファイルのコンテンツ ID を指定するために、エンコード・ビジネス・オブジェクトは `contentId` 属性を使用します。Attachment Data Handler は、ビジネス・オブジェクトから XML を生成するときに、この属性を使用して添付ファイルの

contentId タグを作成します。Attachment Data Handler は、XML からビジネス・オブジェクトを生成するときに、XML メッセージの contentId タグに指定されている値を使用してこの属性を設定します。

#### Attachment Data Handler の配置:

Attachment Data Handler および関連リポジトリ・ファイルは、WebSphere Partner Gateway インストール・メディア上の、表 46 にリストされたロケーションで提供されています。

表 46. Attachment Data Handler のコンポーネントのロケーション

コンポーネント	ロケーション
Attachment Data Handler	Integration/WBI/WICS/Attachment/ bcgwbiattachmentdh.jar
リポジトリ・ファイル	Integration/WBI/WICS/Attachment/ MO_DataHandler_DefaultAttachmentConfigV1.0.in  または  Integration/WBI/WICS/Attachment/ MO_DataHandler_DefaultAttachmentConfigV1.1.in

Web サーバーの資料に従って、ファイルを Web サーバーに配置します。

#### Attachment Data Handler のロケーションの指定: このタスクについて

WebSphere InterChange Server は、Attachment Data Handler を実行時にロードできるように、そのロケーションを認識している必要があります。Attachment Data Handler のロケーションを指定するには、以下の手順を実行します。

1. ICS 始動スクリプト start\_server.bat を編集します。このスクリプトは、InterChange Server 製品ディレクトリーの bin サブディレクトリーにあります (InterChange Server が稼動しているコンピューター上)。
2. このファイル内の CLASSPATH 変数に、Attachment Data Handler の JAR ファイル (bcgwbiattachmentdh.jar) を追加します。

#### Attachment Data Handler の構成

Attachment Data Handler を構成するには、構成ビジネス・オブジェクトを作成するための次の手順を実行します。

- 『Attachment 子メタオブジェクトの作成』
- 144 ページの『トップレベル・データ・ハンドラー・メタオブジェクトの更新』

**注:** Attachment Data Handler の添付ファイル関連ビジネス・オブジェクト定義も作成する必要があります。詳細については、145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』を参照してください。

**Attachment 子メタオブジェクトの作成:** Attachment Data Handler を構成するには、Attachment Data Handler が必要とするクラス名と構成プロパティを提供するための子メタオブジェクトを作成しなければなりません。このメタオブジェクトを作成するには、141 ページの表 47 に示す属性を含むビジネス・オブジェクト定義を

作成します。WebSphere Business Integration Toolset の一部である Business Object Designer を使用して、このビジネス・オブジェクト定義を作成します。

子メタオブジェクトは Attachment Data Handler が必要とするクラス名と構成プロパティを提供します。Business Object Designer ツール内で、ペイロード用の MIME タイプと、受信と思われる添付ファイルのタイプ用の MIME タイプを含む子メタオブジェクトを作成します。

子メタオブジェクトの属性を表 47 に示します。Attachment Data Handler の子メタオブジェクトの例は、143 ページの図 28 に示されています。

**注:** この章に示すサンプル・ビジネス・オブジェクトには、WebSphere InterChange Server には必要であるが Attachment Data Handler には使用されない標準属性 (ObjectEventId など) は含まれていません。

表 47. Attachment 子メタオブジェクトの構成プロパティ

属性名	説明
ClassName	クラス名 (必須)。次のデータ・ハンドラー・クラスを指しています。 <code>com.ibm.bcg.DataHandlers.AttachmentDataHandler</code>
ContentTypeMap_x	ペイロードのコンテンツ・タイプ・マップおよび XML ラッパーでの受信が予想される各種の添付ファイルのコンテンツ・タイプ・マップ。  詳細については、『コンテンツ・タイプ・マップ』を参照してください。
PayloadDataHandlerMimeType	関連の添付ファイルが存在しない ペイロードを処理する、デフォルトのデータ・ハンドラーを示すために使用する MIME タイプ。
wbipackaging_version	この属性の値は、1.0 または 1.1 です。1.1 を指定すると、Attachment Data Handler は、添付ファイルの contentId が含まれている XML メッセージを解析および生成します。

**重要:** 表 47 の属性に値を割り当てるには、属性のデフォルト値を設定してください。例えば、Attachment Data Handler がそのデフォルトのデータ・ハンドラーとして XML データ・ハンドラーを使用する場合は、PayloadDataHandlerMimeType 属性のデフォルト値を text/xml に設定してください。

**コンテンツ・タイプ・マップ:** コンテンツ・タイプ・マップを指定すると、関連のコンテンツ・タイプで形式が設定された情報を変換するために Attachment Data Handler が呼び出すデータ・ハンドラーが指定されます。例えば、ペイロードのコンテンツ・タイプが application/xml の場合、Attachment Data Handler は、ContentType 属性に値 application/xml が含まれているコンテンツ・タイプ・マップを検索します。一致するコンテンツ・タイプが見つからない場合、Attachment Data Handler は、関連の添付ファイルをビジネス・オブジェクトに変換しないものとみなします。

143 ページの表 49 に示すような属性レベルのアプリケーション固有情報により、これらのコンテンツ・タイプごとにコンテンツ・タイプ・マップを作成します。

コンテンツ・タイプ・マップを表す子メタオブジェクト内に属性を作成する場合は、以下の点を念頭においてください。

- コンテンツ・タイプ・マップ属性の名前の形式は、次のようになっています。

ContentTypeMap\_x

ここで x は、ビジネス・オブジェクト定義内のコンテンツ・タイプ・マップを一意的に示す整数です。

注: ContentTypeMap\_x 属性は、順序どおりに並べる必要があります。例えば、コンテンツ・タイプ・マップが 3 個ある場合、属性は ContentType\_1、ContentType\_2、および ContentType\_3 という名前にする必要があります。

- コンテンツ・タイプ・マップ属性のデフォルト値には、有効なタグの組み合わせが含まれていなければなりません。

表 48 に、この属性のデフォルト値に含めることのできるタグをリストします。

表 48. コンテンツ・タイプ・マップ属性のデフォルト値で有効なタグ

タグ名	説明	必須かどうか
ContentType	トランスポート・エンベロープの形になっている実際のコンテンツ・タイプ (text/xml など)。	はい
MimeType	関連のコンテンツ・タイプをビジネス・オブジェクトに変換するときに、データ・ハンドラーを示すために使用する MIME タイプ。MimeType を指定しないと、データ・ハンドラーは ContentType の値を使用して、データ・ハンドラーのインスタンスを生成します。	いいえ
CharSet	Attachment Data Handler がバイトをストリングに、またはストリングをバイトに変換するために使用する文字セットの名前 (UTF-8 など)。  CharSet を指定しない場合、Attachment Data Handler は以下の操作を実行します。 <ul style="list-style-type: none"> <li>• インバウンド・データの場合、base64 からメッセージをデコードして生成されたデータ・バイトが、ビジネス・オブジェクトへの変換に使用されます。</li> <li>• アウトバウンド・データの場合は、バイトを戻す (ストリングは戻さない) 子データ・ハンドラーのメソッドが呼び出されます。</li> </ul>	いいえ
ConvertAttachment	添付ファイルをビジネス・オブジェクトに変換すべきであるかどうかを示すブール値。デフォルトは false です。	いいえ

コンテンツ・タイプ・マップでは、エンコードの文字セットだけでなく、添付ファイルをビジネス・オブジェクトに変換するかどうかについても指定できます。子メタオブジェクト属性の説明および例については、140 ページの『Attachment 子メタオブジェクトの作成』を参照してください。

例えば、文書内に次のコンテンツ・タイプがある場合を考えてください。

- application/xml
- text/xml
- application/octet-stream

表 49. コンテンツ・タイプ・マップのサンプル

コンテンツ・タイプ	属性名	デフォルト値
text/xml	ContentType_1	ContentType=text/xml;MimeType=myxml; CharSet=UTF-8;ConvertAttachment=false;
application/xml	ContentType_2	ContentType=application/xml; MimeType=mynewxml;CharSet=UTF-16; ConvertAttachment=true;
application/octet-stream	ContentType_3	ContentType=application/octet-stream; MimeType=myoctet

**サンプル子メタオブジェクト:** WebSphere Partner Gateway には、次に示す InterChange Server リポジトリ入力ファイルが用意されています。ここでは、Attachment Data Handler の子メタオブジェクトのサンプルが格納されています。

```
ProductDir/Integration/WBI/WICS/Attachment/  
MO_DataHandler_DefaultAttachmentConfigV1.0.in
```

```
ProductDir/Integration/WBI/WICS/Attachment/  
MO_DataHandler_DefaultAttachmentConfigV1.1.in
```

ここで、*ProductDir* は、WebSphere Partner Gateway 製品のインストール先のディレクトリを表します。リポジトリ・ファイルは、関連付けられている子メタオブジェクトが *MO\_DataHandler\_DefaultAttachmentConfig* である 1 つの Attachment Data Handler を定義します。図 28 には、Attachment Data Handler の子メタオブジェクトのサンプルを示します。このメタオブジェクトは、*ContentTypeMap\_1* および *ContentTypeMap\_2* の 2 つのコンテンツ・タイプ・マップを定義しています。

MO_DataHandler_DefaultAttachmentConfig
Name = ClassName Default Value = com.ibm.bcg.DataHandlers. AttachmentDataHandler
Name = ContentTypeMap_1 Default Value = ContentType=application/xml; MimeType=text/xml;CharSet=UTF-8; ConvertAttachment=true;
Name = ContentTypeMap_2 Default Value = ContentType=text/xml; MimeType=text/xml;CharSet=UTF-8;
Name = PayloadDataHandlerMimeType Default Value = text/xml

図 28. Attachment Data Handler のサンプル子メタオブジェクト

*MO\_DataHandler\_DefaultAttachmentConfigV1.1.in* には、以下の追加属性が含まれています。

```
[Attribute]
  Name = WBIPackaging_Version
  Type = String
  MaxLength = 255
  IsKey = false
  IsForeignKey = false
  IsRequired = true
  DefaultValue = 1.0
  IsRequiredServerBound = false
[End]
```

## トップレベル・データ・ハンドラー・メタオブジェクトの更新: このタスクについて

WebSphere Business Integration Adapter (Adapter for JMS など) は、`MO_DataHandler_Default` メタオブジェクトを使用して、使用可能なデータ・ハンドラーを確認します。メタオブジェクトで、Attachment Data Handler への参照を追加します。

`MO_DataHandler_Default` メタオブジェクトに対し、以下の変更を加えます。

1. 属性名が Attachment Data Handler のインスタンスに関連した MIME タイプを示す属性を追加します。つまり、この MIME タイプが格納されている文書では、関連のデータ・ハンドラーがビジネス・オブジェクトへの変換を処理できません。

この属性の属性タイプは、Attachment Data Handler の子メタオブジェクト用のビジネス・オブジェクト定義です (140 ページの『Attachment 子メタオブジェクトの作成』 参照)。

2. サポートされている添付ファイル MIME タイプが、トップレベル・データ・ハンドラー・メタオブジェクトに存在しない場合は、これらの MIME タイプごとに属性を追加します。

これらの属性の属性タイプは、関連データ・ハンドラーの子メタオブジェクトになります。

例えば、Attachment Data Handler が 143 ページの図 28 に示すように構成されているものとし、145 ページの図 29 には、`wbic_attachment` MIME タイプと、`MO_DataHandler_DefaultAttachmentConfig` 子メタオブジェクトによって構成された Attachment Data Handler のインスタンスとを関連付ける属性を持つ、`MO_DataHandler_Default` メタオブジェクトを示します。このトップレベル・データ・ハンドラー・メタオブジェクトは、文書の MIME タイプ (`text/xml`) と XML データ・ハンドラーの子メタオブジェクトも関連付けます。

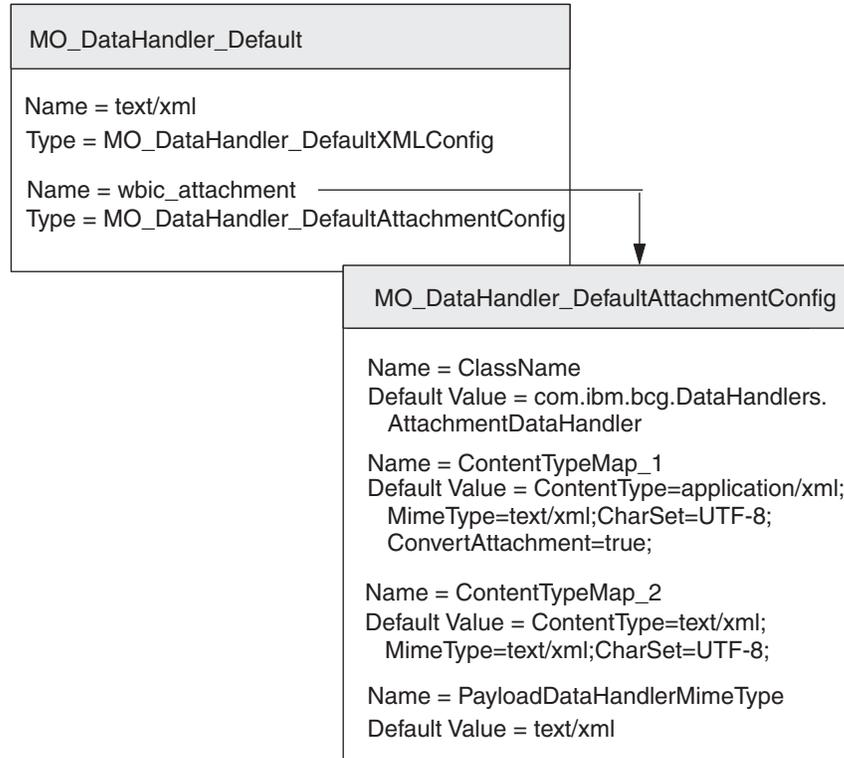


図 29. wbic\_attachment MIME タイプと Attachment Data Handler との関連付け

サポートが必要なコンテンツ・タイプの一意的組み合わせごとに、この処理を繰り返します。このためには、Attachment Data Handler インスタンスに関連付けられた MIME タイプが属性名であり、関連の子メタオブジェクトの名前がタイプである、適切なトップレベル・データ・ハンドラー・メタオブジェクト内に属性を追加します。構成済みの MIME タイプ (およびその子メタオブジェクト) がトップレベル・メタオブジェクト内に存在することも確認してください。

### 添付ファイル関連のビジネス・オブジェクト定義の作成

XML トランスポート・エンベロープでラップされている文書を送信または受信する場合、ペイロード・ビジネス・オブジェクトには、添付ファイルの情報が格納されている必要があります。文書フローには、ペイロードが 1 つあり、必要に応じて複数の添付ファイルが存在します。Attachment Data Handler は、この添付ファイル情報が添付ファイル関連のビジネス・オブジェクト内に存在すると想定します。そこで、この情報を表現するためのビジネス・オブジェクト定義を作成する必要があります。ビジネス・オブジェクト定義は、InterChange Server が使用する情報の形式です。Business Object Designer ツールを使用して、ビジネス・オブジェクト定義を作成します。

146 ページの図 30 には、XML トランスポート・エンベロープでラップされているペイロードのビジネス・オブジェクト構造を示します。

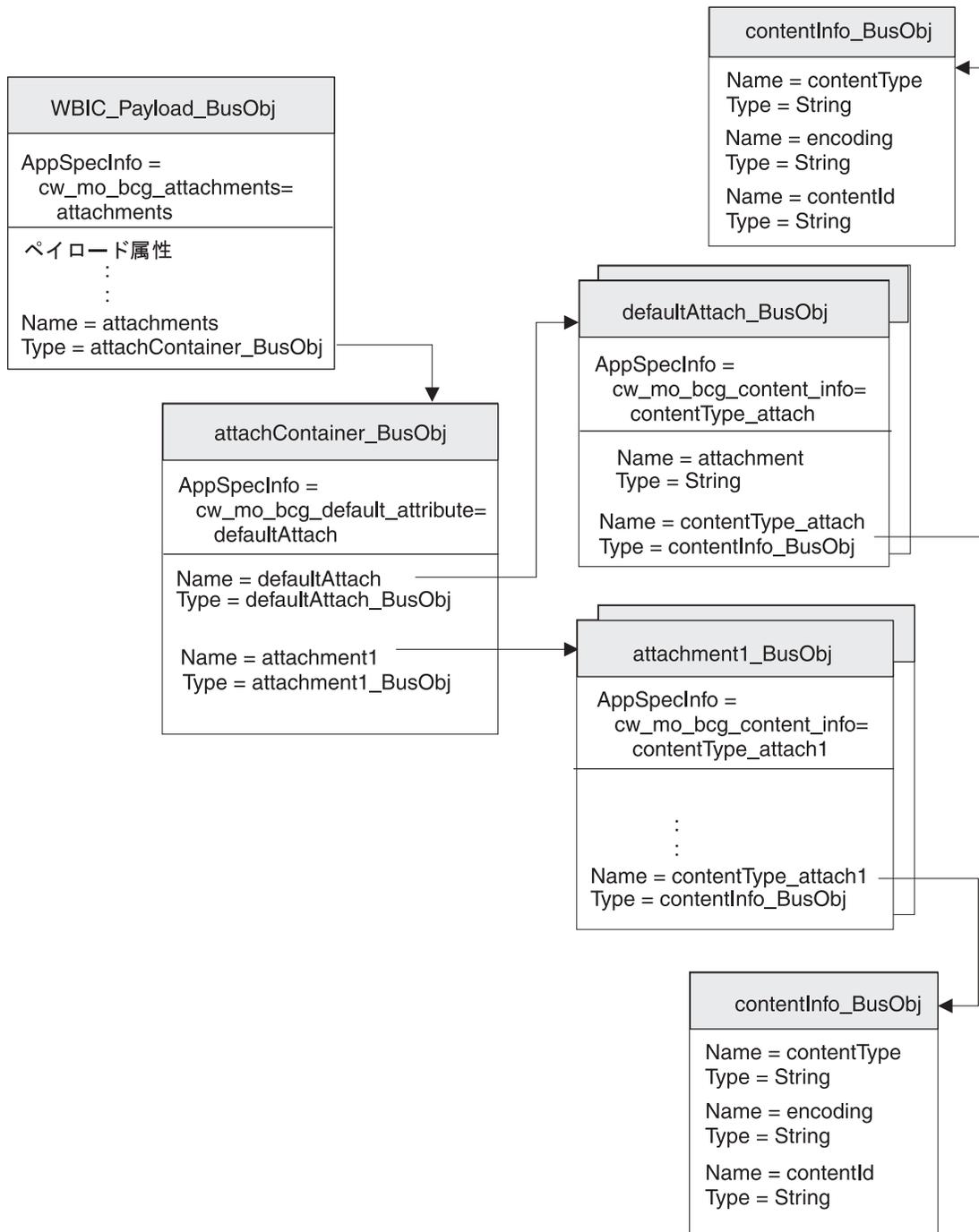


図 30. ペイロード・ビジネス・オブジェクトと添付ファイル・ビジネス・オブジェクトとの関係

図 30 に示すように、すべての添付ファイルは、添付ファイル・コンテナ・ビジネス・オブジェクトに格納されています。添付ファイルがある場合、ペイロード・ビジネス・オブジェクトには、添付ファイル・コンテナ・ビジネス・オブジェクトに対応する属性が存在します。

次の手順を実行して、ビジネス・オブジェクト構造に添付ファイル関連のビジネス・オブジェクトが含まれていることを確認します。

1. バックエンド統合パッケージ化に必要なコンテンツ・タイプ・エンコード・プロパティを保持するビジネス・オブジェクト定義を作成します。
2. 添付ファイルの種類ごとにビジネス・オブジェクト定義を作成します。
3. 添付ファイル・コンテナ・ビジネス・オブジェクトのビジネス・オブジェクト定義を作成します。
4. ペイロード・ビジネス・オブジェクトのビジネス・オブジェクト定義を変更します。

この手順の各ステップについて、以下のセクションで説明します。

### コンテンツ情報の表現:

関連のペイロードまたは添付ファイルのコンテンツ・タイプやエンコード方式を格納するには、コンテンツ情報ビジネス・オブジェクトを作成します。コンテンツ情報ビジネス・オブジェクト定義を作成するには、表 50 に示す属性を作成します。

表 50. コンテンツ情報ビジネス・オブジェクトの属性

属性	属性タイプ	説明	キー属性かどうか
contentType	ストリング	関連のペイロードまたは添付ファイルのコンテンツ・タイプ。	はい
エンコード	ストリング	関連のペイロードまたは添付ファイルの文字エンコード方式。	いいえ

146 ページの図 30 では、contentInfo\_BusObj ビジネス・オブジェクト定義に、添付ファイルのコンテンツ・タイプやエンコード方式の属性が格納されています。これらの属性のすべてに、関連プロトコル・ヘッダーの名前を指定する属性レベルのアプリケーション固有情報があります。例えば、x-aux-sender-id 属性には次のようなアプリケーション固有情報セットがあります。

```
name=x-aux-sender-id
```

コンテンツ情報ビジネス・オブジェクト定義を識別する名前は、自由に選択できます。これが Content Type Encoding Business Object タイプであるかどうかは、添付ファイル・ビジネス・オブジェクトのアプリケーション固有情報によって決まります。146 ページの図 30 に、contentType\_BusObj というコンテンツ・タイプ・エンコード・ビジネス・オブジェクトの例を示します。

### 添付ファイル・データの表現:

#### このタスクについて

ビジネス・オブジェクトに変換されない添付ファイル・データの場合は、デフォルト添付ファイル・ビジネス・オブジェクトを作成します。このビジネス・オブジェクトは、トランスポート・エンベロープを基にした Base64 方式でエンコードされたデータを格納する場合に有益です。

デフォルト添付ファイル・ビジネス・オブジェクト定義を作成するには、以下の手順を実行します。

1. 148 ページの表 51 に示す属性を作成します。

- コンテンツ情報ビジネス・オブジェクトを作成する場合は、デフォルト添付ファイル・ビジネス・オブジェクト定義のアプリケーション固有情報に、コンテンツ情報を含む属性を示す `cw_mo_bcg_content_info` タグを追加します。

`cw_mo_bcg_content_info` タグの形式は、次のようになっています。

```
cw_mo_bcg_content_info=contentInfoAttr
```

ここで、`contentInfoAttr` は、添付ファイル・コンテナ・ビジネス・オブジェクトを含む属性の名前です。

表 51. デフォルト添付ファイル・ビジネス・オブジェクトの属性

属性	属性タイプ	説明	キー属性かどうか
添付ファイル	ストリング	1 つの添付ファイル・データ。 注: この属性は、ビジネス・オブジェクト定義のキー属性です。	はい
コンテンツ情報を保持するための属性。	ビジネス・オブジェクト	コンテンツ情報ビジネス・オブジェクトを保持するためのオプションの属性。この属性により、添付ファイル・データのコンテンツ・タイプとエンコード方式がわかります。この属性のカーディナリティーは 1 つです。 注: この属性が存在しない場合、Attachment Data Handler はビジネス・オブジェクトに添付ファイル・データを設定しません。  コンテンツ情報ビジネス・オブジェクトの形式の詳細については、147 ページの『コンテンツ情報の表現』を参照してください。	いいえ

146 ページの図 30 では、コンテンツ・タイプとエンコード方式を保持するため、`defaultAttach_BusObj` ビジネス・オブジェクト定義に、コンテンツ情報ビジネス・オブジェクトなど、1 つの添付ファイル・データの属性が格納されています。このデフォルト添付ファイル・ビジネス・オブジェクトが表す 1 つの添付ファイル・データには、`contentType_attach` 属性によって表される、コンテンツ・タイプ・エンコード方式が存在します。したがって、デフォルト添付ファイル・ビジネス・オブジェクト定義には、そのビジネス・オブジェクト・レベルのアプリケーション固有情報に、次のタグが含まれます。

```
cw_mo_bcg_content_info=contentType_attach
```

#### 添付ファイルの表現:

##### このタスクについて

ビジネス・オブジェクトを変換する文書内の添付ファイルの種類ごとに、別個の添付ファイル・ビジネス・オブジェクト定義を作成する必要があります。添付ファイル・ビジネス・オブジェクト定義は、文書の添付ファイルの中の実際のデータを表します。添付ファイル・ビジネス・オブジェクト定義を作成するには、以下の手順を実行します。

- それぞれの添付ファイル・データごとに属性を作成します。

属性タイプには、ストリング (単純ないくつかのデータの場合) またはビジネス・オブジェクト定義 (複雑なデータの場合) があると考えられます。

2. 添付ファイルにコンテンツ・タイプ・エンコードが必要な場合は、次の手順を実行します。

- a. コンテンツ・タイプ・エンコードの属性を作成します。

この属性の属性タイプは、コンテンツ・タイプ・エンコードのビジネス・オブジェクト定義です (147 ページの『コンテンツ情報の表現』 参照)。

- b. コンテンツ・タイプ・エンコードが格納されている属性を識別するため、添付ファイル・ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報に、`cw_mo_bcg_content_info` タグを追加します。

`cw_mo_bcg_content_info` タグの形式は、次のようになっています。

```
cw_mo_bcg_content_info=contentTypeEncodingAttr
```

ここで、`contentTypeEncodingAttr` は、コンテンツ・タイプ・エンコード・ビジネス・オブジェクトを含む属性の名前です。

146 ページの図 30 では、ペイロード文書に、`attachment1_BusObj` ビジネス・オブジェクト定義によって表現される 1 つの添付ファイルがあります。この添付ファイルには、`contentTypeEncoding` 属性によって表現されるコンテンツ・タイプ・エンコードがあります。したがって、添付ファイル・ビジネス・オブジェクト定義では、そのビジネス・オブジェクト・レベルのアプリケーション固有情報に次のタグが含まれます。

```
cw_mo_bcg_content_info=contentTypeEncoding
```

#### 添付ファイル・コンテナの表現:

##### このタスクについて

添付ファイル・コンテナには、トランスポート・エンベロープ内にある文書の添付ファイルがすべて格納されています。InterChange Server の添付ファイル・コンテナを表現するには、添付ファイル・コンテナ・ビジネス・オブジェクトを作成する必要があります。添付ファイル・コンテナ・ビジネス・オブジェクト内の各属性は、1 つの添付ファイルを表現しています。

添付ファイル・コンテナ・ビジネス・オブジェクト定義を作成するには、以下の手順を実行します。

1. ビジネス・オブジェクトに変換する文書の添付ファイルごとに、属性を追加します。

これらの属性の属性タイプは、属性ごとに添付ファイル・ビジネス・オブジェクトに関連付けられます (148 ページの『添付ファイルの表現』を参照)。各属性には、複数のカーディナリティーがあります。

2. 属性を添付ファイルとして識別するため、各属性のアプリケーション固有情報に `wbic_type` タグを追加します。

`wbic_type` タグの形式は、次のようになっています。

```
wbic_type=Attachment
```

注: 添付ファイル属性には、複数のカーディナリティーを保持できます。

3. ビジネス・オブジェクトに変換できない 添付ファイル・データがペイロードに格納されている場合は、次の手順を実行します。

- a. デフォルト添付ファイル・ビジネス・オブジェクトに属性を追加します。

この属性の属性タイプは、デフォルト添付ファイル・ビジネス・オブジェクト定義です (147 ページの『添付ファイル・データの表現』参照)。これは、添付ファイル・コンテナ・ビジネス・オブジェクトのキー属性です。この属性は、アプリケーション固有情報に `wbic_type` タグを必要としません。

**注:** 添付ファイル・コンテナ・ビジネス・オブジェクトに格納できるデフォルトの添付ファイル属性は、1 つ だけです。ただし、この属性は複数カーディナリティーを持つことができます。

- b. 添付ファイル・データが格納されている属性を識別するため、添付ファイル・ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報に、`cw_mo_bcg_default_attribute` タグを追加します。

`cw_mo_bcg_default_attribute` タグの形式は、次のようになっています。

```
cw_mo_bcg_content_info=defaultAttachmentAttr
```

ここで、`defaultAttachmentAttr` は、デフォルト添付ファイル・ビジネス・オブジェクトを含む属性の名前です。

**重要:** デフォルトの添付ファイル属性が存在しない場合、Attachment Data Handler は、関連のコンテンツ・タイプ・マップのない添付ファイル、またはビジネス・オブジェクトに変換されない添付ファイルを変換できません。これらの添付ファイルは、ビジネス・オブジェクト表現への変換時に失われます。

146 ページの図 30 では、添付ファイル・コンテナが `attachContainer_BusObj` ビジネス・オブジェクト定義によって表現されています。この添付ファイル・コンテナ・ビジネス・オブジェクト定義には、次の属性があります。

- `attachment1` 属性は、文書の単一の添付ファイルであることを表しています。したがって、添付ファイル・コンテナ・ビジネス・オブジェクト定義では、その属性レベルのアプリケーション固有情報に次のタグが含まれます。

```
wbic_type=Attachment
```

この添付ファイルは、`attachment1_BusObj` ビジネス・オブジェクト定義によって表現されます。

- `defaultAttach` 属性は、ビジネス・オブジェクト表現への変換を必要としない添付ファイル・データを表します。したがって、添付ファイル・コンテナ・ビジネス・オブジェクト定義では、そのビジネス・オブジェクト・レベルのアプリケーション固有情報に次のタグが含まれます。

```
cw_mo_bcg_default_attribute=defaultAttach
```

#### ペイロード・ビジネス・オブジェクト定義の変更: このタスクについて

ペイロード・ビジネス・オブジェクト定義は、文書内の情報を表します。WebSphere Partner Gateway および InterChange Server 間で転送する各情報の属性が含まれてい

ます。ペイロード・ビジネス・オブジェクト定義の作成については、127 ページの『文書用ビジネス・オブジェクト』を参照してください。

添付ファイルが含まれる文書を送信または受信する場合、ペイロード・ビジネス・オブジェクトには、添付ファイルの情報が格納されている必要があります。文書フローには、ペイロードが 1 つあり、必要に応じて複数の添付ファイルが存在します。文書のペイロードに添付ファイルが含まれている場合、以下のようにペイロード・ビジネス・オブジェクト定義を変更する必要があります。

1. ペイロード・データを保持するための属性を作成します。

実際のペイロード・データが独立したビジネス・オブジェクト定義に格納されていると、使いやすい場合があります。この場合、トップレベルのペイロード・ビジネス・オブジェクトには、属性タイプが実際のペイロード・データのビジネス・オブジェクト定義であるペイロード・データの属性が格納されています。

2. 次のようにして添付ファイル・コンテナを追加します。

- a. 添付ファイル・コンテナを保持するための属性を追加します。

この属性の属性タイプは、添付ファイル・コンテナ・ビジネス・オブジェクト定義です (149 ページの『添付ファイル・コンテナの表現』を参照)。この属性のカーディナリティーは 1 つです。

- b. ペイロード・ビジネス・オブジェクト定義のアプリケーション固有情報に、添付ファイル・コンテナを含む属性を示す `cw_mo_bcg_attachment` タグを追加します。

この `cw_mo_bcg_attachment` タグの形式は、次のようになっています。

```
cw_mo_bcg_attachment=attachContainerAttr
```

ここで、`attachContainerAttr` は、添付ファイル・コンテナ・ビジネス・オブジェクトを含む属性の名前です。

3. 必要に応じて、ペイロードのコンテンツ・タイプを指定できます。Attachment Data Handler は、このコンテンツ・タイプを使用して、ペイロード・データを変換するためにインスタンスを生成するデータ・ハンドラーを判別します。

Attachment Data Handler は、一致するコンテンツ・タイプをコンテンツ・タイプ・マップ内に検出すると、このコンテンツ・タイプのデータ・ハンドラーについてインスタンスを生成します。

- a. ペイロードのコンテンツ・タイプとエンコード方式を保持するためのオプションの属性であるコンテンツ情報属性を追加します。この属性のカーディナリティーは 1 つです。

注: この属性が存在しない場合、Attachment Data Handler はデータ・ハンドラーを取得して、その子メタオブジェクトの `PayloadDataHandlerMimeType` 構成プロパティの設定を基にペイロードを変換します。

- b. ペイロード・ビジネス・オブジェクト定義のアプリケーション固有情報に、コンテンツ情報が格納されている属性を示す `cw_mo_bcg_content_info` タグを追加します。

`cw_mo_bcg_content_info` タグの形式は、次のようになっています。

```
cw_mo_bcg_attachment=contentInfoAttr
```

ここで、*contentInfoAttr* は、コンテンツ情報ビジネス・オブジェクトが格納されている属性の名前です。コンテンツ情報ビジネス・オブジェクトの形式の詳細については、147 ページの『コンテンツ情報の表現』を参照してください。

4. トランスポート・プロトコルで必要な構成属性を追加します。

例えば、JMS トランスポート・プロトコルを使用している場合、ペイロード・ビジネス・オブジェクト定義には、JMS 動的ビジネス・オブジェクトが格納されている必要があります。詳細については、トランスポート・プロトコルのサポートにおけるビジネス・オブジェクト定義の作成方法のセクションを参照してください。

---

## HTTP を通じての InterChange Server の統合

この章では、WebSphere Partner Gateway と WebSphere InterChange Server を HTTP トランスポート・プロトコルを通じて統合する方法について説明します。また、InterChange Server (ICS)、および HTTP を通じて通信する際に必要なアダプターの構成方法についても説明します。

**注:** HTTP を通じて InterChange Server と通信するように WebSphere Partner Gateway を構成する方法については、123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』を参照してください。InterChange Server の構成に関する一般情報については、126 ページの『InterChange Server の構成』を参照してください。

この章では、HTTP トランスポート・プロトコルを使用して WebSphere Partner Gateway と WebSphere InterChange Server との間で文書を送受信する方法を、以下のトピックにより説明します。

- 『ICS での HTTP トランスポート・プロトコルの使用』
- 167 ページの『HTTP/S を通じた SOAP 文書の送信』

## ICS での HTTP トランスポート・プロトコルの使用

WebSphere Partner Gateway は、HTTP トランスポート・プロトコルを通じて、WebSphere InterChange Server (ICS) との間で文書を送受信することができます。

**注:** HTTP トランスポート・プロトコルを通じて SOAP 文書を交換する場合は、167 ページの『HTTP/S を通じた SOAP 文書の送信』を参照してください。

このセクションでは、InterChange Server、および HTTP を通じて WebSphere Partner Gateway とともに使用する適切なアダプターの構成方法を、以下のトピックにより説明します。

- 153 ページの『HTTP トランスポートを通じた ICS への文書に必要なコンポーネント』
- 155 ページの『ICS での HTTP トランスポート用環境の設定』
- 158 ページの『HTTP を使用する ICS のビジネス・オブジェクト定義の作成』
- 166 ページの『HTTP を使用する ICS の成果物の作成』

## HTTP トランスポートを通じた ICS への文書に必要なコンポーネント

WebSphere Partner Gateway が HTTP トランスポート・プロトコルを使用して InterChange Server と通信するには、以下に示す 2 つのコンポーネントを構成する必要があります。表 52 に、これらの構成ステップの要約を示します。

表 52. WebSphere Partner Gateway および InterChange Server の構成

コンポーネント	バージョン	詳細の参照先
WebSphere Partner Gateway	6.2	123 ページの『HTTP トランスポート・プロトコルで文書を ICS に送信するための構成』  125 ページの『HTTP トランスポート・プロトコルで文書を ICS から受信するための構成』
WebSphere InterChange Server	4.3	166 ページの『HTTP を使用する ICS の成果物の作成』

また、HTTP トランスポート・プロトコルを使用して WebSphere Partner Gateway と InterChange Server の間で文書を送受信するには、表 53 にリストされているコンポーネントを使用する必要があります。

表 53. InterChange Server で HTTP を通じて文書を転送する際に必要なコンポーネント

コンポーネント	説明	注記および制約事項
WebSphere Business Integration Adapter for HTTP (Adapter for HTTP)	このアダプターによって、InterChange Server は、HTTP ストリームの形式でデータを送受信するアプリケーションとビジネス・オブジェクトを交換できます。	Adapter for HTTP のバージョン 4.2.1 を使用してください。
ペイロード・データ・ハンドラー	このデータ・ハンドラーは、文書のペイロードを文書形式 (通常 XML) とビジネス・オブジェクト表現の間で変換します。	このデータ・ハンドラーは必須であり、ペイロード文書の MIME タイプをサポートしていなければなりません。
Attachment Data Handler	このデータ・ハンドラーは、文書メッセージの添付文書进行处理します。	このデータ・ハンドラーは、文書に添付ファイルが含まれている場合のみ 必要です。

以降のセクションでは、WebSphere Partner Gateway と InterChange Server との間で HTTP トランスポート・プロトコルを通じて文書を送受信する際に、表 53 に示すコンポーネントがどのように機能しているのかについて説明します。

**HTTP を通じた ICS への文書の送信方法:** WebSphere Partner Gateway が HTTP トランスポート・プロトコルを使用して InterChange Server に文書を送信するには、WebSphere Partner Gateway が HTTP ストリームとして送信した文書を、Adapter for HTTP を使用して取得する必要があります。アダプターは、取得した文書を InterChange Server に送付します。154 ページの図 31 は、WebSphere Partner Gateway が HTTP トランスポート・プロトコルを通じて InterChange Server に文書

を送信する手順の概要を示したものです。

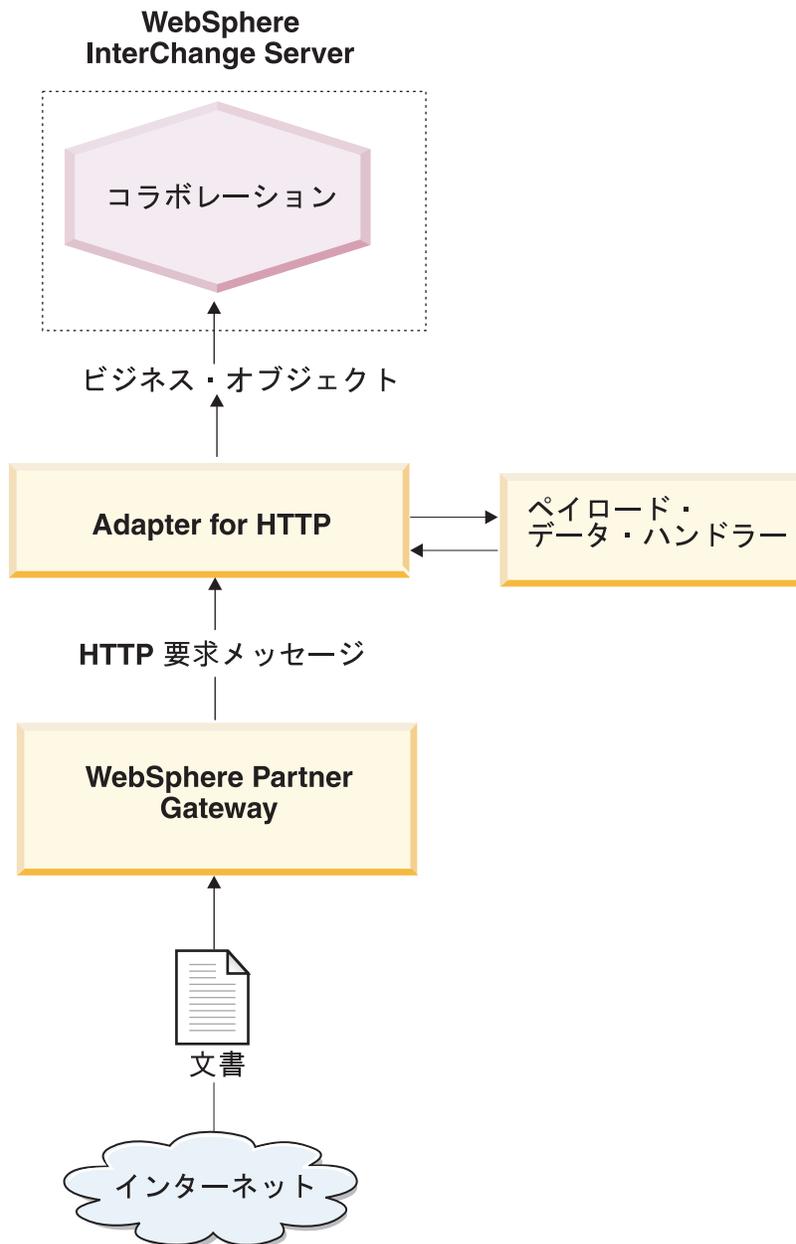


図 31. HTTP トラnsポート・プロトコルによる WebSphere Partner Gateway からコラボレーションへのメッセージ・フロー

**HTTP を通じた ICS から文書の受信方法:** WebSphere Partner Gateway が HTTP トラnsポート・プロトコルを使用して InterChange Server から文書を受信するには、Adapter for HTTP を使用する必要があります。Adapter for HTTP は、InterChange Server から受信したメッセージを、WebSphere Partner Gateway が取得できる HTTP ストリームとして送信します。155 ページの図 32 は、WebSphere Partner Gateway が HTTP トラnsポート・プロトコルを通じて InterChange Server から文書を受信する手順の概要を示したものです。

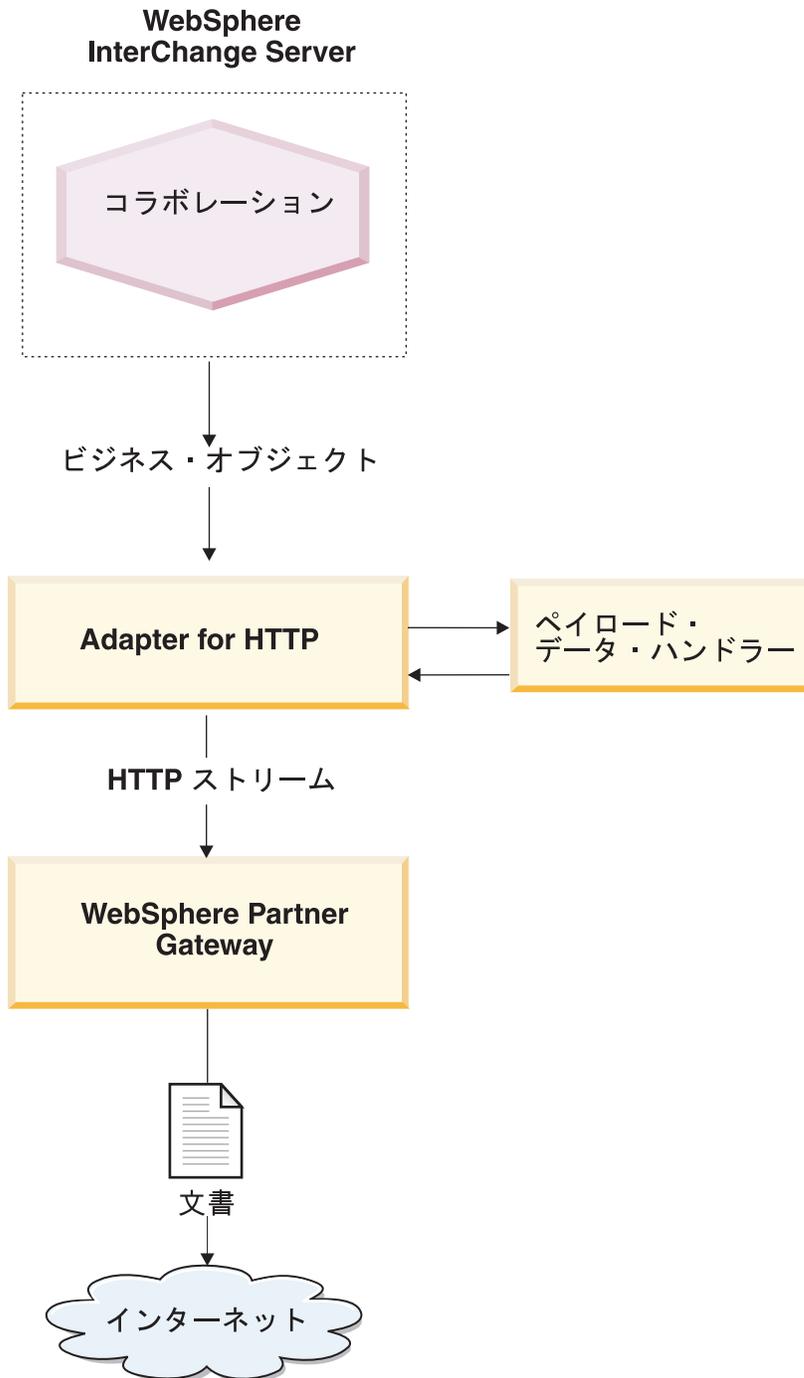


図 32. HTTP トラnsポート・プロトコルによるコラボレーションから WebSphere Partner Gateway へのメッセージ・フロー

### ICS での HTTP トラnsポート用環境の設定

InterChange Server との間で文書を送受信するにはアダプターおよびデータ・ハンドラーが必要となります。そのため、Adapter for HTTP に対して設定タスクと構成タスクを実行する必要があります。HTTP を通じて InterChange Server とともに使用

するように WebSphere Partner Gateway を構成する方法については、123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』を参照してください。

Adapter for HTTP により、WebSphere Partner Gateway は、HTTP メッセージ形式で InterChange Server と文書を交換できます。以下のような InterChange Server との対話をサポートします。

- 要求処理の場合、アダプターは InterChange Server から要求ビジネス・オブジェクトを受信し、それを HTTP ストリームに変換して、所定の URL に送信します。WebSphere Partner Gateway は、この URL から HTTP ストリームを受信します。
- イベント通知の場合、アダプターは、WebSphere Partner Gateway が文書を送信する所定の URL を listen します。文書を受信すると、アダプターは、(データ・ハンドラーを使用して) その文書をイベント・ビジネス・オブジェクトに変換し、InterChange Server に送信します。

**重要:** WebSphere Partner Gateway には WebSphere Business Integration Adapter for HTTP は組み込まれていません。この製品は別途入手する必要があります。「*Adapter for HTTP ユーザーズ・ガイド*」の説明に従ってインストールする必要があります。アダプターの資料を参照して、そのバージョンのアダプターに、ご使用の InterChange Server のバージョンとの互換性があることを確認します。

InterChange Server と通信するように Adapter for HTTP を構成したら、これらのセクションの手順に従って、WebSphere Partner Gateway からの HTTP メッセージを listen するようにこのアダプターを構成します。

**ペイロード・データ・ハンドラーの指定:** 155 ページの図 32 に示すように、Adapter for HTTP は、データ・ハンドラーを使用して、InterChange Server から受信したビジネス・オブジェクトを適切な HTTP ストリームに変換します。

**注:** Adapter for HTTP が呼び出すデータ・ハンドラーは、文書のペイロードを変換します。文書が XML トランスポート・エンベロープでラップされている (添付ファイルを含んでいる、またはエンベロープ・フラグが Yes に設定されている) 場合は、Attachment Data Handler をペイロード・データ・ハンドラーとして構成します。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

ペイロードを変換する際に使用するデータ・ハンドラーを指定するには、128 ページの『ビジネス・オブジェクトの変換』に示す手順を実行する必要があります。また、このペイロード・データ・ハンドラーを使用するように Adapter for HTTP を構成する必要もあります。ペイロード・データ・ハンドラーは、以下のいずれかの方法で設定できます。

- Connector Configurator を使用して、Adapter for HTTP がデータ・ハンドラーを識別する際に使用するトップレベル・データ・ハンドラー・メタオブジェクトの名前を、DataHandlerMetaObjectName コネクタ構成プロパティに指定します。アダプター用のサポートされるビジネス・オブジェクトのリストに、トップレベル・データ・ハンドラー・メタオブジェクトが含まれていることを確認してください。

- トップレベル・ビジネス・オブジェクトで、MIME タイプを保持する MimeType 属性を使用して、ペイロード・データ・ハンドラーを指定します。このビジネス・オブジェクトの詳細については、160 ページの『トップレベル・ビジネス・オブジェクト』を参照してください。

**プロトコル・ハンドラー・パッケージ名の構成:** Adapter for HTTP は、JavaProtocolHandlerPackages コネクタ構成プロパティを使用して、Java プロトコル・ハンドラー・パッケージの名前を識別します。WebSphere Partner Gateway と統合する場合は、以下に示すデフォルト値が JavaProtocolHandlerPackage プロパティに設定されていることを確認してください。

```
com.ibm.net.ssl.internal.www.protocol
```

### HTTP プロトコル・リスナーの構成:

Adapter for HTTP は、プロトコル・リスナーの構成に必要な情報を保持する階層構成プロパティをサポートします。トップレベル構成プロパティは、ProtocolListenerFramework と呼ばれます。このトップレベル・プロパティには、いくつかのレベルのサブプロパティが含まれます。Adapter for HTTP とともに使用するプロトコル・ハンドラーを構成するには、以下の手順に従って、ProtocolListener プロパティ内のプロパティを構成します。

1. 以下の構成プロパティの下のサブプロパティを使用して、プロトコル・リスナーを構成します。

```
ProtocolListenerFramework
  ProtocolListeners
    HttpListener1
```

プロトコル・リスナーを構成するには、表 54 に示すサブプロパティを設定します。

表 54. プロトコル・リスナーの構成

プロパティ	説明	値
プロトコル	プロトコル・リスナーのタイプ: • HTTP • HTTPS	http または https
Host	プロトコル・リスナーが listen する IP アドレス	WebSphere Partner Gateway が稼働しているローカル・コンピュータの IP アドレス
Port	プロトコル・リスナーが要求を listen するポート	8080

2. プロトコル・リスナーがサポートする URL 構成を、以下の構成プロパティの下のサブプロパティを使用して構成します。

```
ProtocolListenerFramework
  ProtocolListeners
    HttpListener1
      URLsConfiguration
        URL1
```

プロトコル・リスナーが受信する HTTP 要求の URI を ContextPath プロパティに設定します。

注: このディレクトリーは、WebSphere Partner Gateway がその受信する URI として指定するディレクトリーと同じにする必要があります。詳細については、123 ページの『HTTP トランスポート・プロトコルで文書を ICS に送信するための構成』を参照してください。

3. 文書に添付ファイルが含まれる場合は、以下の構成プロパティのサブプロパティを設定して、プロトコル・リスナーの変換を構成する必要があります。

```

ProtocolListenerFramework
  ProtocolListeners
    HttpListener1
      URLsConfiguration
        URL1
          TransformationRules
            TransformationRule1
  
```

プロトコル・リスナーの添付ファイル変換を構成するには、表 55 に示すサブプロパティを設定します。使用する Attachment Data Handler のインスタンスごとに変換ルールが 1 つ必要です。Attachment Data Handler の詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

表 55. プロトコル・リスナーの添付ファイル変換の構成

プロパティ	説明	値
ContentType	データ・ハンドラーで変換するデータのコンテンツ・タイプ	添付ファイル・データに関連付けられたコンテンツ・タイプ
MimeType	呼び出すデータ・ハンドラーを指定する際に使用する MIME タイプ	Attachment Data Handler のインスタンスに関連付けられた MIME タイプ
Charset	指定されたコンテンツ・タイプのデータを変換する際に使用する文字セット	添付ファイル・データの文字セット

これらのプロパティの詳細については、「Adapter for HTTP ユーザーズ・ガイド」を参照してください。

## HTTP を使用する ICS のビジネス・オブジェクト定義の作成

Adapter for HTTP が InterChange Server との間で文書を送受信する際は、ペイロード・ビジネス・オブジェクト形式が使用されます。WebSphere Partner Gateway の文書を送受信する際、Adapter for HTTP は、ペイロード・データ・ハンドラーを呼び出して、このビジネス・オブジェクトを処理します。

- 要求処理の場合、ペイロード・データ・ハンドラーは、要求ビジネス・オブジェクトをそれに対応する HTTP ストリームに変換します。
- イベント通知の場合、データ・ハンドラーは、HTTP ストリームをイベント・ビジネス・オブジェクトに変換します。

したがって、159 ページの表 56 に示すビジネス・オブジェクト定義を作成する際は、Adapter for HTTP で前提としているペイロード・ビジネス・オブジェクト構造を表すように、これらのビジネス・オブジェクト定義を作成する必要があります。

表 56. Adapter for HTTP のビジネス・オブジェクト定義

条件	ビジネス・オブジェクト定義	詳細の参照先
メッセージにパッケージ化なしまたはバックエンド統合パッケージ化を使用しており、同時に文書に添付ファイルが含まれていない場合	ペイロード・ビジネス・オブジェクト: ・ トップレベル・ビジネス・オブジェクト ・ 要求ビジネス・オブジェクト ・ 応答ビジネス・オブジェクト (オプション) ・ 障害ビジネス・オブジェクト (オプション)	『HTTP を使用する ICS のペイロード・ビジネス・オブジェクト構造の作成』
メッセージにバックエンド統合パッケージ化を使用している場合	メッセージ・ヘッダー情報を保持するビジネス・オブジェクトを、ペイロード・ビジネス・オブジェクトに追加します。 ・ 動的メタオブジェクト ・ HTTP プロパティ・ビジネス・オブジェクト	163 ページの『ICS 用の HTTP トランスポート・レベル・ヘッダー情報の作成』.
文書に添付ファイルが含まれる場合	添付ファイルを表現する追加ビジネス・オブジェクトを作成する必要があります。	145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』 145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』

### HTTP を使用する ICS のペイロード・ビジネス・オブジェクト構造の作成:

Adapter for HTTP は、以下のビジネス・オブジェクトから構成されるペイロード・ビジネス・オブジェクト構造を前提としています。

- ・ トップレベル・ビジネス・オブジェクト
- ・ 要求ビジネス・オブジェクト
- ・ 障害ビジネス・オブジェクト (オプション)
- ・ 応答ビジネス・オブジェクト (オプション)

160 ページの図 33 は、ペイロード・ビジネス・オブジェクト定義のビジネス・オブジェクト構造のサンプルを示しています。この構造は、HTTP トランスポート・プロトコルを使用する InterChange Server で使用されます。

注: このビジネス・オブジェクト構造の詳細な説明については、「Adapter for HTTP ユーザーズ・ガイド」を参照してください。

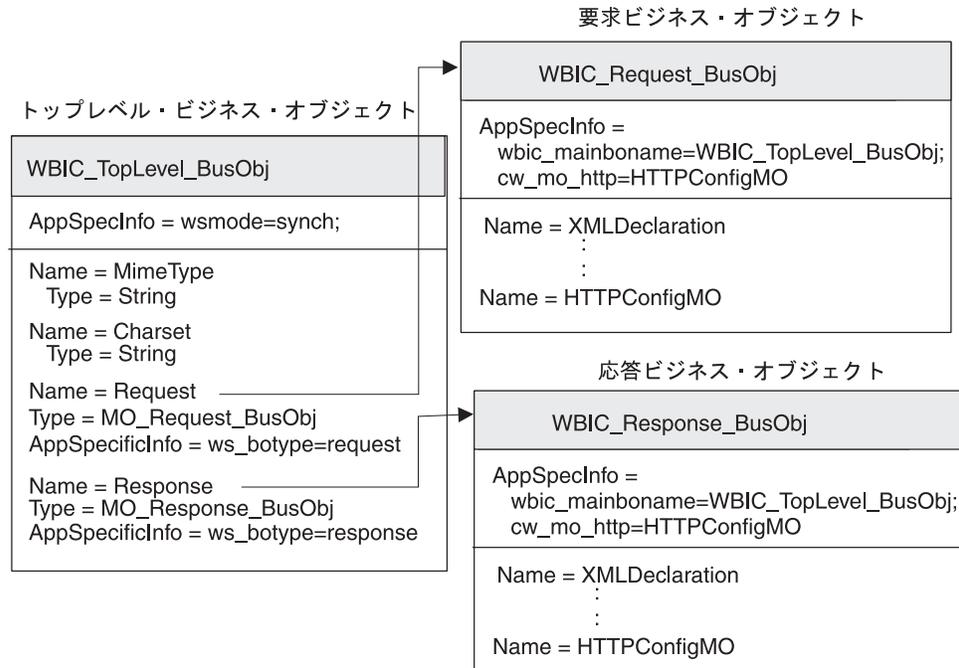


図 33. ICS 用の HTTP ペイロード・ビジネス・オブジェクトのビジネス・オブジェクト構造

**トップレベル・ビジネス・オブジェクト:**

トップレベル・ビジネス・オブジェクトは要求ビジネス・オブジェクトと応答ビジネス・オブジェクトのラッパーです。このビジネス・オブジェクト用にビジネス・オブジェクト定義を作成する必要があります。表 57 に、トップレベル・ビジネス・オブジェクト定義の属性をまとめています。

表 57. トップレベル・ビジネス・オブジェクトの属性

属性	属性タイプ	説明
MimeType	ストリング	URL に渡されるデータのコンテンツ・タイプとフォーマットを定義します。
Charset	ストリング	呼び出すデータ・ハンドラーを決定するために使用されます。
Request	ビジネス・オブジェクト	要求メッセージを表す子ビジネス・オブジェクト。このビジネス・オブジェクトの目的は、要求処理に参加しているか、イベント通知に参加しているかによって異なります。このビジネス・オブジェクトの構造について詳しくは、161 ページの『要求ビジネス・オブジェクト』を参照してください。

表 57. トップレベル・ビジネス・オブジェクトの属性 (続き)

属性	属性タイプ	説明
Response	ビジネス・オブジェクト	応答メッセージを表す子ビジネス・オブジェクト (応答があることを前提としている場合)。このビジネス・オブジェクトの目的は、要求処理に参加しているか、イベント通知に参加しているかによって異なります。このビジネス・オブジェクトの構造については、162 ページの『応答ビジネス・オブジェクト』を参照してください。

**注:** Adapter for HTTP を WebSphere Partner Gateway とともに使用するときは、障害ビジネス・オブジェクトをトップレベル・ビジネス・オブジェクトに含める必要はありません。

表 58 に、トップレベル・ビジネス・オブジェクト定義に組み込むことのできるアプリケーション固有情報をまとめています。

表 58. トップレベル・ビジネス・オブジェクト定義のアプリケーション固有情報

アプリケーション固有情報	タグ	説明
ビジネス・オブジェクト・レベル	ws_mode	対話が同期と非同期のどちらであることを定義します。
属性レベル	ws_botype	要求ビジネス・オブジェクトまたは応答ビジネス・オブジェクトをどの属性に格納するのかを定義します。

トップレベル・ビジネス・オブジェクトの構造とそのアプリケーション固有の情報の詳細については、「*Adapter for HTTP ユーザーズ・ガイド*」を参照してください。

**要求ビジネス・オブジェクト:** 要求ビジネス・オブジェクトには、URL に渡すデータが格納されています。要求ビジネス・オブジェクトは、HTTP 要求メッセージを表します。この要求ビジネス・オブジェクトの目的は、以下のように、参加している InterChange Server タスクの種類によって異なります。

- イベント通知 (InterChange Server に文書を送信する) の場合、要求ビジネス・オブジェクトには、WebSphere Partner Gateway からの要求メッセージが格納され、これがイベントとして InterChange Server に送信されます。
- 要求処理 (InterChange Server から文書を受信する) の場合、要求ビジネス・オブジェクトには、InterChange Server の WebSphere Partner Gateway に対する要求が格納されます。

**注:** トップレベル・ビジネス・オブジェクトは、子ビジネス・オブジェクトを要求ビジネス・オブジェクトと応答ビジネス・オブジェクトとして識別します。ただし、この構造は、要求処理とイベント通知のどちらにおいても使用されます。

要求ビジネス・オブジェクトの構造の基本的な説明については「*Adapter for HTTP ユーザーズ・ガイド*」を参照してください。WebSphere Partner Gateway とともに使用する場合は、要求ビジネス・オブジェクト定義の構造を 2 箇所カスタマイズする必要があります。

- WebSphere Partner Gateway から InterChange Server に送信する文書で「バックエンド統合パッケージ化」を使用する場合は、HTTP プロトコル構成メタオブジェクトを識別する特別な属性を要求ビジネス・オブジェクト定義に追加する必要があります。

この属性は、メッセージのトランスポート・レベル・ヘッダーの構成情報を提供します。詳細については、163 ページの『ICS 用の HTTP トランスポート・レベル・ヘッダー情報の作成』を参照してください。

- 要求ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報に、表 59 で示すタグを追加します。

表 59. 要求ビジネス・オブジェクトのアプリケーション固有情報内のタグ

アプリケーション固有情報のタグ	説明	必須かどうか
ws_tloname	トップレベル・ビジネス・オブジェクトの名前を示します。	ビジネス・オブジェクト定義がイベント通知に参加している場合のみ必要
cw_mo_http	HTTP トランスポート・レベル・ヘッダー・フィールドを含む HTTP プロトコル構成メタオブジェクトを指定します。詳細については、163 ページの『ICS 用の HTTP トランスポート・レベル・ヘッダー情報の作成』を参照してください。	バックエンド統合パッケージ化を使用している場合のみ必要

注: XML トランスポート・エンベロープでラップされた文書を Attachment Data Handler を使用して処理する場合は、添付ファイルを保持する要求ビジネス・オブジェクトを変更する必要があります。詳しくは、145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』を参照してください。

**応答ビジネス・オブジェクト:** 応答ビジネス・オブジェクトには、URL から受信するデータが格納されます。応答メッセージ内のさまざまな XML タグの属性が含まれています。この応答ビジネス・オブジェクトの目的は、以下のように、参加している InterChange Server タスクの種類によって異なります。

- イベント通知の場合、応答ビジネス・オブジェクトには、InterChange Server 内のコラボレーションから送信される応答メッセージが格納されます。
- 要求処理の場合、応答ビジネス・オブジェクトには、InterChange Server が送信した要求に対する応答として、WebSphere Partner Gateway からの情報が格納されません。

応答ビジネス・オブジェクトが送信されるのは、WebSphere Partner Gateway と InterChange Server との間で情報が同期的に交換されており、要求に対する応答としてビジネス応答が期待される場合に限られます。応答がイベント通知と要求処理のどちらの一部であるのかは関係ありません。

障害ビジネス・オブジェクトの構造の基本的な説明については「*Adapter for HTTP ユーザーズ・ガイド*」を参照してください。WebSphere Partner Gateway とともに使用する場合は、要求ビジネス・オブジェクト定義の構造をカスタマイズする必要があります。

- WebSphere Partner Gateway から InterChange Server に送信する文書で「バックエンド統合パッケージ化」を使用する場合は、HTTP プロトコル構成メタオブジェクトを識別する特別な属性を応答ビジネス・オブジェクト定義に追加する必要があります。

この属性は、メッセージのトランスポート・レベル・ヘッダーの構成情報を提供します。詳細については、『ICS 用の HTTP トランスポート・レベル・ヘッダー情報の作成』を参照してください。

- 応答ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報に対して、162 ページの表 59 に示すタグを追加します。
- トップレベル・ビジネス・オブジェクトで、応答ビジネス・オブジェクトに対応する属性の属性レベルのアプリケーション固有情報に `ws_botype` タグを追加します。

このタグの構文は次のようになっています。

```
ws_botype=response
```

WebSphere Partner Gateway と InterChange Server の間の交換が非同期 である場合、WebSphere Partner Gateway は応答があることを前提としないため、応答ビジネス・オブジェクトを作成する必要はありません。

**ICS 用の HTTP トランスポート・レベル・ヘッダー情報の作成:** HTTP トランスポート・プロトコルを通じてバックエンド統合パッケージ化を使用して文書を送信する場合、要求ビジネス・オブジェクトには、カスタム・トランスポート・レベル・ヘッダー情報が含まれている必要があります。Adapter for HTTP は、このカスタム・ヘッダー情報が動的メタオブジェクト内に存在することを前提としています。

164 ページの図 34 は、要求ビジネス・オブジェクトのビジネス・オブジェクト構造を示しています。これは、HTTP トランスポート・プロトコルを通じて「バックエンド統合パッケージ化」を使用する WebSphere Partner Gateway 文書を表しています。

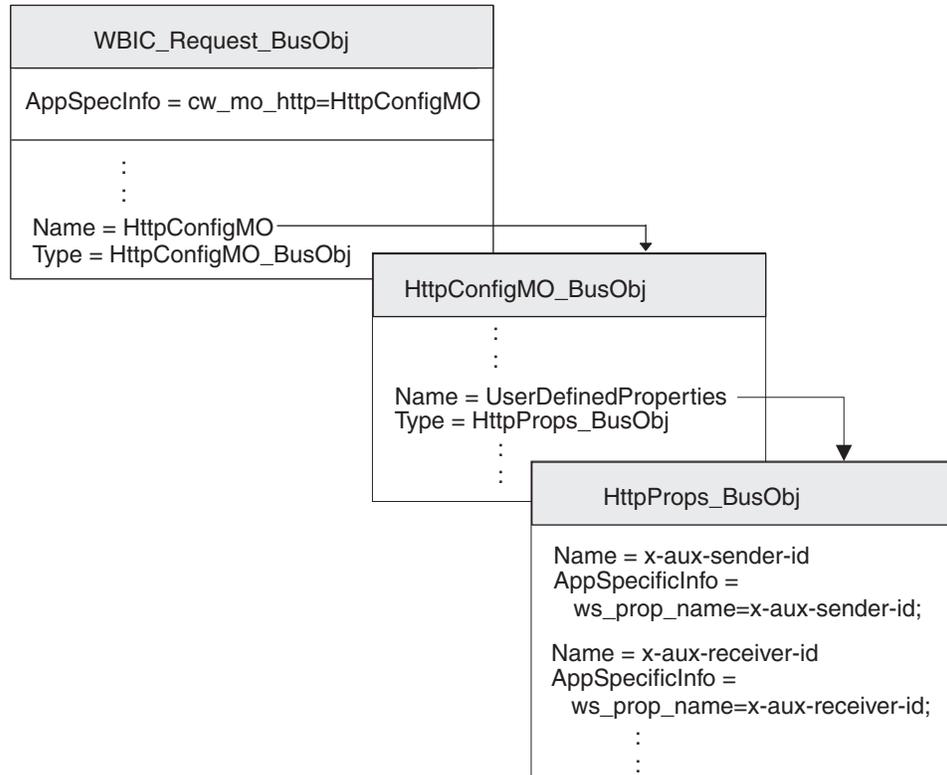


図 34. 要求ビジネス・オブジェクトと HTTP プロトコル構成メタオブジェクトとの関係

以下の手順を実行して、ビジネス・オブジェクト構造に HTTP プロトコル構成メタオブジェクトが含まれるようにします。

1. バックエンド統合パッケージ化に必要な HTTP プロパティを保持するビジネス・オブジェクト定義を作成します。
2. HTTP プロトコル構成メタオブジェクトのビジネス・オブジェクト定義を作成します。
3. 要求ビジネス・オブジェクトが HTTP プロトコル構成メタオブジェクトの属性を組み込むようにビジネス・オブジェクト定義を変更します。

この手順の各ステップについて、以下のセクションで説明します。

**ユーザー定義プロパティ・ビジネス・オブジェクトの作成:** Adapter for HTTP は HTTP プロトコル構成メタオブジェクトにカスタム・プロパティを保持するユーザー定義プロパティ・ビジネス・オブジェクトをサポートします。WebSphere Partner Gateway はこのビジネス・オブジェクトを使用して、バックエンド統合パッケージ化に必要な HTTP プロパティを保持します。また、コンテンツ・タイプ・ヘッダーを要求メッセージ内に設定することを指定する Content-Type 属性と、メッセージの長さをバイト単位で指定するコンテンツ長属性を含めることもできます。29 ページの表 8 では、有効なトランスポート・ヘッダー・フィールドのそれぞれについて説明しています。

HTTP ヘッダー・フィールド用のユーザー定義プロパティ・ビジネス・オブジェクト定義を作成するには、以下の手順を実行します。

1. 各トランスポート・ヘッダー・フィールドごとにビジネス・オブジェクト定義内部の属性を作成します。

すべての属性の属性タイプをストリングにする必要があります。属性には、(29ページの表8のヘッダー・フィールドの列に示すような) HTTP プロパティの名前をそのまま付けることができます。

2. HTTP プロパティ・ビジネス・オブジェクト内の各属性ごとに、関連属性の目的を示すアプリケーション固有情報を追加します。

この属性レベルのアプリケーション固有情報の形式は、次のようになっています。

```
ws_prop_name=HTTPproperty
```

ここで、*HTTPproperty* は、29ページの表8のヘッダー・フィールド列の値のいずれかです。

164ページの図34では、*HttpProps\_BusObj* ビジネス・オブジェクト定義にはさまざまなトランスポート・ヘッダー・フィールドの属性が含まれています。これらの属性のすべてに、関連プロトコル・ヘッダーの名前を指定する属性レベルのアプリケーション固有情報があります。例えば、*x-aux-sender-id* 属性には次のようなアプリケーション固有情報セットがあります。

```
ws_prop_name=x-aux-sender-id
```

**HTTP プロトコル構成メタオブジェクトの作成:** イベント通知の場合、要求、応答、または障害の各ビジネス・オブジェクトには、構成情報 (ヘッダー情報など) を保持する動的メタオブジェクト、*HTTP* プロトコル構成メタオブジェクトを格納できます。

HTTP プロトコル構成ビジネス・オブジェクトの構造の基本的な説明については「*Adapter for HTTP ユーザーズ・ガイド*」を参照してください。WebSphere Partner Gateway とともに使用する場合は、HTTP プロトコル構成ビジネス・オブジェクト定義の構造をカスタマイズする必要があります。

1. 必要なすべてのフィールドについて、ビジネス・オブジェクト定義内部の属性を作成します。

すべての属性の属性タイプをストリングにする必要があります。

**注:** HTTP プロトコル構成メタオブジェクト内の属性の完全なリストについては、「*Adapter for HTTP ユーザーズ・ガイド*」を参照してください。

2. *UserDefinedProperties* 属性をこのビジネス・オブジェクト定義に追加します。

この属性の属性タイプは、ユーザー定義プロパティ・ビジネス・オブジェクトのビジネス・オブジェクト定義です (164ページの『ユーザー定義プロパティ・ビジネス・オブジェクトの作成』を参照)。

例えば、164ページの図34では、*HttpConfigM0\_BusObj* ビジネス・オブジェクト定義には、属性タイプが *HttpProps\_BusObj* である *UserDefinedProperties* が含まれています。

**要求ビジネス・オブジェクト定義の変更:**

## このタスクについて

要求ビジネス・オブジェクト定義は、WebSphere Partner Gateway から要求された情報を表します。要求ビジネス・オブジェクトの作成方法については、161 ページの『要求ビジネス・オブジェクト』を参照してください。動的メタオブジェクトをパイロード・ビジネス・オブジェクト構造に取り込むには、要求ビジネス・オブジェクト定義に以下の変更を加える必要があります。

1. HTTP プロトコル構成メタオブジェクトを保持する属性を要求ビジネス・オブジェクト定義に追加します。

この属性の属性タイプは、HTTP プロトコル構成メタオブジェクトのビジネス・オブジェクト定義です (165 ページの『HTTP プロトコル構成メタオブジェクトの作成』を参照)。

2. HTTP プロトコル構成メタオブジェクトを含む属性を識別するために、要求ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報に `cw_mo_http` タグを追加します。

`cw_mo_http` タグの形式は、次のようになっています。

```
cw_mo_http=HttpConfigMetaObjAttr
```

ここで、`HttpConfigMetaObjAttr` は、プロトコル構成メタオブジェクトを保持する要求ビジネス・オブジェクト内の属性の名前です。

例えば、164 ページの図 34 では、`HttpConfigM0` という名前の属性が要求ビジネス・オブジェクト定義 `hub_HttpRequest_BusObj` に追加されています。この属性には、タイプ `HttpConfigM0_BusObj` の子ビジネス・オブジェクトである動的メタオブジェクトが含まれています。さらに、この動的メタオブジェクトを識別するため、要求ビジネス・オブジェクトのアプリケーション固有情報は、次のような `cw_mo_http` タグを組み込むように変更されています。

```
cw_mo_http=HttpConfigM0
```

## HTTP を使用する ICS の成果物の作成

HTTP トランスポート・プロトコルを通じて WebSphere Partner Gateway と通信するように InterChange Server を構成するには、表 60 に示す InterChange Server 成果物を作成する必要があります。

表 60. HTTP トランスポート・プロトコルを通じて ICS と通信するための成果物

ICS 成果物	目的	詳細の参照先
ビジネス・オブジェクト定義	文書を表します。	158 ページの『HTTP を使用する ICS のビジネス・オブジェクト定義の作成』
コネクタ・オブジェクト	実行時の Adapter for HTTP を表します。	『HTTP コネクタ・オブジェクトの作成』
コラボレーション・テンプレートとコラボレーション・オブジェクト	文書処理の際に InterChange Server が使用するビジネス・プロセスを表します。	167 ページの『Adapter for HTTP と通信する場合のコラボレーションのバインド』

### HTTP コネクタ・オブジェクトの作成:

## このタスクについて

Adapter for HTTP のインスタンスを実行時に取得するには、System Manager で以下の手順を実行する必要があります。

1. 以下のようにコネクタ・オブジェクトを作成します。
  - Adapter for HTTP のインスタンスを表現するコネクタ・オブジェクトを生成します。

**注:** Connector Configurator の「サポートされているビジネス・オブジェクト」タブで、Adapter for HTTP で使用するために作成したビジネス・オブジェクト定義をすべて指定してください。これらのビジネス・オブジェクト定義の説明については、158 ページの『HTTP を使用する ICS のビジネス・オブジェクト定義の作成』を参照してください。

- コラボレーションで必要であれば、Port Connector 用のコネクタ・ビジネス・オブジェクトを作成します。
2. コネクタ・オブジェクトを構成します。

WebSphere Partner Gateway で使用するために Adapter for HTTP コネクタ・オブジェクトを構成する方法については、155 ページの『ICS での HTTP トランスポート用環境の設定』を参照してください。

**Adapter for HTTP と通信する場合のコラボレーションのバインド:** 130 ページの『コラボレーションの作成』で説明したように、ビジネス・オブジェクトの受信場所や送信場所を InterChange Server で認識するには、コラボレーション・オブジェクトが実行時に存在しなければなりません。WebSphere Partner Gateway との間で情報を送受信する際に Adapter for HTTP を使用するコラボレーションに対してコラボレーション・オブジェクトを作成するときは、以下のようにコラボレーション・ポートをバインドする必要があります。

- 要求を処理する場合は、WebSphere Partner Gateway に要求を送信する「送信先」ポートを、Adapter for HTTP 用に作成したコネクタ・オブジェクトに設定する必要があります。すなわち、Adapter for HTTP は宛先 アダプターです。
- イベントの通知の場合は、WebSphere Partner Gateway からイベントを受信する「受信側」ポートを、Adapter for HTTP 用に作成したコネクタ・オブジェクトに設定する必要があります。すなわち、Adapter for HTTP はソース・アダプターです。

## HTTP/S を通じた SOAP 文書の送信

SOAP 文書は、HTTP/S を通じて交換される他のタイプの文書とは異なり、標準の Adapter for Web Services を使用します。Adapter for Web Services は、SOAP データ・ハンドラーを呼び出して SOAP メッセージをビジネス・オブジェクトに変換し、ビジネス・オブジェクトを SOAP メッセージに変換します。このセクションでは、HTTP トランスポート・プロトコルを通じて WebSphere Partner Gateway と WebSphere InterChange Server の間で SOAP 文書を送受信する方法について説明します。

注: WebSphere Partner Gateway と WebSphere InterChange Server との間で HTTP トランスポート・プロトコルを通じて SOAP 以外の文書を送受信するには、152 ページの『ICS での HTTP トランスポート・プロトコルの使用』を参照してください。

ビジネス・オブジェクトの構造および WSDL Object Discovery Agent (ODA) については、Adapter for Web Services の資料を参照してください。ODA は、受信側 Web サービスの情報を含む SOAP ビジネス・オブジェクトを生成するときに使用できる設計時ツールです。

「WebSphere Partner Gateway ハブ構成ガイド」で説明されているように、バックエンド・システムから Web サービス呼び出しを受信する受信側 (Web サービス受信側) と、外部パートナーから Web サービス呼び出しを受信する受信側 (外部 Web サービス受信側) を設定しておく必要があります。

## 送受信に必要なコンポーネント

HTTP トランスポート・プロトコルを使用して WebSphere Partner Gateway から InterChange Server へ SOAP 文書を送信するには表 61 にリストされているコンポーネントを使用します。

表 61. HTTP を通じて InterChange Server へ SOAP 文書を送信するために必要なコンポーネント

コンポーネント	説明	注記および制約事項
WebSphere Business Integration Adapter for Web Services	このアダプターによって、InterChange Server は、HTTP メッセージの形式でデータを送受信するアプリケーションとビジネス・オブジェクトを交換できます。	<ol style="list-style-type: none"> <li>このアダプターは、SOAP 以外の文書では使用できません。</li> <li>Adapter for Web Services 3.4.0 (またはそれ以降) を使用するよう to してください。「Adapter for Web Services ユーザーズ・ガイド」を参照して、そのレベルのアダプターに、ご使用のバージョンとの互換性があることを確認します。</li> </ol>

注: SOAP 文書に添付ファイルが含まれる場合、Attachment Data Handler を使用してそれらのファイルを処理する必要はありません。

## 外部パートナーによる Web サービスの起動方法

内部パートナーが提供する Web サービスとして公開されているコラボレーションに対し、外部パートナーが要求を送信すると、以下の手順が実行されます。

- 外部パートナーは、コラボレーション用に生成された WSDL 文書内で指定された宛先へ、SOAP 要求メッセージを送信します。WSDL に指定されているエンドポイントは、実際のエンドポイントではなく、WebSphere Partner Gateway の Web サービスの受信側 (URL) です。
- WebSphere Partner Gateway はメッセージを受信し、それを Adapter for Web Services へ送付します。
- Adapter for Web Services は、SOAP データ・ハンドラーに SOAP メッセージを送信して、SOAP メッセージをビジネス・オブジェクトに変換します。アダプターは、Web サービスとして公開されているコラボレーションを起動します。

4. これが要求/応答操作の場合、コラボレーションは SOAP 応答 (または障害) ビジネス・オブジェクトを戻します。
5. コラボレーションが SOAP 応答 (または障害) ビジネス・オブジェクトを戻した場合、Adapter for Web Services は SOAP データ・ハンドラーを呼び出して SOAP 応答 (または障害) ビジネス・オブジェクトを SOAP 応答メッセージに変換します。アダプターはその応答を WebSphere Partner Gateway に戻します。コラボレーションが SOAP 応答 (または障害) ビジネス・オブジェクトを戻さなかった場合、Adapter for Web Services は、適切な HTTP 応答状況コードを戻します。
6. WebSphere Partner Gateway はその応答を Web サービスに送付します。

## 内部パートナーによる Web サービスの起動方法

WebSphere Partner Gateway によって提供される Public WSDL を使用すると、WSDL ODA によってビジネス・オブジェクトを作成できます。内部パートナーが使用するために外部パートナーが Web サービスを提供する場合、内部パートナーが Web サービスを起動するために使用するパブリック URL には、次の照会ストリングが含まれていなければなりません。

?to=<External Partner Web Service Provider's business ID>

例えば次のアドレスは、Web サービスのプロバイダーが、ビジネス ID 123456789 のパートナーであることを、WebSphere Partner Gateway に知らせます。

http://<Hub\_IP\_address>/bcgreceiver/Receiver?to=123456789

WSDL ODA は、Web サービスのトップレベル・ビジネス・オブジェクトの URL 属性のデフォルト値に、照会ストリングを追加しなくなります。

コラボレーションが、外部パートナーの Web サービスを起動する要求を (Adapter for Web Services へ) 送信すると、以下の手順が実行されます。

1. コラボレーションは、アダプターにサービス呼び出し要求を送信します。アダプターは、SOAP データ・ハンドラーを呼び出して、ビジネス・オブジェクトを SOAP 要求メッセージに変換します。
2. アダプターは、WebSphere Partner Gateway 上の外部 Web サービス受信側 (URL) に SOAP メッセージを送信することによって、Web サービスを起動します。
3. WebSphere Partner Gateway は、プロキシとして機能し、宛先 (外部パートナー) の Web サービスに対応するエンドポイントに SOAP メッセージを送信します。これにより Web サービスが起動されます。
4. 起動された Web サービスは、SOAP 要求メッセージを受信し、要求された処理を実行します。
5. 起動された Web サービスは、SOAP 応答 (または障害) メッセージを送信します。片方向操作の場合は、適切な HTTP 状況コードが戻されます。
6. これが要求/応答 Web サービスである場合、WebSphere Partner Gateway は SOAP 応答 (または障害) メッセージをアダプターに送付します。アダプターは、データ・ハンドラーを呼び出して、そのメッセージを応答または障害ビジネス・オブジェクトに変換します。コネクタは SOAP 応答または障害ビジネス・オブジェクトをコラボレーションに戻します。

## JMS を通じての InterChange Server との統合

この章では、WebSphere Partner Gateway と WebSphere InterChange Server を JMS トランスポート・プロトコルを通じて統合する方法について説明します。また、InterChange Server、および JMS を通じて通信するために必要なアダプターの構成方法についても説明します。

注: JMS を通じて InterChange Server と通信するように WebSphere Partner Gateway を構成する方法については、123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』を参照してください。InterChange Server の構成に関する一般情報については、126 ページの『InterChange Server の構成』を参照してください。

この章では、JMS トランスポート・プロトコルを使用して WebSphere Partner Gateway と WebSphere InterChange Server との間で文書を送受信する方法を、以下のトピックにより説明します。

- 『JMS トランスポートを通じた文書に必要なコンポーネント』
- 176 ページの『JMS トランスポート用環境の設定』
- 179 ページの『JMS 用のビジネス・オブジェクト定義の作成』

### JMS トランスポートを通じた文書に必要なコンポーネント

WebSphere Partner Gateway と InterChange Server が JMS トランスポート・プロトコルを通じて通信するには、コンポーネントを JMS に対応するように構成する必要があります。表 62 に、これらの構成ステップの要約を示します。

表 62. JMS トランスポート・プロトコルを使用する場合の WebSphere Partner Gateway と InterChange Server の構成

コンポーネント	バージョン	詳細の参照先
WebSphere Partner Gateway	6.2	124 ページの『JMS トランスポート・プロトコルで文書を ICS に送信するための構成』  125 ページの『JMS トランスポート・プロトコルで文書を ICS から受信するための構成』
WebSphere InterChange Server	4.3	184 ページの『JMS 用の ICS 成果物の作成』

また、JMS トランスポート・プロトコルを通じて WebSphere Partner Gateway と InterChange Server の間で文書を送受信するには、171 ページの表 63 に示すコンポーネントも使用する必要があります。

表 63. JMS を通じて InterChange Server と文書をやり取りするために必要なコンポーネント

コンポーネント	説明	注記および制約事項
WebSphere Business Integration Adapter for JMS (Adapter for JMS)	このアダプターによって、InterChange Server は、JMS メッセージの形式でデータを送受信するアプリケーションとビジネス・オブジェクトを交換できます。Adapter for JMS と WebSphere Partner Gateway は、JMS キューを通じて通信します。	カスタム・ヘッダー・プロパティをサポートする Adapter for JMS バージョン 2.7.0 (またはそれ以降) を使用するよう to してください。アダプターの資料を参照して、そのバージョンのアダプターに、ご使用の InterChange Server のバージョンとの互換性があることを確認します。
ペイロード・データ・ハンドラー	このデータ・ハンドラーは、ペイロードを文書形式とビジネス・オブジェクト表現の間で変換します。	詳細については、177 ページの『ペイロード・データ・ハンドラーの指定』を参照してください。
Attachment Data Handler	このデータ・ハンドラーは、添付ファイルを含む文書をビジネス・オブジェクトに変換します。	このデータ・ハンドラーは、文書に添付ファイルが含まれている場合のみ 必要です。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

以降のセクションでは、WebSphere Partner Gateway と InterChange Server との間で JMS トランスポート・プロトコルを通じて文書を送受信する際に、表 63 に示すコンポーネントがどのように機能しているのかについて説明します。

### JMS トランスポートを使用した文書の送信方法

WebSphere Partner Gateway が JMS トランスポート・プロトコルを使用して文書を InterChange Server に送信できるようにするには、Adapter for JMS を使用して、WebSphere Partner Gateway が JMS キューに格納したメッセージを取り出します。次に、アダプターが、そのメッセージを InterChange Server に送付します。172 ページの図 35 では、WebSphere Partner Gateway が JMS トランスポート・プロトコルを通じて InterChange Server へ文書を送信する方法の概要を示します。

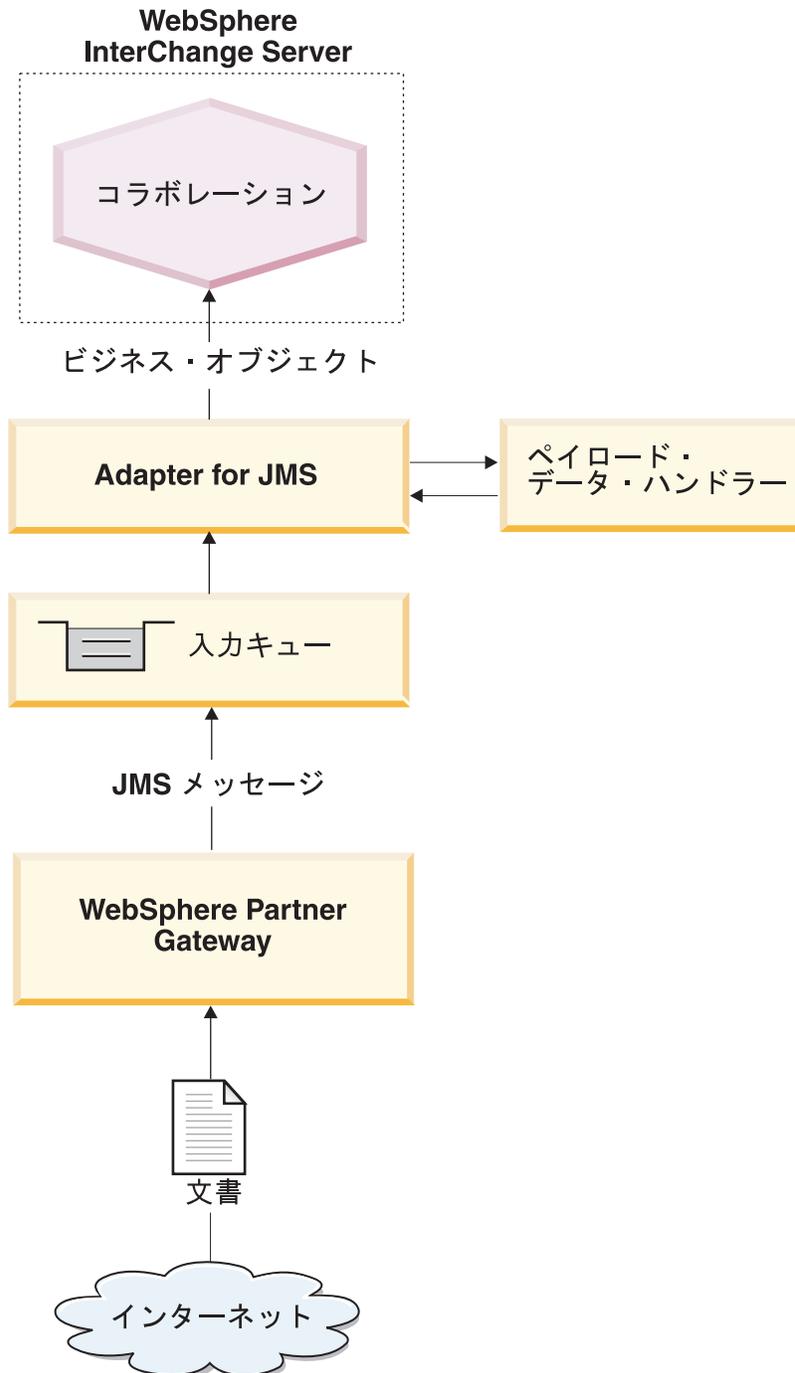


図 35. JMS トラnsポート・プロトコルによる WebSphere Partner Gateway からコラボレーションへのメッセージ・フロー

WebSphere Partner Gateway が、JMS トラnsポート・プロトコルを通じて InterChange Server 内部のコラボレーションへ文書を送信することによって、イベント通知に参加する手順を以下に説明します。

1. WebSphere Partner Gateway は、メッセージを JMS アウトバウンド・キューにポストします。

文書のパッケージ化タイプがバックエンド統合であれば、WebSphere Partner Gateway は、このメッセージ内にカスタム・プロパティを用意しています。JMS メッセージ・ヘッダーである JMSType が、ペイロードのコンテンツ・タイプとともに設定されます。

**注:** WebSphere Partner Gateway 内部で、WebSphere Partner Gateway からのメッセージの送信先であり、Adapter for JMS のポーリング対象である JMS キューを示す宛先を構成する必要があります。

2. Adapter for JMS は、入力キューのいずれかにメッセージを見つけた場合、メッセージを取り出します。

WebSphere Partner Gateway がアウトバウンド・キューとして使用する JMS キューは、Adapter for JMS が入力キューとして使用するキューと同じです。このキューの設定方法については、176 ページの『JMS キューの構成』を参照してください。Adapter for JMS の処理については、「*Adapter for JMS ユーザーズ・ガイド*」を参照してください。

3. Adapter for JMS は、その進行中キューにメッセージを移動します。
4. Adapter for JMS は、JMS メッセージの本文を抽出して、メッセージの本文とともにデータ・ハンドラーを呼び出します。このデータ・ハンドラーは、JMS メッセージの本文をビジネス・オブジェクトに変換します。

**注:** メッセージに添付ファイルがある場合、Attachment Data Handler をインストールし、それを呼び出して JMS メッセージの本文をビジネス・オブジェクトに変換するように Adapter for JMS を構成します。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

パッケージ化のタイプがバックエンド統合で、文書に添付ファイルが含まれる場合、構成済みデータ・ハンドラーがペイロードと添付ファイルの処理を担当します。

5. データ・ハンドラーは、ビジネス・オブジェクトを Adapter for JMS に戻します。

**注:** Attachment Data Handler が使用されていた場合、このビジネス・オブジェクトには添付ファイルだけでなく、ペイロードも含まれています。

6. Adapter for JMS が動的子メタオブジェクト (ビジネス・オブジェクト・レベルのアプリケーション固有情報内の `cw_mo_conn` を使用して指定される) を検出した場合、アダプターは、ビジネス・オブジェクトに存在するユーザー定義 JMS ヘッダーに JMS メッセージ内に存在するヘッダーを取り込みます。
7. Adapter for JMS は、サブスクリプション送達の一部として、InterChange Server へビジネス・オブジェクトを送達します。

## JMS トランスポートを使用した文書の受信方法

WebSphere Partner Gateway が JMS トランスポート・プロトコルを使用して文書を InterChange Server から受信できるようにするには、Adapter for JMS を使用します。アダプターは、InterChange Server から取り出したメッセージを JMS キューに格納し、それを WebSphere Partner Gateway が取り出します。174 ページの図 36 は、WebSphere Partner Gateway が JMS トランスポート・プロトコルを通じて

InterChange Server から文書を受信する手順の概要を示したものです。

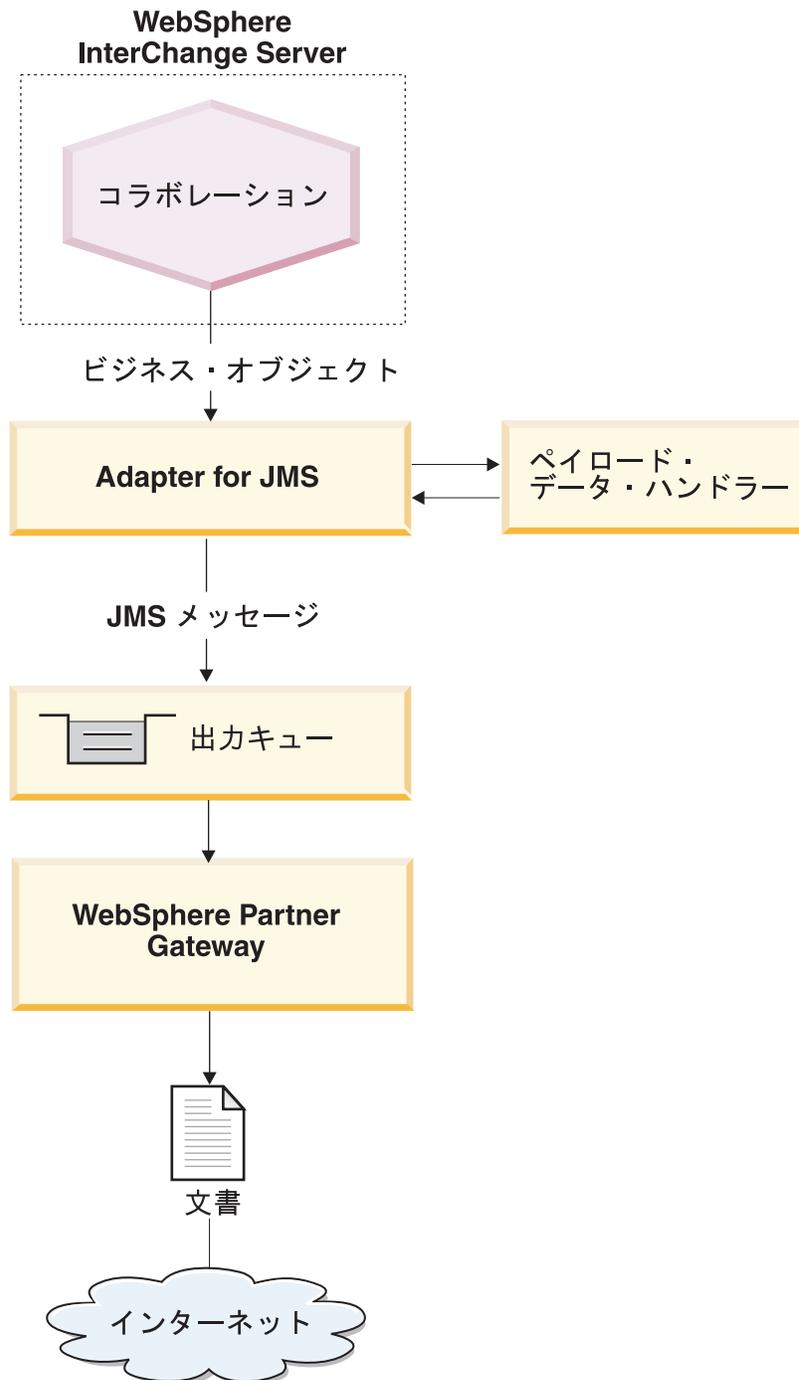


図 36. JMS トランスポート・プロトコルによるコラボレーションから WebSphere Partner Gateway へのメッセージ・フロー

WebSphere Partner Gateway が、JMS トランスポート・プロトコルを通じて InterChange Server 内部のコラボレーションから文書を受信することによって、要求処理に参加する手順を以下に説明します。

1. InterChange Server 内部のコラボレーションは、Adapter for JMS に対して要求ビジネス・オブジェクトを送信することにより、サービス呼び出しを行います。

要求ビジネス・オブジェクトには、JMS トランスポート・レベル・ヘッダー情報を含む動的メタオブジェクトを指すアプリケーション固有情報が含まれていません。WebSphere Partner Gateway はこの情報があることを前提としています。

2. Adapter for JMS はデータ・ハンドラーを使用して、コラボレーションが送信したビジネス・オブジェクトを JMS メッセージに変換します。

アダプターは、DataHandlerMimeType および DataHandlerConfigMO プロパティを読み出して、使用するデータ・ハンドラーを決定します。詳細については、177 ページの『ペイロード・データ・ハンドラーの指定』を参照してください。

**注:** 文書に添付ファイルがある場合は、Attachment Data Handler をインストールして、それを呼び出すように Adapter for JMS を構成します。これにより、要求ビジネス・オブジェクトが添付ファイル付き文書に変換されます。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

3. データ・ハンドラーはビジネス・オブジェクトをストリングに変換し、それを Adapter for JMS に戻します。
4. Adapter for JMS は、要求ビジネス・オブジェクトから、カスタム JMS プロパティの動的メタオブジェクトの名前を判別します。

アダプターは要求ビジネス・オブジェクトのアプリケーション固有情報を調べて cw\_mo\_conn タグを探します。このタグは動的メタオブジェクトを含む属性を示します。文書にバックエンド統合パッケージ化を使用している場合、この動的メタオブジェクト内でトランスポート・レベル・ヘッダー情報を指定することができます。

5. Adapter for JMS は、動的メタオブジェクトを調べて JMSProperties 属性を探します。

この属性にデータが取り込まれている場合、アダプターは要求文書にトランスポート・レベル・ヘッダー・フィールドを設定します。JMSProperties 属性内で、コンテンツ・タイプ標準 JMS ヘッダーを指定することもできます。詳細については、180 ページの『JMS ヘッダー情報の作成』を参照してください。

6. Adapter for JMS は、データ・ハンドラーによって戻されたストリングを使用して JMS メッセージを作成します。また、動的メタオブジェクトに定義されているように、カスタム・プロパティを設定します。
7. Adapter for JMS は、生成された要求メッセージを出力キューに送信します。

キューは、静的メタオブジェクトまたは動的メタオブジェクト内で指定することができます。キューの指定方法については、178 ページの『JMS キューの識別』を参照してください。WebSphere Partner Gateway は、この JMS キュー (受信側定義でインバウンド・キューとして構成されている) で listen します。

8. WebSphere Partner Gateway は、受信側で構成されているように、JMS インバウンド・キューからメッセージを受信します。

**注:** WebSphere Partner Gateway は、JMS を通じた InterChange Server との非同期の対話のみをサポートします。したがって、通常は応答を待つ必要はありません。外部パートナーまたは WebSphere Partner Gateway からの応答は、異なるキューに来る可能性もあります。そのキューをポーリングするように Adapter for JMS を構

成することができます。そのキューにきた応答は、イベント送達の一部として InterChange Server に送達することが可能です。

## JMS トランスポート用環境の設定

InterChange Server との間で文書を送受信するには、JMS キューと Adapter for JMS が必要となります。そのため、表 64に示す設定タスクと構成タスクを実行する必要があります。JMS を通じて InterChange Server と共に使用するよう WebSphere Partner Gateway を構成する方法については、123 ページの『WebSphere Partner Gateway を InterChange Server 用に構成する』を参照してください。

表 64. JMS トランスポート・プロトコルを使用する環境の設定

構成手順	詳細の参照先
1. JMS キューを構成します。	『JMS キューの構成』
2. WebSphere Business Integration Adapter for JMS を構成します。	177 ページの『Adapter for JMS の構成』

注: 文書に添付ファイルが含まれている場合、Attachment Data Handler をインストールおよび構成する必要もあります。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

### JMS キューの構成

JMS トランスポート・プロトコルを InterChange Server と共に使用するためには、WebSphere MQ が提供する JMS システムをセットアップしなければなりません。InterChange Server のサポートされるバージョンは、最新のフィックスパックが適用された WebSphere MQ バージョン 6.0 を JMS プロバイダーとして使用します。「WebSphere Partner Gateway ハブ構成ガイド」の手順を使用して、JMS トランスポート・プロトコル機構を設定できます。

**重要:** 「WebSphere Partner Gateway ハブ構成ガイド」の手順は、WebSphere Partner Gateway が存在するコンピューターで実行する必要があります。本書では、Adapter for JMS および InterChange Server が必要とする JMS トランスポート機構が、InterChange Server インストールの一部としてすでに設定されていることを前提としています。

WebSphere Partner Gateway と InterChange Server の間で使用する JMS キューを作成するときは、以下の点を考慮してください。

- InterChange Server のインストール・プロセスの一部として、WebSphere MQ キュー・マネージャーの作成が含まれます。このキュー・マネージャーは、WebSphere Partner Gateway と共に使用できます。
- JMS キューの別名を作成するときは、WebSphere Partner Gateway と InterChange Server の間のフローの方向がわかるような名前を付けることができます。例えば、177 ページの表 65 の元のキュー名の列に示すキューを作成する場合、これらのキューの名前を、177 ページの表 65 の方向を示すキュー名の列に示すように InterChange Server の方向を示す名前に変更できます。

表 65. InterChange Server の JMS キュー名の方向を示す名前への変更

元のキュー名	方向を示すキュー名
inQ	ICS2HUB
outQ	HUB2ICS

## Adapter for JMS の構成

Adapter for JMS により、WebSphere Partner Gateway は、JMS メッセージ形式で InterChange Server と文書を交換できます。以下のような InterChange Server との対話をサポートします。

- 要求処理の場合、InterChange Server から要求ビジネス・オブジェクトを受信して、それを JMS メッセージに変換し (データ・ハンドラーを使用)、その JMS メッセージを JMS キューに格納します (174 ページの図 36 参照)。WebSphere Partner Gateway は JMS キューから JMS メッセージを取り出すことができます。
- イベント通知の場合、WebSphere Partner Gateway からの JMS メッセージを取得するため JMS キューをポーリングします。JMS メッセージを検出すると、それをイベント・ビジネス・オブジェクトに変換し (データ・ハンドラーを使用)、InterChange Server に送信します。

**重要:** WebSphere Partner Gateway には WebSphere Business Integration Adapter for JMS は組み込まれていません。この製品は別途入手する必要があります。「Adapter for JMS ユーザーズ・ガイド」の説明に従ってインストールする必要があります。Adapter for JMS を正しくインストールおよび構成するには、本書で説明する手順を読んでおくことが重要です。

InterChange Server と通信するように Adapter for JMS を構成した場合、このセクションの手順に従って、JMS メッセージを WebSphere Partner Gateway から受け入れるようにこのアダプターを構成します。

- 『ペイロード・データ・ハンドラーの指定』
- 178 ページの『JMS キューの識別』

**ペイロード・データ・ハンドラーの指定:** 174 ページの図 36 に示すように、Adapter for JMS はデータ・ハンドラーを使用して、InterChange Server から受信したビジネス・オブジェクトを適切な JMS メッセージに変換します。

**注:** Adapter for JMS が呼び出すデータ・ハンドラーは、文書のペイロードを変換します。文書が XML トランスポート・エンベロープでラップされている (添付ファイルを含んでいる、またはエンベロープ・フラグが Yes に設定されている) 場合は、Attachment Data Handler をペイロード・データ・ハンドラーとして構成します。詳細については、131 ページの『添付ファイルのある文書の処理』を参照してください。

ペイロードを変換する際に使用するデータ・ハンドラーを指定するには、128 ページの『ビジネス・オブジェクトの変換』に示す手順を実行する必要があります。また、このペイロード・データ・ハンドラーを使用するように Adapter for JMS を構成する必要もあります。Connector Configurator で以下の手順を実行します。

1. ペイロード・データ・ハンドラーを指定するために、以下のコネクタ構成プロパティを設定します。

- `DataHandlerConfigMO` をトップレベル・データ・ハンドラー・メタオブジェクトの名前で設定し、`DataHandlerMimeType` プロパティをサポートされる MIME タイプで設定します。
- `DataHandlerClassName` プロパティを、インスタンスを生成するデータ・ハンドラー・クラスの名前で設定します。

**注:** `DataHandlerConfigMO` プロパティと `DataHandlerMimeType` プロパティの組み合わせ、または `DataHandlerClassName` プロパティのいずれかを設定します。

2. サポートされるビジネス・オブジェクトのリストに、トップレベル・データ・ハンドラー・メタオブジェクトを含めます。

静的メタオブジェクトまたは動的メタオブジェクトで使用するデータ・ハンドラーを指定することもできます。これらのメタオブジェクトでは、同じプロパティ (`DataHandlerMimeType`、`DataHandlerConfigMO`、および `DataHandlerClassName`) が属性として使用可能です。詳細については、「*Adapter for JMS ユーザーズ・ガイド*」を参照してください。

### JMS キューの識別:

*Adapter for JMS* が *InterChange Server* から文書を受信すると、アウトバウンド・キュー (*WebSphere Partner Gateway Receiver* がポーリングしている) にメッセージを格納します。同様に、*WebSphere Partner Gateway* が *InterChange Server* に文書を送信する場合、*Adapter for JMS* がポーリングしているアウトバウンド・キューに文書を格納します。

表 66 では、*Adapter for JMS* が文書の送受信に使用する JMS キューの構成方法をまとめています。

**注:** JMS キューの構成方法の詳細な説明については、「*Adapter for JMS ユーザーズ・ガイド*」を参照してください。

表 66. JMS キュー

JMS キュー	構成セット
入力キュー	<p><code>InputDestination</code> コネクタ構成プロパティを、着信メッセージ取得のために <i>Adapter for JMS</i> がポーリングする JMS キューの名前に設定します。</p> <p>このキューの名前は、<i>WebSphere Partner Gateway</i> が JMS アウトバウンド・キューとして使用しているキューと同じになるようにしてください。このキューが <code>InputDestination</code> で指定されていない場合、<i>Adapter for JMS</i> はキューをポーリングしません。</p> <p><b>注:</b> <code>InputDestination</code> プロパティには、コマンドで区切られた入力キューのリストが格納されています。<i>Adapter for JMS</i> が複数のキューをポーリングする場合、<i>WebSphere Partner Gateway</i> が JMS アウトバウンド・キューとして使用している JMS キューの名前が、このリストに含まれていなければなりません。</p>

表 66. JMS キュー (続き)

JMS キュー	構成セット
出力キュー	コラボレーションは実行時に、動的メタオブジェクト内の OutputQueue 属性を、Adapter for JMS が出力メッセージを送信するために使用する JMS キューの名前に動的に設定することができます。

WebSphere Partner Gateway レシーバーが listen しているキューに書き込むことができるように、静的メタオブジェクトまたは動的メタオブジェクトを構成する必要があります。

## JMS 用のビジネス・オブジェクト定義の作成

Adapter for JMS が InterChange Server との間で文書を送受信する際は、ペイロード・ビジネス・オブジェクト形式が使用されます。 WebSphere Partner Gateway の文書を送受信する際、Adapter for JMS は、ペイロード・データ・ハンドラーを呼び出して、このビジネス・オブジェクトを処理します。

- 要求処理の場合、ペイロード・データ・ハンドラーは、要求ビジネス・オブジェクトをそれに対応する JMS メッセージに変換します。
- イベント通知の場合、データ・ハンドラーは、JMS メッセージをイベント・ビジネス・オブジェクトに変換します。

したがって、表 67 に示すビジネス・オブジェクト定義を作成する際は、Adapter for JMS で前提としているペイロード・ビジネス・オブジェクト構造を表すように、これらのビジネス・オブジェクト定義を作成する必要があります。

表 67. Adapter for JMS のビジネス・オブジェクト定義

条件	ビジネス・オブジェクト定義	詳細の参照先
メッセージにパッケージ化しまたはバックエンド統合パッケージ化を使用しており、同時に 文書に添付ファイルが含まれていない 場合	ペイロード・ビジネス・オブジェクト	『JMS 用のペイロード・ビジネス・オブジェクト構造の作成』.
文書にバックエンド統合パッケージ化を使用している場合	メッセージ・ヘッダー情報を保持するビジネス・オブジェクト: <ul style="list-style-type: none"> <li>• 動的メタオブジェクト</li> <li>• JMS プロパティ・ビジネス・オブジェクト</li> </ul>	180 ページの『JMS ヘッダー情報の作成』.
文書に添付ファイルが含まれる場合	添付ファイルを表現する追加ビジネス・オブジェクトを作成する必要があります。	145 ページの『添付ファイル関連のビジネス・オブジェクト定義の作成』

## JMS 用のペイロード・ビジネス・オブジェクト構造の作成

JMS トランスポート・プロトコル用のペイロード・ビジネス・オブジェクト構造は、パッケージ化のタイプによって異なります。

- 「パッケージ化なし」を文書で使用する場合は特別の要件はなく、JMS トランスポート・プロトコルを通じて送信する文書のペイロード・ビジネス・オブジェクトを作成する必要はありません。

127 ページの『文書用ビジネス・オブジェクト』で説明したように、転送する必要のあるペイロード情報ごとに属性を作成する必要があります。

- 「バックエンド統合パッケージ化」を文書で使用する場合は、以下の手順を実行する必要があります。
  1. 動的メタオブジェクトを指定する特別な属性をペイロード・ビジネス・オブジェクト定義に追加します。この属性は、メッセージのトランスポート・レベル・ヘッダーの構成情報を提供します。
  2. ビジネス・オブジェクト・レベルのアプリケーション固有情報に、動的メタオブジェクトを含む属性を示す `cw_mo_conn` タグを追加します。

この手順について詳しくは、『JMS ヘッダー情報の作成』を参照してください。

**注:** 要求処理の場合、JMS トランスポート・プロトコルは、非同期の対話のみをサポートすることができます。要求ビジネス・オブジェクトを送信することはできませんが、応答を取得することはできません。そのため、要求ビジネス・オブジェクト定義を作成する必要がありますが、応答用のビジネス・オブジェクト定義を作成する必要はありません。

## JMS ヘッダー情報の作成

JMS トランスポート・プロトコルを通じてバックエンド統合パッケージ化を使用して文書を送信または受信する場合、要求ビジネス・オブジェクトには、カスタム・トランスポート・レベル・ヘッダー情報が含まれている必要があります。Adapter for JMS は、このカスタム・ヘッダー情報が動的メタオブジェクト内に存在することを前提としています。

181 ページの図 37 に、バックエンド統合パッケージ化を用いた WebSphere Partner Gateway 文書を表現する、要求ビジネス・オブジェクト用に Adapter for JMS が使用するビジネス・オブジェクト構造を示します。

**注:** 「Adapter for JMS ユーザーズ・ガイド」では、この必要なビジネス・オブジェクト構造に関する情報を提供しています。ビジネス・オブジェクト定義を定義するときには、このガイドを参照してください。

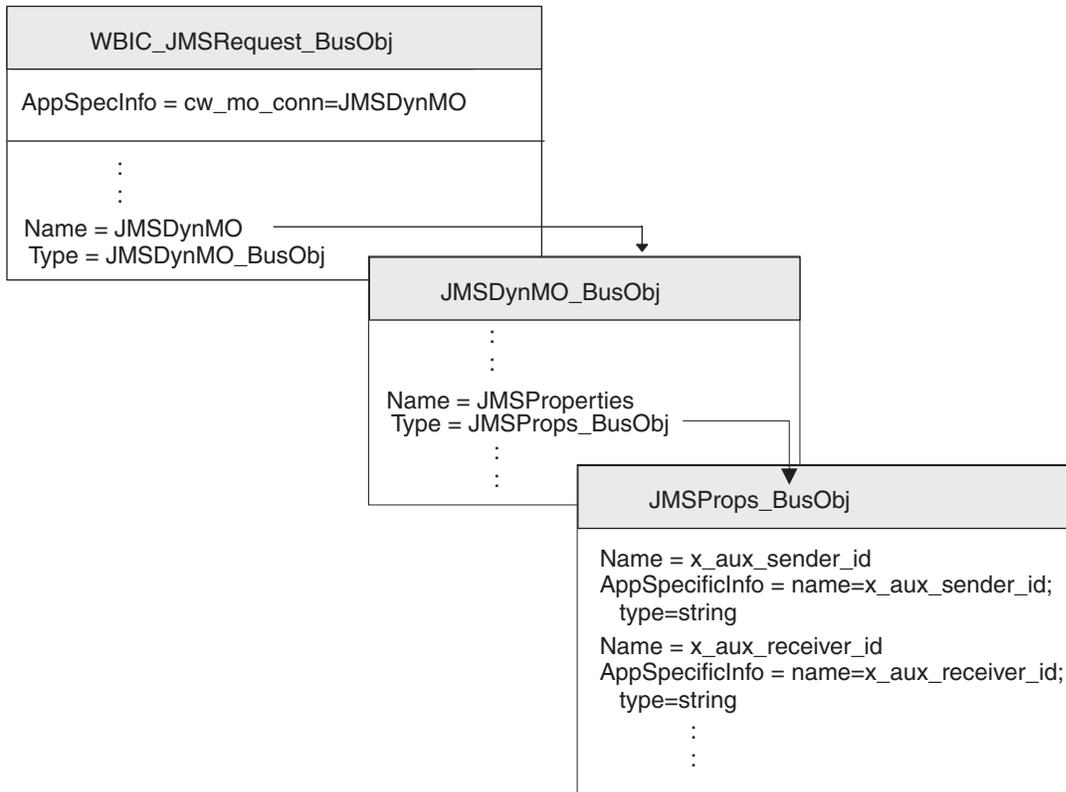


図 37. 要求ビジネス・オブジェクトと JMS 動的メタオブジェクトとの関係

以下の手順を実行して、ビジネス・オブジェクト構造に動的子メタオブジェクトが含まれるようにします。

1. バックエンド統合パッケージ化に必要な JMS プロパティを保持するビジネス・オブジェクト定義を作成します。
2. 動的メタオブジェクトのビジネス・オブジェクト定義を作成します。
3. 要求ビジネス・オブジェクトが動的メタオブジェクトの属性を組み込むようにビジネス・オブジェクト定義を変更します。

この手順の各ステップについて、以下のセクションで説明します。

**JMS プロパティ・ビジネス・オブジェクトの作成:** JMS プロパティ・ビジネス・オブジェクトには、バックエンド統合パッケージ化で必要とされるトランスポート・レベル・ヘッダーに必要な JMS プロパティが含まれています。また、コンテンツ・タイプ・ヘッダーを要求メッセージ内に設定することを指定する Content-Type 属性と、メッセージの長さをバイト単位で指定するコンテンツ長属性を含めることもできます。29 ページの表 8 では、有効なトランスポート・ヘッダー・フィールドのそれぞれについて説明しています。

JMS プロパティ・ビジネス・オブジェクト定義を作成するには、以下の手順を実行します。

1. 各トランスポート・レベル・ヘッダー・フィールドごとにビジネス・オブジェクト定義内部の属性を作成します。

すべての属性の属性タイプをストリングにする必要があります。JMS メッセージの場合、表 68 に示すように、トランスポート・ヘッダー・フィールドの名前にはハイフンの代わりに下線を使用します。

2. JMS プロパティー・ビジネス・オブジェクト定義内の各属性ごとに、関連属性の目的を示すアプリケーション固有情報を追加します。

この属性レベルのアプリケーション固有情報の形式は、次のようになっていなければなりません。

```
name=JMSproperty;type=string
```

ここで、*JMSproperty* は、表 68 の JMS プロパティー名列の値のいずれかです。

3. JMS プロパティー・ビジネス・オブジェクト定義内のどの属性でも、そのトランスポート・レベル・フィールド用の一般的な (または唯一の有効な) 値を示すデフォルト値を追加することができます。

表 68. JMS プロパティー・ビジネス・オブジェクト定義の属性

トランスポート・ヘッダー・フィールド	JMS プロパティー名
x-aux-sender-id	x_aux_sender_id
x-aux-receiver-id	x_aux_receiver_id
x-aux-protocol	x_aux_protocol
x-aux-protocol-version	x_aux_protocol_version
x-aux-process-type	x_aux_process_type
x-aux-process-version	x_aux_process_version
x-aux-create-datetime	x_aux_create_datetime
x-aux-msg-id	x_aux_msg_id
x-aux-production	x_aux_production
x-aux-system-msg-id	x_aux_system_msg_id
x-aux-payload-root-tag	x_aux_payload_root_tag
x-aux-process-instance-id	x_aux_process_instance_id
x-aux-event-status-code	x_aux_event_status_code
x-aux-third-party-bus-id	x_aux_third_party_bus_id
x-aux-transport-retry-count	x_aux_transport_retry_count
content-type	content_type
content-length	content_length

注: 表 68 では、バックエンド統合に必要なヘッダーを完全に網羅したリストを提供していません。ヘッダーの詳細なリストと説明については、29 ページの『トランスポート・レベル・ヘッダーの内容』を参照してください。トランスポート・ヘッダー・フィールド名内のハイフンは下線に置き換えてください。

181 ページの図 37 では、JMSProps\_BusObj ビジネス・オブジェクト定義にはさまざまなトランスポート・レベル・ヘッダー・フィールドの属性が含まれています。これらの属性のすべてに、関連プロトコル・ヘッダーの名前を指定する属性レベルのアプリケーション固有情報があります。例えば、x\_aux\_sender\_id 属性には次のようなアプリケーション固有情報セットがあります。

```
name=x_aux_sender_id;type=string
```

**JMS 動的メタオブジェクトの作成:** この動的メタオブジェクトには、Adapter for JMS の構成情報を所有する子ビジネス・オブジェクトが含まれます。動的メタオブジェクトのビジネス・オブジェクト定義を作成するには、以下の手順を実行します。

1. `JMSProperties` という名前の属性を作成します。その属性タイプは、JMS プロパティー・ビジネス・オブジェクトのビジネス・オブジェクト定義です (181 ページの『JMS プロパティー・ビジネス・オブジェクトの作成』参照)。
2. 必要に応じて他の構成プロパティーを追加します。動的メタオブジェクト内の有効な属性のリストについては、「*Adapter for JMS ユーザーズ・ガイド*」を参照してください。動的メタオブジェクトを構成するための属性の作成方法については、このガイドを参照してください。

Adapter for JMS を WebSphere Partner Gateway と共に動作させるためには、動的メタオブジェクトのビジネス・オブジェクト定義には、`JMSProperties` という名前の属性を組み込む必要があります。この属性の属性タイプは、JMS プロパティー・ビジネス・オブジェクトのビジネス・オブジェクト定義です (181 ページの『JMS プロパティー・ビジネス・オブジェクトの作成』参照)。例えば、181 ページの図 37 では、`JMSDynMO_BusObj` ビジネス・オブジェクト定義にはさまざまな構成プロパティー (図示せず) が含まれ、また `JMSProperties` 属性が組み込まれています。

#### 要求ビジネス・オブジェクト定義の変更: このタスクについて

ビジネス・オブジェクト構造を要求ビジネス・オブジェクトに取り込むには、要求ビジネス・オブジェクト定義に以下の変更を加える必要があります。

1. 動的子メタオブジェクトを保持する属性を要求ビジネス・オブジェクト定義に追加します。

この属性の属性タイプは、動的メタオブジェクトのビジネス・オブジェクト定義です (『JMS 動的メタオブジェクトの作成』参照)。

2. 要求ビジネス・オブジェクト定義のビジネス・オブジェクト・レベルのアプリケーション固有情報に、動的メタオブジェクトを含む属性を示す `cw_mo_conn` タグを追加します。

`cw_mo_conn` タグの形式は、次のようになっています。

```
cw_mo_conn=dynamicMetaObjAttr
```

ここで、`dynamicMetaObjAttr` は、動的メタオブジェクトを保持する要求ビジネス・オブジェクト内の属性の名前です。

例えば、181 ページの図 37 では、`JMSDynMO` という名前の属性が要求ビジネス・オブジェクト定義 `HUB_JMSRequest_BusObj` に追加されています。この属性には、タイプ `JMSDynMO_BusObj` の子ビジネス・オブジェクトである動的メタオブジェクトが含まれています。さらに、この動的メタオブジェクトを識別するため、要求ビジネス・オブジェクトのアプリケーション固有情報は、次のような `cw_mo_conn` タグを組み込むように変更されています。

```
cw_mo_conn=JMSDynMO
```

## JMS 用の ICS 成果物の作成

JMS トランスポート・プロトコルを通じて WebSphere Partner Gateway と通信するように InterChange Server を構成するには、表 69 に示す InterChange Server 成果物を作成する必要があります。

表 69. JMS トランスポート・プロトコルを通じて通信する場合の ICS 成果物

ICS 成果物	目的	詳細の参照先
ビジネス・オブジェクト定義	文書を表します。	179 ページの『JMS 用のビジネス・オブジェクト定義の作成』
コネクタ・オブジェクト	実行時の Adapter for JMS を表します。	『JMS コネクタ・オブジェクトの作成』
コラボレーション・テンプレートとコラボレーション・オブジェクト	文書処理の際に InterChange Server が使用するビジネス・プロセスを表します。	『Adapter for JMS と通信する場合のコラボレーションのバインド』

### JMS コネクタ・オブジェクトの作成 このタスクについて

Adapter for JMS のインスタンスを実行時に取得するには、System Manager で以下の手順を実行します。

1. 以下のようにコネクタ・オブジェクトを作成します。
  - Adapter for JMS のインスタンスを表現するコネクタ・オブジェクトを生成します。

**注:** Connector Configurator の「サポートされているビジネス・オブジェクト」タブで、Adapter for JMS で使用するために作成したビジネス・オブジェクト定義をすべて指定してください。これらのビジネス・オブジェクト定義の説明については、179 ページの『JMS 用のビジネス・オブジェクト定義の作成』を参照してください。

- コラボレーションが必要であれば、Port Connector 用のコネクタ・ビジネス・オブジェクトを作成します。
2. コネクタ・オブジェクトを構成します。

WebSphere Partner Gateway で使用するために Adapter for JMS を構成する方法については、177 ページの『Adapter for JMS の構成』を参照してください。

### Adapter for JMS と通信する場合のコラボレーションのバインド

130 ページの『コラボレーションの作成』で説明したように、ビジネス・オブジェクトの受信場所や送信場所を InterChange Server で認識するには、コラボレーション・オブジェクトが実行時に存在しなければなりません。WebSphere Partner Gateway との間で情報を送受信する際に Adapter for JMS を使用するコラボレーションに対してコラボレーション・オブジェクトを作成するときは、以下のようにコラボレーション・ポートをバインドします。

- 要求を処理する場合は、WebSphere Partner Gateway に要求を送信する「送信先」ポートを、Adapter for JMS 用に作成したコネクタ・オブジェクトに設定する必要があります。すなわち、Adapter for JMS は宛先 アダプターです。
- イベントの通知の場合は、WebSphere Partner Gateway からイベントを受信する「受信側」ポートを、Adapter for JMS 用に作成したコネクタ・オブジェクトに設定する必要があります。すなわち、Adapter for JMS はソース・アダプターです。



---

## 第 5 章 その他のバックエンド・システムとの統合

以下の章では、WebSphere Partner Gateway を WebSphere Message Broker や WebSphere Data Interchange などのバックエンド・システムと統合する手順について詳しく説明します。

---

### WebSphere Message Broker との統合

この章では、WebSphere Partner Gateway を WebSphere Message Broker と統合する方法を説明します。

**注記:**

1. WebSphere Partner Gateway とバックエンド・システムの統合に使用する一般的なプロセスについては、11 ページの『バックエンド統合の計画』を参照してください。
2. この章では、読者が WebSphere Message Broker と、プロジェクトやメッセージ・フローなどの関連コンポーネントを十分に理解していることが前提になっています。

WebSphere Partner Gateway をバックエンド・システムと統合する作業は、たいていの場合、異なる 2 人の担当者、つまり役割が行います。各役割は、その専門技術に応じて、特定のコンポーネントを構成します。したがって、この章では、WebSphere Message Broker との統合は WebSphere Partner Gateway の構成と Message Broker の構成に分割されています。表 70 に、これらの構成の役割と、関連する構成情報が記載されている本章内のセクションを示します。

表 70. WebSphere Message Broker 統合の役割

構成の役割	詳細の参照先
WebSphere Partner Gateway の構成	188 ページの『WebSphere Message Broker との統合の計画』
WebSphere Message Broker の構成	1. 188 ページの『WebSphere Message Broker との統合の計画』 2. 192 ページの『WebSphere Message Broker の構成』

**注:** 上記の各構成の役割は別々に実行できますが、各構成では、2 つのコンポーネント間で通信できるようにするための共通情報も必要になります。

この章で説明する内容は、次のとおりです。

- 188 ページの『WebSphere Message Broker との統合の計画』
- 188 ページの『WebSphere Message Broker との統合の計画』
- 192 ページの『WebSphere Message Broker の構成』
- 193 ページの『WebSphere Message Broker での HTTP トランスポート・プロトコルの使用』

- 197 ページの『SOAP 文書の送信』
- 198 ページの『WebSphere Message Broker での JMS トランスポート・プロトコルの使用』

## WebSphere Message Broker との統合の計画

WebSphere Message Broker との統合を計画するには、11 ページの『バックエンド統合の計画』で説明されている手順を実行してください。表 71 には、WebSphere Partner Gateway と WebSphere Message Broker とを統合するための統合手順がまとめられています。

表 71. WebSphere Message Broker との統合の計画

統合の手順	詳細の参照先
1. サポートされているバージョンの WebSphere Message Broker がインストール済みで、これを WebSphere Partner Gateway が使用可能であることを確認します。	187 ページの『WebSphere Message Broker との統合』：  『WebSphere Partner Gateway がサポートする WebSphere Message Broker のバージョン』
2. WebSphere Message Broker 文書のビジネス・プロトコルを決定します。	11 ページの『バックエンド統合の計画』： 11 ページの『ビジネス・プロトコルの選択』
3. 文書のパッケージ化タイプ（「なし」またはバックエンド統合）を決定します。	11 ページの『バックエンド統合の計画』： 28 ページの『使用するパッケージ化』
4. WebSphere Partner Gateway と WebSphere Message Broker の間で使用するメッセージ・トランスポートを決定します。	187 ページの『WebSphere Message Broker との統合』：  『WebSphere Message Broker によってサポートされるメッセージ・トランスポート』
5. WebSphere Partner Gateway を構成します。	187 ページの『WebSphere Message Broker との統合』

### WebSphere Partner Gateway がサポートする WebSphere Message Broker のバージョン

WebSphere Partner Gateway は、Message Broker の最新のフィックスパックが適用されたバージョン 6.0 との統合をサポートします。Message Broker は、Windows 2000 や UNIX ベースのプラットフォームを含む、いくつかのプラットフォームで使用可能です。詳しくは、WebSphere Message Broker 資料セット内の Message Broker のインストール・ガイドを参照してください。

### WebSphere Message Broker によってサポートされるメッセージ・トランスポート

WebSphere Message Broker では、以下の 2 つのメッセージ・トランスポート・プロトコルがサポートされています。

- HTTP トランスポート・プロトコル (Web サービスを含む)

- JMS トランスポート・プロトコル

これらのメッセージ・トランスポート・プロトコルをサポートするには、IBM WebSphere MQ をインストールして構成する必要があります。

**HTTP:** Message Broker では、Web サービスのトランザクションに HTTP トランスポート・プロトコルを使用します。HTTP プロトコルで WebSphere Partner Gateway と Message Broker の間で文書を送受信する場合、追加のソフトウェアは不要です。ただし、Message Broker から別の宛先に文書を送信する場合は、WebSphere MQ が必要です。

**注:** WebSphere Partner Gateway は、HTTP による Message Broker との対話として、非同期の対話と同期的な対話の両方をサポートします。

**JMS:** Message Broker では、ほとんどのトランザクションに JMS トランスポート・プロトコルを使用します。WebSphere Partner Gateway と Message Broker の間で JMS トランスポート・プロトコルを通じて文書を送受信するには、JMS キューを使用する必要があります。これら 2 つのコンポーネントが別々のコンピューターにある場合は、各コンピューター上に JMS キューを作成してください。基本的に、JMS をサポートするには、Message Broker 内のメッセージ・フローとその基盤である JMS キューを使用する必要があります。JMS の構成方法については詳しくは、198 ページの『WebSphere Message Broker での JMS トランスポート・プロトコルの使用』を参照してください。

**注:** WebSphere Partner Gateway は、JMS を通じた Message Broker との非同期の対話のみをサポートします。

## WebSphere Message Broker 統合のサポート

WebSphere Partner Gateway には、Message Broker との統合プロセスで役立つサンプルがあります。これらのサンプルは、WebSphere Partner Gateway 製品ディレクトリー内の以下のサブディレクトリーにあります。

Integration/WBI/WBIMB/samples

## WebSphere Partner Gateway を WebSphere Message Broker 用に構成する

バックエンド・システムと通信するように WebSphere Partner Gateway を構成する方法の一般的な概要は、51 ページの『WebSphere Partner Gateway の構成』に記載されています。この章では、Message Broker と通信するように WebSphere Partner Gateway を構成するために必要な手順をまとめています。

WebSphere Partner Gateway を構成するには、以下の手順を実行します。

- 発信文書をサポートするための構成

WebSphere Partner Gateway から Message Broker への文書の送信については、190 ページの『発信文書のサポート』を参照してください。

- 着信文書用の構成

Message Broker から WebSphere Partner Gateway への文書の送信については、191 ページの『着信文書のサポート』を参照してください。

## 発信文書のサポート

WebSphere Partner Gateway が任意のバックエンド・システムに文書を送信できるようにするには、53 ページの『パートナー文書の送信先の定義』で説明されている手順を実行する必要があります。バックエンド・システムが Message Broker である場合、トランスポート・タイプが WebSphere Partner Gateway および Message Broker 間のメッセージに使用されるトランスポート・プロトコルに一致する宛先を作成する必要があります。ハブは、文書を Message Broker に送信する場合、文書の送付先を知る必要があります。このロケーションは、使用するトランスポート・プロトコルに準拠していなければなりません。トランスポート・プロトコルは、Message Broker でサポートされているものでなければなりません (188 ページの『WebSphere Message Broker によってサポートされるメッセージ・トランスポート』参照)。

以下のセクションでは、Message Broker がサポートするトランスポート・プロトコル用の宛先を作成する方法をまとめています。

- 『HTTP トランスポート・プロトコルで文書を送信するための構成』
- 『JMS トランスポート・プロトコルでの受信文書の構成』

**HTTP トランスポート・プロトコルで文書を送信するための構成:** ハブは、HTTP プロトコルを通じて文書を Message Broker に送信する場合、定義された宛先を経由してメッセージを送付します。この宛先は、Message Broker による文書の受信が可能な URL を示します。HTTP プロトコルを使用する場合、Message Broker は、指定された URL に関連付けられたメッセージ・フローの HTTPInput ノードに文書を送付します。

ハブが HTTP トランスポート・プロトコルを通じて宛先経由で文書を送信できるようにするには、コミュニティー・コンソールの「宛先の詳細」ページから宛先を作成する必要があります。この宛先の構成では、HTTP 1.1 トランスポート・プロトコルを使用し、適切な HTTPInput ノードが listen する URL に書き込むように構成する必要があります。表 72 に示すように、この URL は、宛先定義の「受信側 URI」フィールドに指定します。

注: 宛先の作成方法の概要は、53 ページの『パートナー文書の送信先の定義』で説明されています。

表 72. WebSphere Message Broker と通信するための「宛先の詳細」ページの HTTP 値

「レシーバーの詳細」フィールド	値	注記および制約事項
レシーバー URI (Receiver URI)	URL は、Message Broker メッセージ・フローの HTTPInput ノード用に構成されたものと同じでなければなりません。	WebSphere Message Broker 統合内のメッセージ・フローの構成からこの URL を取得します。

**JMS トランスポート・プロトコルでの受信文書の構成:** ハブが JMS プロトコルを通じて文書を Message Broker に送信する場合、ハブは適切な JMS キューへ文書を送付します。この JMS キューから、Message Broker が取り出すことのできる JMS キューに文書を転送できます。ハブがこの JMS ロケーションを取得できるようにするには、WebSphere Partner Gateway 内に JMS トランスポート・プロトコルを使用する宛先を作成する必要があります。この宛先の構成では、書き込み先のキュー

として、Message Broker がメッセージを受信するキュー宛にキュー内容を転送するようなキューを指定する必要があります。

注：宛先の作成方法の概要については、53 ページの『パートナー文書の送信先の定義』を参照してください。

ハブが JMS トランスポート・プロトコルを通じて宛先経由で文書を送信できるようにするには、コミュニティー・コンソールの「宛先の詳細」ページから宛先を作成する必要があります。最新のフィックスバックが適用された WebSphere MQ バージョン 6.0 を JMS プロバイダーとして使用する場合の詳細な手順については、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。さらに、「宛先の詳細」ページで JMS プロトコルに表 73 で指定された情報を使用します。

表 73. WebSphere Message Broker と通信するための「宛先の詳細」ページの JMS 値

「宛先の詳細」フィールド	値	注記および制約事項
JMS キュー名	WebSphere Partner Gateway があるコンピューター上の JMS キューの名前。	このキューで受信される文書は、Message Broker があるコンピューター上の JMS キューに転送されます。

## 着信文書のサポート

WebSphere Partner Gateway が任意のバックエンド・システムから文書を受信できるようにするには、57 ページの『バックエンド文書を取り出す場所の定義』で説明されている手順を実行する必要があります。バックエンド・システムが Message Broker である場合、以下の手順を実行する必要があります。

1. 内部パートナーのパートナー・プロファイルの一部として、宛先タイプを定義し、レシーバーが listen する関連 IP アドレスを指定します。
2. トランスポート・タイプが WebSphere Partner Gateway および Message Broker 間の文書に使用されるトランスポート・プロトコルに一致する受信側を作成します。

ハブは、Message Broker から文書を受信するには、メッセージを取り出すロケーションを知る必要があります。このロケーションは、使用するトランスポート・プロトコルに準拠していなければなりません。

以下のセクションでは、Message Broker がサポートするトランスポート・プロトコル用の受信側を作成する方法をまとめています。

**HTTP トランスポート・プロトコルでの着信文書の構成：** ハブが HTTP トランスポート・プロトコルを通じて文書を受信すると、受信側は定義された受信側から文書を取り出します。この受信側は、受信側が Message Broker からの文書を取得するために listen する URL を示します。Message Broker が HTTP トランスポート・プロトコルを使用する場合は、HTTPRequest ノードが適切な URL に文書を送信し、ハブはその URL で文書を受信できます。

ハブが HTTP トランスポート・プロトコルを通じて受信側経由で文書を受信できるようにするには、コミュニティー・コンソールの「受信側リスト」ページから受信

側を作成する必要があります。この受信側は HTTP 1.1 トランスポート・プロトコルを使用する必要があります。ハブは、この URL を以下の情報の組み合わせと判断します。

- ホスト・コンピューターの IP アドレス。内部パートナーのプロファイル内部から取得します。
- 受信側 URL。受信側定義の URL フィールドから取得します。

注: 受信側の作成方法の概要は、57 ページの『バックエンド文書を取り出す場所の定義』で説明されています。

Message Broker がこの受信側に文書を送信できるようにするには、メッセージ・フローの HTTPRequest ノードが、この URL に文章を送信するように構成されていなければなりません。したがって、この受信側 URL は Message Broker 構成で確実に使用できるようにしてください。

**JMS トランスポート・プロトコルでの着信文書の構成:** ハブが JMS プロトコルを通じて文書を Message Broker から受信する場合、ハブは適切な JMS 入力キューから文書を取得します。文書は、Message Broker によって JMS 出力キューに送信され、そこから JMS 入力キューに転送されています。ハブがこの JMS ロケーションを取得できるようにするには、WebSphere Partner Gateway 内に JMS トランスポート・プロトコルを使用する受信側を作成する必要があります。ハブは、その入力キュー上にある文書を、受信側を通じて listen し、取得します。

注: 受信側の作成方法の概要については、57 ページの『バックエンド文書を取り出す場所の定義』を参照してください。

ハブが JMS トランスポートを通じて受信側経由で文書を受信できるようにするには、コミュニティー・コンソールの「受信側リスト」ページから受信側を作成する必要があります。最新のフィックスパックが適用された WebSphere MQ バージョン 6.0 を JMS プロバイダーとして使用する場合の詳細な手順については、

「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。さらに、「受信側の詳細」ページで JMS プロトコルに表 74 で使用された情報を指定します。

表 74. *WebSphere Message Broker* と通信するための「受信側の詳細」ページの JMS 値

「レシーバーの詳細」フィールド	値	注記および制約事項
JMS キュー名	Message Broker の出力キューから文書を受信する JMS 入力キューの名前。	この入力キュー内の文書は、Message Broker があるコンピューター上の JMS 出力キューから転送されます。

## WebSphere Message Broker の構成

WebSphere Partner Gateway と Message Broker の間の対話用に、Message Broker のツールキットの Broker Application Development Perspective 内にメッセージ・フロー・プロジェクトを作成する必要があります。このプロジェクトには、以下の成果物が組み込まれます。

- メッセージ・フロー
- PIP ファイル (RosettaNet のみ) またはメッセージ定義ファイル

注: メッセージ・フロー・プロジェクトの作成方法について詳しくは、WebSphere Message Broker の資料セットを参照してください。

## メッセージ・フローの作成

情報を処理するために必要な実際のビジネス・ロジックを実行するのは、Message Broker 内部のメッセージ・フローです。そのため、WebSphere Partner Gateway 文書を正しく処理するには、Message Broker 用の適切なメッセージ・フローが存在しなければなりません。必要なビジネス・ロジックを提供するメッセージ・フローが存在することを確認してください。

- 現在そのようなメッセージ・フローが存在しない場合は、メッセージ・フローを作成するか、インポートする必要があります。
- メッセージ・フローが存在する場合、その使用方法を理解しておく必要があります。

Message Broker が着信文書および発信文書を処理できるようにするには、そのメッセージ・フローで特殊なトランスポート・ノードを使用します。使用するトランスポート・ノードのタイプは、個々のトランスポート・プロトコルによって以下のように異なります。

表 75. トランスポート・プロトコル別のメッセージ・フローの作成

トランスポート・プロトコル	詳細の参照先
HTTP HTTP (SOAP 文書)	195 ページの『HTTP トランスポート用のメッセージ・フローの作成』
JMS	204 ページの『JMS トランスポート用のメッセージ・フローの作成』

## プロジェクトの配置

メッセージ・フロー・プロジェクトに適切な成果物が格納されたら、そのプロジェクトを Message Broker に配置する必要があります。メッセージ・フロー・プロジェクトを配置するには、Message Broker ツールキットの Broker Administrator Perspective を使用します。

## WebSphere Message Broker での HTTP トランスポート・プロトコルの使用

このセクションでは、HTTP トランスポート・プロトコルを使用して WebSphere Partner Gateway と WebSphere Message Broker の間で文書を送受信する方法について説明します。

注: HTTP トランスポート・プロトコルに言及する部分はすべて、HTTPS にも同様に適用されます。

### HTTP トランスポートを通じた文書に必要なコンポーネント

HTTP トランスポート・プロトコルを使用して WebSphere Partner Gateway と Message Broker の間で文書を送受信する場合、追加のソフトウェアは不要です。必要なのは、WebSphere Partner Gateway と Message Broker のみです。WebSphere

Partner Gateway が HTTP トラnsポート・プロトコルを使用して Message Broker バージョン 5.0 と通信するには、これら 2 つのコンポーネントを構成しておく必要があります。表 76 に、これらの構成ステップの要約を示します。

表 76. WebSphere Partner Gateway および WebSphere Message Broker の構成

コンポーネント	バージョン	詳細の参照先
WebSphere Partner Gateway	6.1	190 ページの『HTTP トラnsポート・プロトコルで文書を送信するための構成』  191 ページの『HTTP トラnsポート・プロトコルでの着信文書の構成』
WebSphere Message Broker	最新のフィックスパックが適用された 6.0	192 ページの『WebSphere Message Broker の構成』

さらに、HTTP トラnsポート・プロトコルを使用して WebSphere Partner Gateway と Message Broker の間で文書を送受信するには、JMS プロバイダーとして最新のフィックスパックが適用された IBM WebSphere MQ バージョン 6.0 を使用する必要があります。

**HTTP トラnsポートを通じた文書の送信:** WebSphere Partner Gateway が HTTP トラnsポート・プロトコルを通じて Message Broker に文書を送信できるようにするには、Message Broker メッセージ・フロー内の特殊な HTTP トラnsポート・ノードを使用して、WebSphere Partner Gateway によって HTTP ストリームとして送信された文書を取り出します。メッセージ・フローのノードは、必要な計算を実行して、文書を宛先 (JMS 出力キュー) に送付します。

WebSphere Partner Gateway が HTTP トラnsポート・プロトコルを通じて Message Broker 内のメッセージ・フローに文書を送信する手順を以下に示します。

1. WebSphere Partner Gateway は Message Broker に HTTP メッセージを送信します。

文書のパッケージ化タイプがバックエンド統合であれば、WebSphere Partner Gateway は、このメッセージ内にカスタム・プロパティを用意しています。

**注:** WebSphere Partner Gateway 内部で、WebSphere Partner Gateway からのメッセージの送信先であり、Message Broker のポーリング対象である URL を示す宛先を構成する必要があります。詳細については、190 ページの『HTTP トラnsポート・プロトコルで文書を送信するための構成』を参照してください。

2. メッセージ・フローの HTTPInput ノードは、文書を取り出して、メッセージ・フローの次のノードに送信します。このノードは、通常は Compute ノードです。
3. メッセージ・フローのノードが、ビジネス・ロジックを実行します。

ビジネス・ロジックが完了すると、メッセージ・フローは結果の文書を HTTPReply ノードに送信します。

4. HTTPReply ノードは、デフォルトで出力メッセージをクライアント (WebSphere Partner Gateway) に送信します。

あるいは、メッセージ・フローはメッセージを MQOutput ノードに置くこともできます。MQOutput ノードは、文書を受信すると、その文書を適切な JMS キューまたは他のアプリケーションに送信します。

**HTTP トラnsポートを通じた文書の受信:** WebSphere Partner Gateway が HTTP トラnsポート・プロトコルを使用して Message Broker から文書を受信できるようにするには、Message Broker メッセージ・フロー内の特殊な HTTP トラnsポート・ノードを使用して、WebSphere Partner Gateway が HTTP ストリームとして受信する予定の文書を送信します。メッセージ・フローのノードは、必要な計算を実行し、WebSphere Partner Gateway に関する要求と応答を処理します (同期対話の場合)。

WebSphere Partner Gateway が HTTP トラnsポート・プロトコルを通じて Message Broker 内のメッセージ・フローから文書を受信する手順を以下に示します。

1. Message Broker 内のメッセージ・フローは、MQInput ノード (JMS 入力キュー) で文書を受信します。
2. メッセージ・フローの MQInput ノードは、文書を受信し、それを HTTPRequest ノードに送信します。
3. HTTPRequest ノードは、指定された URL を使用して、クライアント (WebSphere Partner Gateway) との要求および応答の対話を処理します。
4. WebSphere Partner Gateway は、受信側で構成されているように、URL からメッセージを受信します。

受信側について詳しくは、191 ページの『HTTP トラnsポート・プロトコルでの着信文書の構成』を参照してください。

## HTTP トラnsポート用のメッセージ・フローの作成

Message Broker メッセージ・フローが HTTP トラnsポート・プロトコルを通じて文書処理するには、次のトラnsポート・ノードを使用します。

- HTTPInput
- HTTPReply
- HTTPRequest

これらのトラnsポート・ノードの使用順序は、通信の方向によって以下のように異なります。

- WebSphere Partner Gateway が Message Broker に文書を送信する場合、メッセージ・フローには 196 ページの表 77 のタイプのノードが (表中の順序で) 組み込まれて、ビジネス・ロジックが記述されます。
- WebSphere Partner Gateway が Message Broker から文書を受信する場合、メッセージ・フローには 196 ページの表 78 のタイプのノードが (表中の順序で) 組み込まれて、ビジネス・ロジックが記述されます。

表 77. HTTP を通じて WebSphere Message Broker に文書を送信するためのノード

ノード・タイプ	目的	注記
HTTPInput	メッセージ・フローに入る WebSphere Partner Gateway 要求文書を受信します。	このトランスポート・ノードの「URL セレクター (URL Selector)」フィールド (「基本 (Basic)」プロパティ内) を、WebSphere Partner Gateway が文書を送信する URL (WebSphere Partner Gateway レシーバーで構成される URL) に設定します。URL は、次のような形式になります。  <code>http://hostName:port/path</code>  ここで、 <i>hostName</i> は Message Broker があるコンピューターの名前で、 <i>port</i> は Message Broker が listen する HTTP ポート番号です。 <i>path</i> は、このコンピューターのロケーションを示します。  詳細については、190 ページの『HTTP トランスポート・プロトコルで文書を送信するための構成』を参照してください。
Compute	ヘッダー情報の更新など、ビジネス・ロジックのタスクを実行します。	ESQL を使用して、ビジネス・ロジックを実行します。Compute ノードは結果のメッセージを HTTPReply ノードに送信します。
HTTPReply	応答を WebSphere Partner Gateway に戻します。	デフォルトで、このノードは出力メッセージをクライアントに送信します。ただし、出力メッセージを MQOutput ノードに送信するよう構成することもできます。
MQOutput	HTTPReply ノードから文書を受信し、それを WebSphere Partner Gateway に送信します。	このトランスポート・ノードは、結果の文書を JMS 出力キューに送信し、JMS 出力キューは文書を次の宛先に送付します。

表 78. HTTP を通じて WebSphere Message Broker から文書を受信するためのノード

ノード・タイプ	目的	注記
MQInput	WebSphere Partner Gateway から文書を受信します。	このトランスポート・ノードは、JMS 入力キューから着信文書を受信します。
HTTPRequest	WebSphere Partner Gateway との要求/応答の対話を処理します。	このトランスポート・ノードでは、「Web サービス URL (Web Services URL)」フィールド (「基本 (Basic)」プロパティ内) を、WebSphere Partner Gateway が文書を listen する URL (WebSphere Partner Gateway レシーバーで構成される URL) に設定します。URL は、次のような形式になります。  <code>http://hostName:port/bcgreceiver/path</code>  ここで、 <i>hostName</i> は WebSphere Partner Gateway があるコンピューターの名前で、 <i>port</i> は WebSphere Partner Gateway 受信側が listen する HTTP ポート番号です。 <i>path</i> は、このコンピューターのロケーションを示します。  詳細については、191 ページの『HTTP トランスポート・プロトコルでの着信文書の構成』を参照してください。

メッセージ・フロー・ノードの作成および構成方法についての詳細は、WebSphere Message Broker の資料を参照してください。

## SOAP 文書の送信

SOAP 文書は、HTTP/S で交換される他のタイプの文書とは異なります。このセクションでは、HTTP トランスポート・プロトコルを通じて WebSphere Partner Gateway と WebSphere Message Broker の間で SOAP 文書を送受信する方法について説明します。

SOAP 文書を転送できるように WebSphere Partner Gateway と Message Broker を構成する方法は、HTTP プロトコルを通じて非 SOAP 文書を転送する場合の構成と非常によく似ています。表 79 に、これら 2 つの統合コンポーネントの構成方法についての詳細の参照先をまとめます。

表 79. SOAP 文書を転送するための WebSphere Partner Gateway と WebSphere Message Broker の構成

統合コンポーネント	構成手順	詳細の参照先
WebSphere Partner Gateway	HTTP を通じて非 SOAP 文書を転送する場合と同じ方法で、SOAP 用に受信側と宛先を構成します。	190 ページの『HTTP トランスポート・プロトコルで文書を送信するための構成』  191 ページの『HTTP トランスポート・プロトコルでの着信文書の構成』
WebSphere Message Broker	SOAP 文書を処理するメッセージ・フローは、HTTP を通じて非 SOAP 文書を転送する場合と非常によく似ています。SOAP 文書の処理では、トランスポート・ノードを 1 つだけ追加する必要があります。	Message Broker への SOAP 文書の送信については、198 ページの表 80 を参照してください。  Message Broker からの SOAP 文書の受信については、195 ページの『HTTP トランスポート用のメッセージ・フローの作成』を参照してください。

WebSphere Partner Gateway から送信された SOAP 文書を Message Broker が正しく処理できるようにするには、Web サービス・クライアントとの通信を処理する HTTPRequest ノードをメッセージ・フロー内に入れる必要があります。198 ページの表 80 に、WebSphere Partner Gateway によって送信される SOAP 文書を処理するために必要な Message Broker メッセージ・フロー内のノードを示します。

表 80. WebSphere Message Broker に SOAP 文書を送信するためのノード

ノード・タイプ	目的	注記
HTTPInput	メッセージ・フローに入る WebSphere Partner Gateway 要求文書を受信します。	このトランスポート・ノードの「URL セレクター (URL Selector)」フィールド (「基本 (Basic)」プロパティ内) を、WebSphere Partner Gateway が文書を送信する URL (WebSphere Partner Gateway 宛先内で構成される URL) に設定します。URL は、次のような形式になります。 <code>http://hostName:port/path</code>  ここで、 <i>hostName</i> は WebSphere Partner Gateway があるコンピューターの名前で、 <i>port</i> は WebSphere Partner Gateway 受信側が listen する HTTP ポート番号です。 <i>path</i> は、このコンピューターのロケーションを示します。  詳細については、190 ページの『HTTP トランスポート・プロトコルで文書を送信するための構成』を参照してください。
Compute	ヘッダー情報の更新など、ビジネス・ロジックのタスクを実行します。	ESQL を使用して、ビジネス・ロジックを実行します。Compute ノードは結果のメッセージを HTTPReply ノードに送信します。
HTTPRequest	SOAP 要求を外部 Web サービス・プロバイダー (WebServices) に送信して、その Web サービスから応答を受信します。	なし
HTTPReply	応答を WebSphere Partner Gateway に戻します。	デフォルトで、このノードは出力メッセージをクライアントに送信します。

## WebSphere Message Broker での JMS トランスポート・プロトコルの使用

このセクションでは、JMS トランスポート・プロトコルを使用して WebSphere Partner Gateway と WebSphere Message Broker の間で文書を送受信するようにコンポーネントを構成する方法について説明します。文書を送受信する方法について、以下の情報を提供します。

- 『JMS トランスポートを通じた文書に必要なコンポーネント』
- 171 ページの『JMS トランスポートを使用した文書の送信方法』
- 173 ページの『JMS トランスポートを使用した文書の受信方法』

### JMS トランスポートを通じた文書に必要なコンポーネント

JMS トランスポート・プロトコルを使用して WebSphere Partner Gateway と Message Broker バージョン 6.0 の間で文書を送受信する場合は、WebSphere MQ が JMS プロバイダーでなければなりません。以降のセクションでは、HTTP トランスポート・プロトコルを通じて文書を交換するために、WebSphere Partner Gateway、Message Broker、および WebSphere MQ がどのように連動するかについて説明します。

**JMS トランスポートを使用した文書の送信方法:** WebSphere Partner Gateway が JMS トランスポート・プロトコルを使用して文書を Message Broker に送信できるようにするには、JMS キューを使用します。WebSphere Partner Gateway は文書を JMS 出力キューに送信し、ここで文書は、Message Broker が listen する JMS 入力キューに転送されます。Message Broker は文書を受信すると、入力キューからその文書を取り出します。Message Broker のメッセージ・フローには、JMS キューへのアクセスを処理する、特殊な WebSphere MQ (JMS) トランスポート・ノードが含まれます。200 ページの図 38 では、WebSphere Partner Gateway が JMS トランスポート・プロトコルを通じて Message Broker へ文書を送信する方法の概要を示します。

## WebSphere Business Integration Message Broker

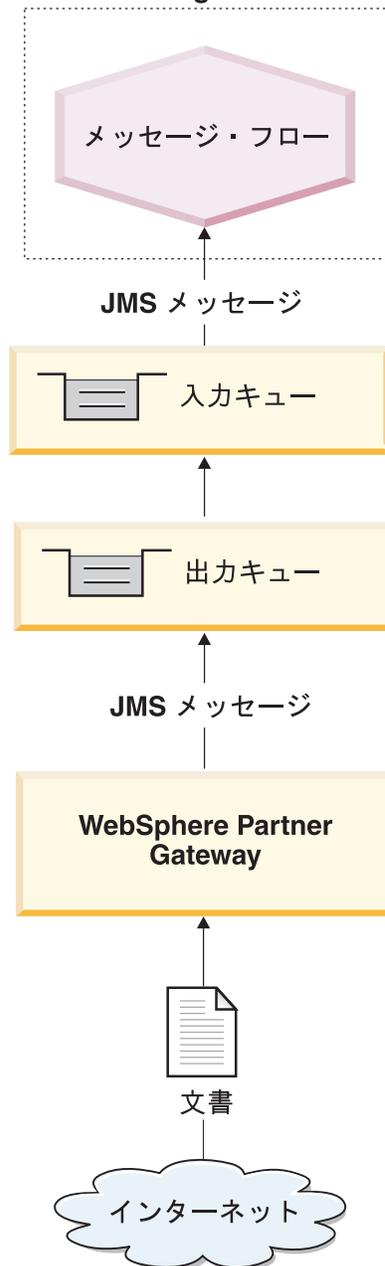


図 38. JMS トラnsポート・プロトコルによる WebSphere Partner Gateway からメッセージ・フローへのメッセージ・フロー

WebSphere Partner Gateway が JMS トラnsポート・プロトコルを通じて Message Broker 内のメッセージ・フローに文書を送信する手順を以下に示します。

1. WebSphere Partner Gateway は、メッセージを JMS 出力キューにポストします。

文書のパッケージ化タイプがバックエンド統合であれば、WebSphere Partner Gateway は、このメッセージ内にカスタム・プロパティを用意しています。JMS メッセージ・ヘッダーである JMSType が、ペイロードのコンテンツ・タイプとともに設定されます。

注: WebSphere Partner Gateway 内部で、WebSphere Partner Gateway からのメッセージの送信先であり、Message Broker のポーリング対象である JMS 出力キューを示す宛先を構成する必要があります。詳細については、190 ページの『JMS トランスポート・プロトコルでの受信文書の構成』を参照してください。

2. WebSphere MQ は、WebSphere Partner Gateway があるコンピューター上の出力キューから、Message Broker がポーリングする入力キューに文書を転送します。
3. Message Broker は、入力キューにあるメッセージを認識すると、そのメッセージを取り出して、適切なメッセージ・フローに送信します。

このキューの設定方法については、203 ページの『JMS トランスポート用環境の設定』を参照してください。

4. MQInput ノードは、メッセージ・フローの次のノードに文書を送信します。このノードは、通常は Compute ノードです。
5. メッセージ・フローのノードが、ビジネス・ロジックを実行します。

ビジネス・ロジックが完了すると、メッセージ・フローは結果の文書を MQOutput ノードに送信します。

6. MQOutput ノードは、文書を適切なキューに送信します。

**JMS トランスポートを使用した文書の受信方法:** WebSphere Partner Gateway が JMS トランスポート・プロトコルを通じて文書を Message Broker から受信できるようにするには、JMS キューを使用します。Message Broker は文書を JMS 出力キューに送信し、ここで文書は、WebSphere Partner Gateway が listen する JMS 入力キューに転送されます。WebSphere Partner Gateway は文書を受信すると、入力キューからその文書を取り出します。Message Broker のメッセージ・フローには、JMS キューへのアクセスを処理する、特殊な WebSphere MQ (JMS) トランスポート・ノードが含まれます。202 ページの図 39 に、文書が Message Broker から WebSphere Partner Gateway に送信される仕組みの概要を示します。

## WebSphere Business Integration Message Broker

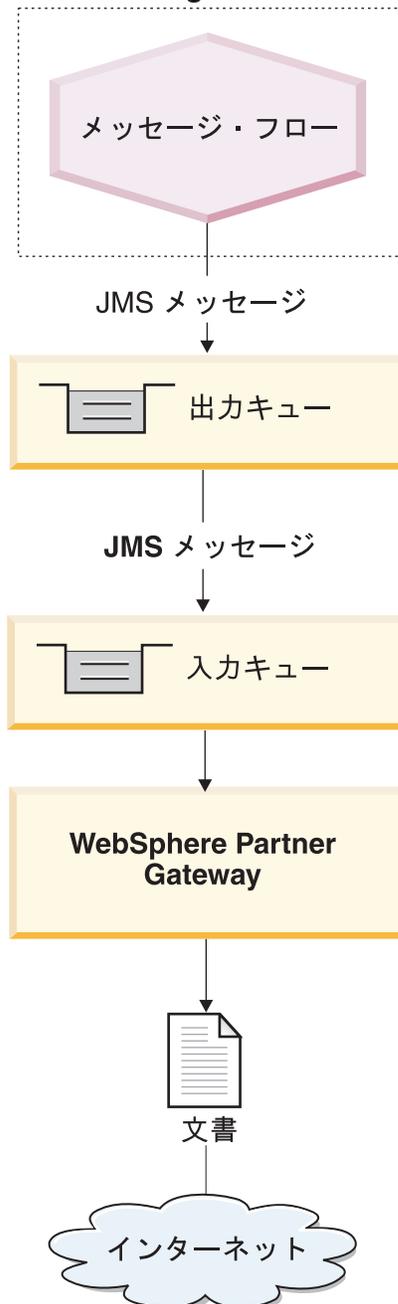


図 39. JMS トラnsポート・プロトコルによるメッセージ・フローから WebSphere Partner Gateway へのメッセージ・フロー

WebSphere Partner Gateway が JMS トラnsポート・プロトコルを通じて Message Broker 内のメッセージ・フローから文書を受信する手順を以下に示します。

1. Message Broker 内のメッセージ・フローは、MQInput ノードで文書を受信します。

メッセージ・フローは、JMS 入力キューから着信メッセージを受信します。

2. メッセージ・フローの MQInput キューは、文書を受信し、それをメッセージ・フローの次のノードに送信します。このノードは、通常は Compute ノードです。
3. メッセージ・フローのノードが、ビジネス・ロジックを実行します。

ビジネス・ロジックが完了すると、メッセージ・フローは結果の文書を MQOutput ノードに送信します。

4. MQOutput ノードは、文書を適切な JMS 出力キューに送信します。
5. WebSphere MQ は、Message Broker があるコンピューター上の出力キューから、WebSphere Partner Gateway がポーリングするキューに文書を転送します。
6. WebSphere Partner Gateway は、受信側で構成されているように、JMS からメッセージを受信します。

受信側について詳しくは、192 ページの『JMS トラnsポート・プロトコルでの着信文書の構成』を参照してください。このキューの設定方法については、『JMS トラnsポート用環境の設定』を参照してください。

## JMS トラnsポート用環境の設定

Message Broker との文書の送受信には、JMS キュー (リモートおよびローカル) が必要です。JMS を通じて Message Broker とともに使用するよう WebSphere Partner Gateway を構成する方法については、188 ページの『WebSphere Message Broker との統合の計画』を参照してください。JMS トラnsポート・プロトコルを Message Broker で使用するには、WebSphere MQ が提供する JMS システムをセットアップします。Message Broker バージョン 6.0 では、JMS プロバイダーとしてバージョン 6.0 の WebSphere MQ を使用します。「*WebSphere Partner Gateway* ハブ構成ガイド」の手順を使用して、JMS トラnsポート・プロトコル機構を設定できます。

**重要:** 「*WebSphere Partner Gateway* ハブ構成ガイド」の手順は、WebSphere Partner Gateway が存在するコンピューターで実行する必要があります。本書では、Message Broker が必要とする JMS トラnsポート機構が、Message Broker インストールの一部としてすでにセットアップされていることを前提としています。

WebSphere Partner Gateway と Message Broker の間で使用する JMS キューを作成するときは、以下の点を考慮してください。

- Message Broker インストール・プロセスの一部として、以下のキュー・マネージャーを作成する必要があります。
  - ブローカー・ドメインに関連付けられた WebSphere MQ キュー・マネージャー

次のコマンドを使用すれば、このキュー・マネージャーと、名前付きキューのセットを作成できます。

```
mqsicreatebroker
```

- Message Broker 用の WebSphere MQ キュー・マネージャー

Message Broker はあらかじめ決められたキュー名のセットを使用するので、ブローカーごとに異なる WebSphere MQ キュー・マネージャーが必要です。

Message Broker は、このキュー・マネージャーのホスティングを、構成マネージャーかオプションの User Name Server、あるいはその両方と共有できます。

詳しくは、「*WebSphere Message Broker Installation and Configuration Guide*」を参照してください。

- JMS キューの別名を作成するときは、WebSphere Partner Gateway と Message Broker の間のフローの方向がわかるような名前を付けることができます。

例えば、177 ページの表 65 の 元のキュー名の列に示すキューを作成する場合、これらのキューの名前を、表 81 の 方向を示すキュー名の列に示すようにフローの方向を示す名前に変更できます。

表 81. WebSphere Message Broker の JMS キュー名の方向を示す名前への変更

元のキュー名	方向を示すキュー名
inQ	MB2HUB
outQ	HUB2MB

## JMS トランスポート用のメッセージ・フローの作成

Message Broker メッセージ・フローが JMS トランスポート・プロトコルを通じて文書を処理するには、次のトランスポート・ノードを使用します。

- MQInput
- MQOutput

これらのトランスポート・ノードの使用順序は、通信の方向によって以下のように異なります。

- WebSphere Partner Gateway が Message Broker に文書を送信 する場合、メッセージ・フローには表 82 のタイプのノードが (表中の順序で) 組み込まれて、ビジネス・ロジックが記述されます。
- WebSphere Partner Gateway が Message Broker から文書を受信 する場合、メッセージ・フローには 205 ページの表 83 のタイプのノードが (表中の順序で) 組み込まれて、ビジネス・ロジックが記述されます。

表 82. JMS を通じて WebSphere Message Broker に文書を送信するためのノード

ノード・タイプ	目的	注記および制約事項
MQInput	WebSphere Partner Gateway から文書を受信します。	このトランスポート・ノードの「キュー名」フィールド (Basic プロパティ内) の値は、メッセージ・フローの入力キューです。この JMS キューが WebSphere Partner Gateway の出力キューから文書を受信するように WebSphere MQ をセットアップする必要があります。詳細については、190 ページの『JMS トランスポート・プロトコルでの受信文書の構成』を参照してください。
Compute	ヘッダー情報の除去など、ビジネス・ロジックのタスクを実行します。	なし
MQOutput	Compute ノードから文書を受信し、それをメッセージ・フロー出力として送信します。	このトランスポート・ノードは、結果の文書を JMS 出力キューに送信し、JMS 出力キューは文書を次の宛先に送付します。

表 83. *WebSphere Message Broker* から文書を受信するためのノード

ノード・タイプ	目的	注記および制約事項
MQInput	メッセージ・フローに入る文書を受信します。	このトランスポート・ノードは、JMS 入力キューから着信文書を受信します。
Compute	ヘッダー情報の更新など、ビジネス・ロジックのタスクを実行します。	なし
MQOutput	Compute ノードから文書を受信し、それを WebSphere Partner Gateway に送信します。	このトランスポート・ノードの「キュー名」フィールド (Basic プロパティ内) の値は、メッセージ・フローの出力キューです。この JMS キューが WebSphere Partner Gateway の入力キューに文書を送信するように WebSphere MQ をセットアップする必要があります。詳細については、192 ページの『JMS トランスポート・プロトコルでの着信文書の構成』を参照してください。

メッセージ・フロー・ノードの作成および構成方法についての詳細は、WebSphere Message Broker の資料を参照してください。

## WebSphere Data Interchange との統合

この章では、WebSphere Partner Gateway を WebSphere Data Interchange と統合する方法を説明します。

注: WebSphere Partner Gateway とバックエンド・システムの統合に使用する一般的なプロセスについては、11 ページの『バックエンド統合の計画』を参照してください。

### 本書の対象読者

WebSphere Partner Gateway には、WebSphere Data Interchange にあるものと類似した EDI 処理機能 (エンベロープ解除および変換など) が組み込まれました。WebSphere Data Interchange の代わりにこれらの EDI 機能を使用する場合、この章の情報は不要です。EDI 文書処理するようハブを設定する方法については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

この章は、以下の読者を対象としています。

- WebSphere Partner Gateway (以前の名称は WebSphere Business Integration Connect) の旧バージョンを使用していて、引き続き WebSphere Data Interchange との統合を考えている方
- WebSphere Partner Gateway EDI サポートに組み込まれていない WebSphere Data Interchange の拡張機能を必要とする方

WDI を使用する場合は、この章のステップを実行してください。WebSphere Partner Gateway バージョン 6.0 では、EDI を構成するためのステップが変更され

ていることに注意してください。使用する EDI のバージョンに固有の文書フローを選択する必要があります。例えば、EDI-X12 文書を交換する場合は、文書フローに ISA(「すべて」ではない) を選択します。

## 本章とともに使用できるリソース

本章では、文書を交換するプロセスを説明し、このような交換のためのサンプル環境をセットアップする手順をリストします。本章全体で使用するシナリオは、

「*Integrating WebSphere Data Interchange V3.2 with WebSphere Business Integration Connect V4.2*」チュートリアルで使用しているシナリオに類似しています。このチュートリアルは次の Web サイトから入手できます。

[www.ibm.com/developerworks/websphere/](http://www.ibm.com/developerworks/websphere/)

チュートリアルには、追加スクリプト (WebSphere MQ の構成に関するセクション) とサンプルの変換マップが記載されています。チュートリアルに従うと、本章で説明する環境をセットアップできます。

**注:** このチュートリアルでは、WebSphere Business Integration Connect バージョン 4.2 との統合について説明しますが、WebSphere Partner Gateway の場合も、わずかな違いはあるもののほとんど同じ手順です。

ここでは、WebSphere Data Interchange の使用法に習熟していることを前提とします。本章を読む際の追加情報については、WebSphere Data Interchange 資料を参照してください。

## 概要

WebSphere Data Interchange は、電子データ交換 (EDI) を WebSphere ビジネス・プロセス、メッセージング、およびインターネット・ベース B2B 機能に統合します。文書およびメッセージを、JMS トランスポート・プロトコルを使用して WebSphere Partner Gateway と WebSphere Data Interchange の間で交換します。文書を WebSphere Data Interchange に送信する場合には、パッケージ化に「なし」を指定する必要があります。

**注:** WebSphere Data Interchange には、ファイル・ベース統合などの、その他の統合オプションが用意されています。ファイル・ベース統合を使用して文書の交換を使用可能にするための詳細については、WebSphere Data Interchange 資料を参照してください。

### WebSphere Data Interchange への文書の送信方法

WebSphere Partner Gateway から WebSphere Data Interchange に EDI 文書を送信する際は、以下の手順が実行されます。

1. 外部パートナーが、EDI 文書を WebSphere Partner Gateway に送信します。文書は、トランスポート・プロトコルを使用した特定のパッケージ (この例では、HTTP による AS2 パッケージ) で送信されます。WebSphere Partner Gateway は、EDI 文書から AS2 パッケージを除去します。
2. WebSphere Partner Gateway は、EDI 文書をキューに入れます。

注: WebSphere Partner Gateway は、EDI 文書の最初の 3 文字を調べて文書で使用されるプロトコルを判別します。その後、プロトコル・タイプから送信側および受信側情報を判別します。詳細については、243 ページの『EDI ルーティングの概要』を参照してください。

3. WebSphere Data Interchange は、キューから EDI 文書を読み取り、その EDI 文書に対してアンラップ、検証、および変換タスクを実行します。

注: WebSphere Data Interchange は、必要なマップ、取引パートナー・プロファイル、およびその他の情報で構成する必要があります。詳しくは、WebSphere Data Interchange 資料を参照してください。

4. WebSphere Data Interchange は、この文書をバックエンド・システムに送信します。バックエンド・システムが WebSphere InterChange Server の場合、WebSphere Data Interchange は、ビジネス・オブジェクトを作成し、InterChange Server 内のコラボレーションを呼び出すために、WebSphere Business Integration Adapter for MQ に文書を送信します。

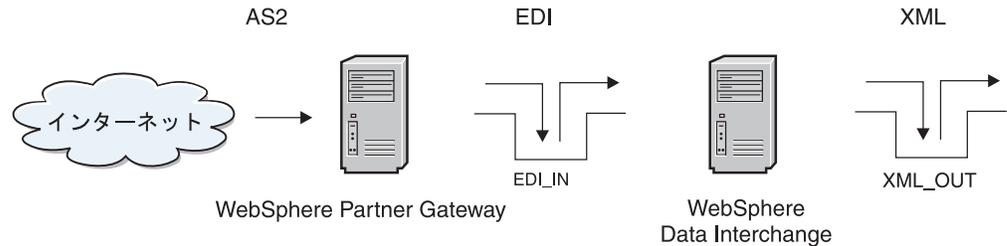


図 40. WebSphere Partner Gateway からの EDI 文書

図 40 では、外部パートナーが WebSphere Partner Gateway に EDI 文書を AS パッケージで送信し、その後 WebSphere Partner Gateway がそれを WebSphere Data Interchange サイドの EDI\_IN キューに送信します。リモート・キュー、伝送キュー、受信先キュー (例では EDI\_IN)、送信側チャンネルおよび受信側チャンネルが、WebSphere Partner Gateway に送信されたメッセージが EDI\_IN キューに転送されるようにセットアップされていなければなりません。WebSphere Data Interchange サーバーは、EDI 文書を取り出し、ユーザー・プロファイル、マッピングなどを検索して、文書を XML に変換し、XML\_OUT キューに入れます。

## WebSphere Data Interchange からの文書の受信方法

WebSphere Partner Gateway が WebSphere Data Interchange からの EDI 文書を受信する際は、以下の手順が実行されます。

1. WebSphere Data Interchange は、EDI 文書をキューに入れます。
2. WebSphere Partner Gateway は、キューからメッセージを読み取ります。

注: WebSphere Partner Gateway は、243 ページの『EDI ルーティングの概要』に従って文書の送付方法を判別します。

3. WebSphere Partner Gateway は、該当する外部パートナーに文書を送付します。

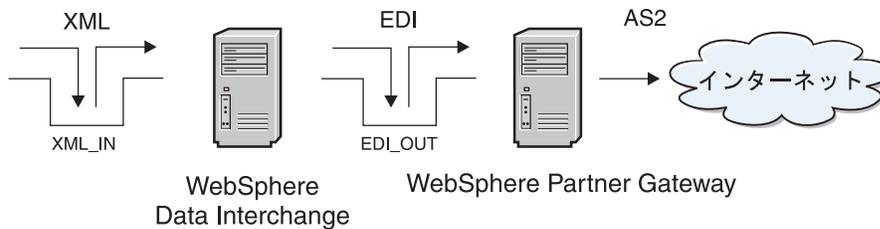


図 41. WebSphere Partner Gateway への EDI 文書の送信

図 41 では、WebSphere Data Interchange が変換を実行するために、XML 文書は XML\_IN キューに入れられます。ユーザー・プロファイル、マッピングなどはすでに実行されていることを前提としています。有効な XML 文書を受信すると、WebSphere Data Interchange はそれを EDI 形式に変換して、出力を EDI\_OUT キュー (リモート・キュー) に入れます。伝送キュー、送信側および受信側チャンネル、WebSphere Partner Gateway サイドの受信先キューがセットアップされていることも前提とします。文書を受信すると、WebSphere Partner Gateway はそれを外部パートナーに送付します。

## 本章で使用するサンプル・シナリオ

この章では、2 つの取引パートナー間での EDI 文書の交換をセットアップする手順について説明します。EDI 文書はインターネットで送信され、AS2 (HTTP を使用) が通信プロトコルとして使用されます。

このサンプルでは、パートナー 1 とパートナー 2 が取引パートナーです。図 42 に、2 つのパートナーの構成を示します。

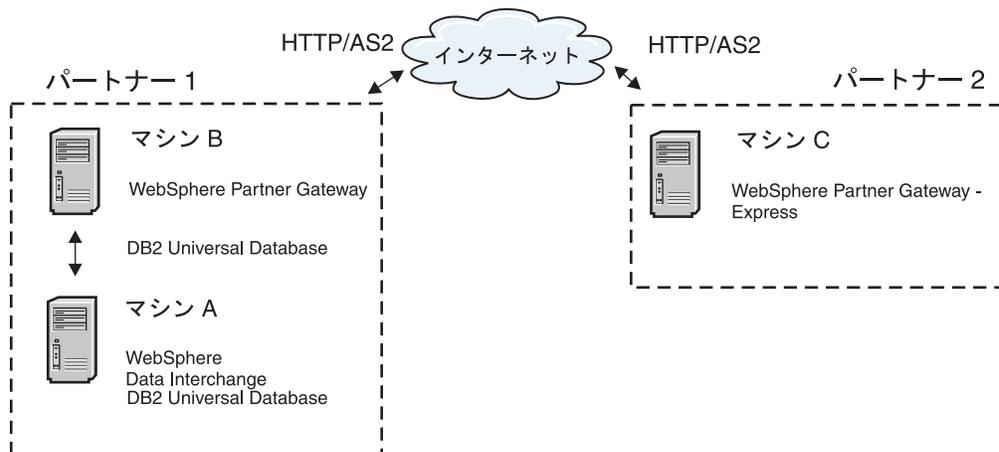


図 42. サンプル・シナリオでの 2 つのパートナーの構成

3 台のコンピューターには、以下のソフトウェアがインストールされています。

- コンピューター A には、WebSphere Data Interchange Server 3.2 と WebSphere Data Interchange Client 3.2 が前提ソフトウェアと共に組み込まれています。
- コンピューター B には、WebSphere Partner Gateway Enterprise Edition が前提ソフトウェアと共に組み込まれています。

- コンピューター C には、WebSphere Partner Gateway - Express が組み込まれています。

ソフトウェア前提条件の詳細については、「*WebSphere Partner Gateway インストール・ガイド*」と WebSphere Data Interchange 資料を参照してください。

この例では、パートナー 1 は 2 台のコンピューターを運用しています。コンピューター A には、WebSphere MQ と WebSphere Data Interchange Server の両方がインストールされています。コンピューター B には、WebSphere MQ と WebSphere Partner Gateway Enterprise Edition がインストールされています。コンピューター B は、2 つの取引先間の通信をサポートします。

WebSphere Data Interchange は WebSphere MQ との統合をサポートして、広範囲のエンタープライズ・アプリケーションやビジネス・プロセス・エンジンとの相互協調処理を可能にします。WebSphere Partner Gateway は、WebSphere MQ を JMS プロバイダーとして使用します。したがって、WebSphere Data Interchange と WebSphere Partner Gateway 間の統合は JMS API クライアントに向けた MQ メッセージを使用します。

WebSphere Partner Gateway は、AS2 プロトコルを使用してインターネットで EDI トランザクションを通信するために使用されます。

この例では、パートナー 2 は WebSphere Partner Gateway - Express を使用して AS2 経由のトランザクションを受信し、トランザクションおよび肯定応答を処理する独自の WebSphere Data Interchange 環境があります。

本章全体で、このサンプル・シナリオで使用されるコンピューターの構成についての詳細が説明されます。メッセージ・フローは双方向で、送信および受信の両方の成果物が含まれています。

## WebSphere Data Interchange との統合の計画

### WebSphere Partner Gateway がサポートする WebSphere Data Interchange のバージョン

WebSphere Data Interchange のサポートされるバージョンは、CSD 22 が適用されたバージョン 3.2 です。WebSphere Data Interchange とのインターフェースとして、WebSphere MQ が使用されます。WebSphere MQ のサポートされるバージョンは、最新のフィックスパックが適用されたバージョン 6.0 です。

### メッセージ交換用の環境の構成

WebSphere Data Interchange と WebSphere Partner Gateway との間で通信を行うには、以下の設定タスクと構成タスクを実行する必要があります。

- 210 ページの『WebSphere MQ 通信の構成』
- 211 ページの『WebSphere Data Interchange の構成』
- 216 ページの『JMS 環境の設定』
- 217 ページの『WebSphere Partner Gateway Enterprise Edition の構成』

## WebSphere MQ 通信の構成

環境をセットアップする際の最初のステップは、WebSphere MQ の相互通信を構成することです。相互通信とは、あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することを意味します。最初のステップは、WebSphere Data Interchange システムと WebSphere Partner Gateway システム用のキュー・マネージャー（および関連オブジェクト）を定義することです。両方向にメッセージを送信する場合は、ソース・キュー・マネージャーと受信側キュー・マネージャーを両システムにセットアップします。ソース・キュー・マネージャーに、送信側チャンネル、リモート・キュー定義、および伝送キューを定義します。受信側キュー・マネージャーに、受信側チャンネルと受信側キューを定義します。

**注:** キュー・マネージャーの定義の詳細については、WebSphere MQ 資料を参照してください。

ここでは、サンプル・シナリオに必要なキュー・マネージャーと関連オブジェクトのセットアップに使用する値を示します。シナリオでは、コンピューター A およびコンピューター B の両方に WebSphere MQ V5.3 がインストールされています。最初のステップは、WebSphere Data Interchange と WebSphere Partner Gateway Enterprise Edition でそれぞれ使用するコンピューター A とコンピューター B の両方に、キュー・マネージャーを作成します。

**注:** 使用する WebSphere Data Interchange キュー・マネージャーは、WDI アダプター・アプリケーションを使用して WebSphere Data Interchange Server を起動するように構成します。

- コンピューター A では、WebSphere Data Interchange で使用するよう定義されたキュー・マネージャーを使用します。本章ではこれ以降、このキュー・マネージャーを WDI32\_QM と呼びます。
- コンピューター B では、WebSphere Partner Gateway Enterprise Edition の初期インストールおよび構成時に作成されたキュー・マネージャーを使用します。本章ではこれ以降、このキュー・マネージャーを HUB\_QM と呼びます。

WebSphere MQ を使用してあるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信するには、次のオブジェクトを定義します。

- ソース・キュー・マネージャー側:
  - 送信側チャンネル
  - リモート・キュー定義
  - 伝送キュー
- 受信側キュー・マネージャー側:
  - 受信側チャンネル
  - 受信側キュー

サンプル・シナリオでは、コンピューター A とコンピューター B の両方とも送信側および受信側として動作します。したがって、各コンピューターにいくつかのオブジェクトを定義する必要があります。

211 ページの表 84 は、コンピューター A とコンピューター B を送信側および受信側として設定する場合に作成するオブジェクトを示しています。

表 84. 作成する WebSphere MQ オブジェクト

WebSphere MQ オブジェクト	コンピューター A	コンピューター B
キュー・マネージャー	WDI32_QM	HUB_QM
送信側チャンネル	TO.HUB60	TO.WDI32
受信側チャンネル	TO.WDI32	TO.HUB60
リモート・キュー	EDI_OUT_A	EDI_OUT_B
伝送キュー	XMITQ_A	XMITQ_B
ローカル・キュー	EDI_IN_A	EDI_IN_B
ローカル・キュー	XML_IN_A	XML_IN_B
ローカル・キュー	XML_OUT_A	XML_OUT_B

図 43 は、コンピューター A とコンピューター B 間のメッセージ・フロー、および表 84 の WebSphere MQ オブジェクトの役割を示したものです。

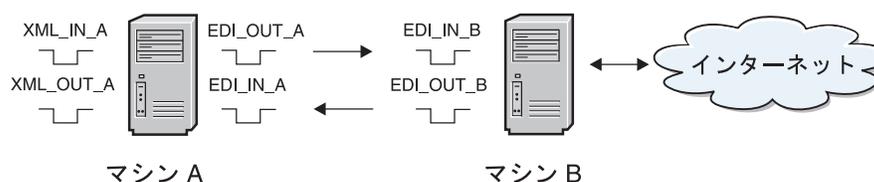


図 43. コンピューター A とコンピューター B 間のメッセージ・フロー

使用する WebSphere MQ プラットフォームにより、さまざまな方法を使用してこれらのオブジェクトを定義できます。例えば、Windows 上の WebSphere MQ Explorer を使用してオブジェクトを定義します。

## WebSphere Data Interchange の構成

WebSphere MQ キューからメッセージを受信し、EDI メッセージをキューに書き込む WebSphere Data Interchange の場合は、WebSphere Data Interchange Client にプロファイルを構成する必要があります。WebSphere Data Interchange Client を使用して次のプロファイルを作成します。このプロファイルについては、これに続くセクションで説明します。

- MQSeries キュー・プロファイル
- ネットワーク・プロファイル
- メールボックス・プロファイル
- サービス・プロファイル

サンプル・シナリオの場合、WebSphere Data Interchange は、WebSphere MQ キュー XML\_IN\_A から XML メッセージを受信し、WebSphere MQ キュー EDI\_OUT\_A に変換結果を書き込みます。これは、XML から EDI への変換と呼ばれます。また、WebSphere Data Interchange は、WebSphere Partner Gateway Enterprise Edition からの EDI を WebSphere MQ キュー EDI\_IN\_A 上で受信し、XML\_OUT\_A に変換結果を書き込みます。

### MQSeries<sup>(R)</sup> キュー・プロファイル:

MQSeries キュー・プロファイルには、WebSphere MQ メッセージ・キューについての情報が含まれます。表 85 に、各プロファイル用に構成するプロパティを示します。

表 85. MQSeries キュー・プロファイルのプロパティ

MQ プロパティ	説明
Queue Profile ID	プロファイルを指定する固有 ID (論理名)
Full Queue Name	WebSphere MQ キューの実際の名前
Queue Manager Name	WebSphere MQ キュー・マネージャーの実際の名前
説明	プロファイルの目的を識別する任意のストリング
Maximum Length	WebSphere MQ に構成された、キューの可能な最大メッセージ
Destructive Reads	選択すると、WebSphere Data Interchange は読み取り時に WebSphere MQ キューからメッセージを除去します。
Syncpoint Control	チェックすると、キュー・メッセージの読み取りおよび書き込みが同期点制御されます。同期点制御が有効な場合、メッセージ・キューの変更は、WebSphere Data Interchange が同期点を発行するまで行われません。

ここでは WebSphere MQ キューを使用しているため、各 キューごとに WebSphere Data Interchange に MQSeries キュー・プロファイルが必要です。つまり、4 つの MQSeries キュー・プロファイル (メッセージ・フローで使用される WebSphere MQ キューごとに 1 つ) を作成します。WebSphere Data Interchange Client のセットアップ域から、次の手順を実行します。

1. XML\_IN\_A および EDI\_OU\_A の MQSeries キュー・プロファイルを作成します。

表 86 は、作成した各 MQSeries キュー・プロファイルに指定された実際のパラメーターを示しています。ここで示したキューは、XML から EDI への変換で使用されます。

表 86. XML\_IN\_A および EDI\_OU\_A の MQSeries キュー・プロファイル

キュー・プロパティ	XML_IN_A の値	EDI_OU_A の値
Queue Profile ID	XML_IN_A	EDI_OU_A
Full Queue Name	XML_IN_A	EDI_OUT_A
Queue Manager Name	WDI32_QM	WDI32_QM
Destructive Reads	チェック済み	チェック済み
Syncpoint Control	チェック済み	チェック済み

注: Queue Profile ID は、最大 8 文字に制限されています。したがって、EDI\_OUT\_A キューのプロファイル ID は EDI\_OU\_A のように設定する必要があります。WebSphere Data Interchange の WebSphere MQ キュー EDI\_OUT\_A への参照では、常に EDI\_OU\_A が使用されます。

2. EDI\_IN\_A と XML\_OU\_A の MQSeries キュー・プロファイルを作成します。  
 表 87 は、EDI から XML への変換で使用される各キューのプロパティを示しています。

表 87. EDI\_IN\_A および XML\_OU\_A の MQSeries キュー・プロファイル

キュー・プロパティ	EDI_IN_A の値	XML_OU_A の値
Queue Profile ID	EDI_IN_A	XML_OU_A
Full Queue Name	EDI_IN_A	XML_OUT_A
Queue Manager Name	WDI32_QM	WDI32_QM
Destructive Reads	チェック済み	チェック済み
Syncpoint Control	チェック済み	チェック済み

### ネットワーク・プロファイル:

ネットワーク・プロファイルで、取引先との通信に使用するネットワークの特性を WebSphere Data Interchange に対して定義します。このシナリオでは、以前に作成された WebSphere MQ キューと通信するネットワーク・プロファイルを作成し、構成します。

表 88 に、各ネットワーク・プロファイル用に構成するプロパティを示します。

表 88. ネットワーク・プロファイルのプロパティ

ネットワーク・プロパティ	説明
Network ID	プロファイルを指定する固有 ID
Communication Routine	ネットワーク・コマンドを作成し、コマンドを処理するネットワーク・プログラムを起動するプログラムの名前
Network Program	要求を処理するために通信ルーチンによって起動されるプログラム
Network Parameters	ネットワーク・プログラムに必要なパラメーター

このサンプル・シナリオでは、前に作成した WebSphere MQ キューと通信するネットワーク・プロファイル (211 ページの『MQSeries<sup>(R)</sup> キュー・プロファイル』を参照) を、以下の手順で作成および構成します。

1. HUB\_IN と呼ばれるネットワーク・プロファイルを作成します。

このネットワーク・プロファイルは、XML から EDI へのシナリオで使用されます。表 89 は、HUB\_IN に指定する実際のパラメーターを示しています。

表 89. HUB\_IN のネットワーク・プロファイル

ネットワーク・プロパティ	HUB_IN プロファイルの値
Network ID	HUB_IN
Communication Routine	VANIMQ
Network Program	EDIMQSR
Network Parameters	SENDMQ=EDI_OU_A RECEIVMQ=XML_IN_A

2. HUB\_OUT と呼ばれる 2 番目のネットワーク・プロファイルを作成します。

このネットワーク・プロファイルは、WebSphere Partner Gateway Enterprise Edition から受信した EDI の変換に使用されます。WebSphere Partner Gateway Enterprise Edition は RFH2 ヘッダーを組み込む WebSphere MQ キューにメッセージを入れるので、2 番目のネットワーク・プロファイルが必要です。表 90 は、HUB\_OUT のプロパティを示しています。

表 90. HUB\_OUT のネットワーク・プロファイル

ネットワーク・プロパティ	HUB_OUT プロファイルの値
Network ID	HUB_OUT
Communication Routine	VANIMQ
Network Program	EDIRFH2
Network Parameters	SENDMQ=XML_OU_A RECEIVEMQ=EDI_IN_A

**メールボックス・プロファイル:**

メールボックス・プロファイルには、WebSphere Data Interchange が、変換される文書を受信する組織内の個人およびグループを識別するために必要な情報が含まれます。表 91 に、各メールボックス・プロファイル用に構成するプロパティを示します。

表 91. メールボックス・プロファイルのプロパティ

メールボックス・プロパティ	説明
Mailbox ID	プロファイルを指定する固有 ID
Network ID	以前に作成されたネットワーク・プロファイルのネットワーク ID

メールボックス・プロファイルは、組織内の個人とグループを識別するために、各 WebSphere MQ キューごとに作成します。

1. 使用する各 WebSphere MQ キューのメールボックス・プロファイルを作成します。

表 92 は、各メールボックス・プロファイルの実際のパラメーターを示しています。

表 92. XML\_IN\_A および EDI\_OU\_A のメールボックス・プロファイル

メールボックス・プロパティ	XML_IN_A の値	EDI_OU_A の値
Mailbox ID	XML_IN_A	EDI_OU_A
Network ID	HUB_IN	HUB_IN
Receive File	XML_IN_A	EDI_OU_A

2. メールボックスの 2 番目のペアを作成します。

表 93 は、それぞれのプロパティーを示しています。

表 93. EDI\_IN\_A および XML\_OU\_A のメールボックス・プロファイル

メールボックス・プロパティー	EDI_IN_A の値	XML_OU_A の値
Mailbox ID	EDI_IN_A	XML_OU_A
Network ID	HUB_OUT	HUB_OUT
Receive File	EDI_IN_A	XML_OU_A

**サービス・プロファイル:**  
このタスクについて

サービス・プロファイルを使用すると、ユーティリティー・コマンドの入力と、そのコマンドの実行時に使用されるすべてのファイルの定義が可能となります。

サンプル・シナリオでは、以下の手順を実行します。

1. XML\_IN\_A の新しいサービス・プロファイルを作成します。「一般」タブで、以下のようにプロパティーを定義します。

- コマンド・チェーンの継続 (Continue Command Chaining): **成功時 (On Success)**
- PERFORM コマンド:  

```
PERFORM TRANSFORM WHERE INFILE(XML_IN_A) SYNTAX(X)
OUTTYPE(MQ)OUTFILE(EDI_OU_A)
```

表 94 は、共通ファイルのプロパティーを示しています。

表 94. XML\_IN\_A の共通ファイル

共通ファイルのプロパティー	値
トラッキング・ファイル	..¥trk¥xml_in.trk
例外ファイル	..¥xex¥xml_in.xex
作業ファイル	..¥wrk¥xml_in.wrk
レポート・ファイル	..¥rpt¥xml_in.rpt
照会ファイル	..¥qry¥xml_in.qry

2. 「出力ファイル (Output Files)」タブに以下を入力します。

- コマンドの名前 (Name in Command): **EDI\_OU\_A**
- システム・ファイル名 (System File Name): **..¥edi¥edi\_out.txt**

注: 文字長の制限のため、EDI\_OUT \_A ではなく EDI\_OU\_A が使用されます。

3. EDI\_IN\_A の 2 番目のサービス・プロファイルを作成します。「一般」タブで以下のようにプロパティーを定義します。

- コマンド・チェーンの継続 (Continue Command Chaining): **成功時 (On Success)**
- PERFORM コマンド:  

```
PERFORM TRANSFORM WHERE INFILE(EDI_IN_A) SYNTAX(E)
OUTTYPE(MQ) OUTFILE(XML_OU_A)
```

表 95 は、共通ファイルのプロパティを示しています。

表 95. EDI\_IN\_A の共通ファイル

共通ファイルのプロパティ	値
トラッキング・ファイル	.. <i>Y</i> trk <i>Y</i> edi_in.trk
例外ファイル	.. <i>Y</i> xex <i>Y</i> edi_in.xex
作業ファイル	.. <i>Y</i> wrk <i>Y</i> edi_in.wrk
レポート・ファイル	.. <i>Y</i> rpt <i>Y</i> edi_in.rpt
照会ファイル	.. <i>Y</i> qry <i>Y</i> edi_in.qry

4. 「出力ファイル (Output Files)」タブに以下を入力します。

- コマンドの名前 (Name in Command): **XML\_OU\_A**
- システム・ファイル名 (System File Name): **..*Y*xml*Y*xml\_out.txt**

注: 文字長の制限のため、XML\_OUT\_A ではなく XML\_OU\_A が使用されます。この制限は、WebSphere Interchange Server の CSD10 で除去されました。

**データ変換マップのインポートおよびコンパイル:** 前のセクションでの説明に従ってプロファイルを作成すると、データを変換するのに必要なマップをインポートできます。その後変換マップをコンパイルして、それぞれの規則を設定します。これらのタスクを実行するには、WebSphere Data Interchange Client を使用します。詳しくは、WebSphere Data Interchange 資料を参照してください。

## JMS 環境の設定

この章の最初の方で説明したように、WebSphere Partner Gateway Enterprise Edition は、Java Message Service (JMS) の WebSphere MQ 実装を使用して WebSphere Data Interchange と統合することができます。

注: あるいは、LDAP または WebSphere Application Server を JNDI プロバイダーとして使用することができます。

ここでは、コンピューター B 上への JMS 環境の作成に関する手順の概要を説明します。

- 『JMSAdmin の構成』
- 217 ページの『JMS オブジェクトの作成』

Java 用の WebSphere MQ クラスおよび JMS 用の WebSphere MQ クラスは、WebSphere MQ for Windows バージョン 5.3 に組み込まれています。

### JMSAdmin の構成:

JNDI に JMS オブジェクトを作成するには、WebSphere MQ で使用できる JMSAdmin ツールを使用します。JMSAdmin.config というデフォルトの構成ファイルの作成方法について詳しくは、「WebSphere Partner Gateway ハブ構成ガイド」を参照してください。

このチュートリアルの JMS オブジェクトを作成するには、次の手順を実行します。

1. ファイル・ベース JNDI プロバイダーを使用するには、JMSAdmin.config ファイルに次の行が含まれていることを確認します。

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.ReffSContextFactory
PROVIDER_URL=file:/opt/mqm/java/JNDI
```

2. JNDI ディレクトリーが存在しない場合は、次のディレクトリーの下に JNDI ディレクトリーを作成します。

```
/opt/mqm/java/bin
```

JMSAdmin ツールを起動する前に、CLASSPATH に以下のエントリーが含まれていることを確認してください。

```
/opt/mqm/java/lib/jms.jar
/opt/mqm/java/lib/com.ibm.mq.jar
/opt/mqm/java/lib/com.ibm.mqjms.jar
/opt/mqm/java/lib/jta.jar
/opt/mqm/java/lib/connector.jar
/opt/mqm/java/lib/jndi.jar
/opt/mqm/java/lib/providerutil.jar
/opt/mqm/java/lib/fscontext.jar
```

注: Linux<sup>(TM)</sup> に関連する上記エントリーは、ファイル・ベース JNDI を使用することを前提とします。

### JMS オブジェクトの作成:

必要な JMS オブジェクトを作成するには、JMSAdmin ツールを使用します。サンプル・シナリオでは、以下のようにします。

1. 新規コンテキストを定義します。

```
DEF CTX(WdiJms)
```

2. 新規コンテキストに切り替えます。

```
CHG CTX(WdiJms)
```

3. キュー接続ファクトリーを定義します。

```
DEF QCF(HUB60_QM_QCF) TRAN(CLIENT) HOST(IP_COMPUTER_B)
PORT(9999) CHAN(java.channel) QMANAGER(HUB60_QM)
```

4. EDI\_IN\_B キューを定義します。

```
DEF Q(EDI_IN_B) QMANAGER(HUB60_QM) QUEUE(EDI_IN_B)
```

5. EDI\_OUT\_B キューを定義します。

```
DEF Q(EDI_OUT_B) QMANAGER(HUB60_QM) QUEUE(EDI_OUT_B)
```

6. JMSAdmin セッションを終了します。

```
END
```

## WebSphere Partner Gateway Enterprise Edition の構成

WebSphere Partner Gateway は、別々の外部パートナーと内部プロセス間の通信レイヤーです。EDI 文書を処理するために WebSphere Partner Gateway をセットアップすると、以下を実行するように構成できます。

- WebSphere Data Interchange との間で EDI 文書を送受信する
- AS2 を使用して外部取引先と EDI トランザクションを通信する

WebSphere Partner Gateway Enterprise Edition および Advanced Edition の構成方法の詳細については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください

さい。ここでは、サンプル・シナリオで説明されている WebSphere Partner Gateway Enterprise Edition の構成例を説明します。以下の手順を説明します。

1. 『パートナーの作成』
2. 220 ページの『B2B 機能の設定』
3. 221 ページの『宛先の作成』
4. 223 ページの『インタラクションの定義』
5. 224 ページの『パートナー接続の作成』
6. 225 ページの『レシーバーの作成』

注: WebSphere Partner Gateway - Express の構成方法については、226 ページの『WebSphere Partner Gateway - Express の構成』を参照してください。

**パートナーの作成:** パートナー・プロファイルで、会社をシステムに識別させます。WebSphere Partner Gateway Enterprise Edition のコミュニティー・コンソールで、パートナー 1 とパートナー 2 のパートナーを作成します。

#### パートナー 1 のパートナーの作成:

##### このタスクについて

コンピューター A とコンピューター B を表すパートナー・プロファイルを作成します (これらのマシンは、パートナー 1 が所有する 2 つのシステムです)。

このパートナー・プロファイルを作成するには、以下の手順を実行します。

1. WebSphere Partner Gateway コミュニティー・コンソールを開きます。
2. **hubadmin** としてログインします。
3. 「アカウント管理」>「プロファイル」>「パートナー」をクリックします。
4. 「作成」をクリックして、次の表 96 にリストした詳細内容を入力します。

表 96. パートナー 1 のパートナー・プロパティ

フィールド名	値
会社ログイン名	partnerOne
パートナー表示名	パートナー 1
パートナー・タイプ	内部パートナー
状況	使用可能
ベンダー・タイプ	その他
Web サイト	http://IP_COMPUTER_A  IP_COMPUTER_A は、コンピューター A のインターネット・プロトコル (IP) アドレスです。
ビジネス ID タイプ (Business ID Type)	フリー・フォーム
ビジネス ID 識別子 (Business ID Identifier)	123456789
IP アドレス宛先タイプ (IP Address Destination Type)	実動

表 96. パートナー 1 のパートナー・プロパティ (続き)

フィールド名	値
IP アドレス (IP Address)	<p><i>IP_COMPUTER_A</i></p> <p><i>IP_COMPUTER_A</i> は、コンピューター A のインターネット・プロトコル (IP) アドレスです。</p>

注: ビジネス ID タイプとビジネス ID 識別子を作成するには、まず、ビジネス ID の下の「新規」ボタンをクリックします。ビジネス ID は、固有でなければなりません。同様に、IP アドレスに関連する詳細内容を作成するには、IP アドレス・ヘッダーの下の「新規」ボタンをクリックします。

5. 「保管」をクリックします。

WebSphere Partner Gateway Enterprise Edition は、(218 ページの表 96 で定義された) ビジネス ID 識別子を使用して、文書の送信側または受信側を識別します。ANSI X12 EDI トランザクションが受信されると、トランザクションのソースおよびターゲットを判別するために交換送信側および受信側データが読み取られます。

重要: 後で必要になるので、パートナー 1 の管理者パスワードをメモしておいてください。パートナー 1 としてコミュニティー・コンソールにログオンするときに、パスワードの入力と、次にパスワードの変更を求められます。

**パートナー 2 のパートナーの作成:**

**このタスクについて**

次に、パートナー 2 を表す外部パートナーを作成します。パートナーを作成するには、次の手順を実行します。

1. 「アカウント管理」>「プロファイル」>「パートナー」をクリックします。
2. 「作成」をクリックします。
3. 以下の表 97 に示す値を入力します。

表 97. パートナー 2 のパートナー・プロパティ

フィールド名	値
会社ログイン名	partnerTwo
パートナー表示名	パートナー 2
パートナー・タイプ	パートナー
状況	使用可能
ベンダー・タイプ	その他
Web サイト	<p><i>http://IP_COMPUTER_C</i></p> <p><i>IP_COMPUTER_C</i> は、コンピューター C のインターネット・プロトコル (IP) アドレスです。</p>
ビジネス ID タイプ (Business ID Type)	フリー・フォーム
ビジネス ID 識別子 (Business ID Identifier)	987654321
IP アドレス宛先タイプ (IP Address Destination Type)	実動

表 97. パートナー 2 のパートナー・プロパティ (続き)

フィールド名	値
IP アドレス (IP Address)	<p><code>IP_COMPUTER_C</code></p> <p><code>IP_COMPUTER_C</code> は、コンピューター C のインターネット・プロトコル (IP) アドレスです。</p>

4. 「保管」をクリックします。

**重要:** 後で必要になるので、パートナー 2 の管理者パスワードをメモしておいてください。パートナー 2 としてコミュニティー・コンソールにログオンするときに、パスワードの入力と、次にパスワードの変更を求められます。

#### B2B 機能の設定:

##### このタスクについて

各パートナーの B2B 機能を、コミュニティー・コンソールを使用して WebSphere Partner Gateway Enterprise Edition に定義します。各パートナーの B2B 機能を定義すると、パートナー間の特定のビジネス・コラボレーション・タイプをサポートするために使用する有効な文書フロー定義を指定できます。

#### パートナー 1 の B2B 機能の設定:

##### このタスクについて

パートナー 1 の B2B 機能を定義するには、以下の手順を実行します。

1. 「アカウント管理」 > 「プロファイル」 > 「パートナー」をクリックします。
2. 「検索」をクリックして、システムに定義されているすべてのパートナーのリストを表示します。
3. パートナー 1 の横の「詳細の表示」アイコンをクリックして、「B2B 機能」をクリックします。

B2B 機能をアクティブに設定するには、「役割はアクティブではありません (Role is not active)」アイコンをクリックします。このサンプルの目的として、シナリオの実装に必要な B2B 機能のみを構成します。

パートナー 1 のソースおよびターゲット・パッケージ化を「なし」に設定するには、次の手順を実行します。

1. 「パッケージのソース設定: なし (Set Source for Package: None)」の下の「役割はアクティブではありません (Role is not active)」アイコンをクリックして有効にします。「ターゲットの設定」について、この手順を繰り返します。
2. 「拡張 (Expand)」アイコンをクリックして、ドリルダウンします。
3. ソースとターゲットの両方で、「プロトコル: EDI-X12 (すべて)」に対して「役割はアクティブではありません (Role is not active)」アイコンをクリックします。
4. 「拡張 (Expand)」アイコンをクリックします。
5. ソースとターゲットの両方で、「文書フロー: ISA (すべて) (Document Flow: ISA (ALL))」に対して「役割はアクティブではありません (Role is not active)」アイコンをクリックします。

## パートナー 2 の B2B 機能の設定:

### このタスクについて

パートナー 2 の B2B 機能を定義するには、以下の手順を実行します。

1. 「アカウント管理」>「プロファイル」>「パートナー」をクリックします。
2. 「検索」をクリックして、システムに定義されているすべてのパートナーのリストを表示します。
3. パートナー 2 の横の「詳細の表示」アイコンをクリックして、「B2B 機能」をクリックします。

パートナー 2 のソースおよびターゲット・パッケージ化を AS に設定するには、以下の手順を実行します。

1. 「パッケージ: AS」に対する「ソースの設定」の下で、「役割はアクティブではありません」アイコンをクリックして有効にします。「ターゲットの設定」について、この手順を繰り返します。
2. 「拡張 (Expand)」アイコンをクリックして、ドリルダウンします。
3. ソースとターゲットの両方で、「プロトコル: EDI-X12 (すべて)」に対して「役割はアクティブではありません (Role is not active)」アイコンをクリックします。
4. 「拡張 (Expand)」アイコンをクリックします。
5. ソースとターゲットの両方で、「文書フロー: ISA (すべて) (Document Flow: ISA (ALL))」に対して「役割はアクティブではありません (Role is not active)」アイコンをクリックします。

次に、パートナー 2 に送信された AS2 のメッセージ処理通知 (MDN) が正しいアドレスに返送されるように、パートナー 2 の AS 定義を更新します。

1. 「編集」アイコンをクリックします。
2. AS MDN の電子メール・アドレスを入力します。

このアドレスは、AS1 の MDN を受信する際に使用されます。

3. 以下のように、AS MDN の HTTP URL を入力します。

```
http://IP_COMPUTER_B:PORT/bcgreceiver/submit
```

注: AS2 に定義する URL では、この章の後半で AS2 受信側に定義するのと同じパラメーターを使用します。

## 宛先の作成:

### このタスクについて

WebSphere Partner Gateway の宛先は、別のネットワークの入り口としての役割を果たすネットワーク・ポイントを定義します。宛先には、Enterprise Application Integration (EAI) レイヤーへの文書の配送方法を WebSphere Partner Gateway に指示する情報が含まれます。

## パートナー 1 の宛先の作成:

## このタスクについて

パートナー 2 は、AS2 を使用してパートナー 1 に EDI 文書を送信します。パートナー 1 の宛先は、AS2 を通じて受信した EDI 文書を JMS キュー経由で WebSphere Data Interchange まで送信するために使用されます。この EDI 文書は、WebSphere Data Interchange で変換されます。

パートナー 1 の新しい宛先を作成するには、以下の手順を実行します。

1. 「アカウント管理」>「プロファイル」>「パートナー」をクリックします。
2. 「検索」をクリックします。
3. 「詳細の表示」アイコンをクリックしてパートナー 1 を選択し、さらに「宛先」を選択します。
4. 「作成」をクリックして、パートナー 1 の新規宛先を作成します。
5. この新しい宛先の値を表 98 のように入力します。

表 98. パートナー 1 宛先のプロパティ

フィールド名	値
宛先名	JMStoPartnerOne
トランスポート	JMS
レシーバー URI (Receiver URI)	file:///opt/mqm/java/JNDI/WdiJms
JMS ファクトリー名	HUB60_QM_QCF
JMS メッセージ・クラス	TextMessage
JMS メッセージ・タイプ	TextMessage
JMS キュー名	EDI_OUT_B
JMS JNDI ファクトリー名	com.sun.jndi.fscontext.RefFSContextFactory

6. 「保管」をクリックします。

JMStoPartnerOne をパートナー 1 のデフォルト宛先に設定します。

1. 「デフォルト宛先の表示」をクリックします。
2. 「実動」リストから、**JMS2toPartnerOne** を選択します。
3. 「保管」をクリックします。

## パートナー 2 の宛先の作成:

### このタスクについて

パートナー 1 は JMS キューを通じて WebSphere Partner Gateway Enterprise Edition に EDI 文書を送信します。パートナー 2 の宛先は、受信した EDI 文書を AS2 を介してパートナー 2 に送信するために使用されます。

パートナー 2 の新しい宛先を作成するには、以下の手順を実行します。

1. 「アカウント管理」>「プロファイル」>「パートナー」をクリックします。
2. 「検索」をクリックします。
3. 「詳細の表示」アイコンをクリックしてパートナー 2 を選択し、さらに「宛先」を選択します。
4. 「作成」をクリックして、パートナー 2 の新規宛先を作成します。

5. この宛先に入力する値を、表 99 に示します。

表 99. パートナー 2 宛先のプロパティ

宛先名	AS2toPartnerTwo
トランスポート	HTTP/1.1
レシーバー URI (Receiver URI)	http://IP_COMPUTER_C/input/AS2
ユーザー名	partnerOne
パスワード	partnerOne

6. 「保管」をクリックします。

注: 上で入力したユーザー名とパスワードは、WebSphere Partner Gateway - Express で定義した、HTTP 用のインバウンド・パートナー・マッピング・メソッドを参照します。

WebSphere Partner Gateway - Express でこれらのプロパティを設定する例については、226 ページの『WebSphere Partner Gateway - Express の構成』を参照してください。

AS2toPartnerTwo が、「使用可能」の状況でオンラインとして表示されています。

以下の手順を実行し、AS2toPartnerTwo をパートナー 2 のデフォルト宛先に設定します。

1. 「デフォルト宛先の表示」をクリックします。
2. 「実動」リストから、**AS2toPartnerTwo** を選択します。
3. 「保管」をクリックします。

#### インタラクションの定義: このタスクについて

文書フロー定義とは、パートナーの文書処理機能を定義する「メタ情報」の集合です。システムでビジネス文書进行处理するには、複数の文書フロー定義をリンクしてインタラクションを作成する必要があります。

パートナー 1 とパートナー 2 との間のインタラクションを作成するには、以下の手順を実行します。

1. 「ハブ管理」>「ハブ構成」>「文書定義」をクリックします。
2. 「インタラクションの管理」をクリックし、次に「インタラクションの作成」をクリックします。
3. ソース列から、以下を選択します。
  - a. パッケージ: なし
  - b. プロトコル: **EDI-X12** (すべて)
  - c. 文書フロー: **ISA** (すべて)
4. レシーバー列から、以下を選択します。
  - a. パッケージ: **AS**
  - b. プロトコル: **EDI-X12** (すべて)

- c. 文書フロー: **ISA (すべて)**
- 5. アクションに「パススルー」を設定します。
- 6. 「保管」をクリックします。
- 7. 「**インタラクションの作成**」を再度クリックします。
- 8. ソース列から、以下を選択します。
  - a. パッケージ: **AS**
  - b. プロトコル: **EDI-X12 (すべて)**
  - c. 文書フロー: **ISA (すべて)**
- 9. レシーバー列から、以下を選択します。
  - a. パッケージ: なし
  - b. プロトコル: **EDI-X12 (すべて)**
  - c. 文書フロー: **ISA (すべて)**
- 10. アクションに「パススルー」を設定します。
- 11. 「保管」をクリックします。

**パートナー接続の作成:**  
このタスクについて

パートナー接続とは、システムが、内部パートナーとそのさまざまなパートナーとの間で文書を処理し、送付できるようにするメカニズムです。接続には、各文書フローを適切に交換するために必要な情報が含まれます。

パートナー 1 とパートナー 2 との間のパートナー接続を作成するには、以下の手順を実行します。

1. メインメニューの「**アカウント管理**」をクリックし、水平ナビゲーション・バーの「**パートナーの接続**」をクリックします。
2. 「ソース」リストから「**パートナー 1 (Partner One)**」を選択します。
3. 「ターゲット」リストから「**パートナー 2 (Partner Two)**」を選択します。
4. 「**検索**」をクリックします。
5. 「**アクティブ化**」ボタンをクリックして、下に表示されるパートナー接続を活動化します。これにより、表 100 に示す **B2B** 機能が表示されます。

表 100. パートナー 1 からパートナー 2 へのパートナー接続の活動化

文書フロー・タイプ	ソース	ターゲット
パッケージ	なし (該当せず)	AS (該当せず)
プロトコル	EDI-X12 (すべて)	EDI-X12 (すべて)
文書フロー	ISA (すべて)	ISA (すべて)

パートナー 2 がソースでパートナー 1 がターゲットであるようなパートナー接続を作成するには、以下の手順を実行します。

1. メインメニューの「**アカウント管理**」をクリックし、水平ナビゲーション・バーの「**パートナーの接続**」をクリックします。
2. 「ソース」リストから「**パートナー 2 (Partner Two)**」を選択します。
3. 「ターゲット」リストから「**パートナー 1 (Partner One)**」を選択します。

4. 「検索」をクリックします。
5. 表 101 に示す詳細情報を指定して接続を活動化します。

表 101. パートナー 2 からパートナー 1 へのパートナー接続の活動化

文書フロー・タイプ	ソース	ターゲット
パッケージ	AS (該当せず)	なし (該当せず)
プロトコル	EDI-X12 (すべて)	EDI-X12 (すべて)
文書フロー	ISA (すべて)	ISA (すべて)

#### レシーバーの作成: このタスクについて

「レシーバー・リスト」ビューに表示されるロケーション情報をもとに、WebSphere Partner Gateway の文書マネージャーは、着信文書のトランスポート・タイプに基づいて、文書を適切なシステム・ロケーションから取り出すことができます。トランスポート・タイプに基づいて別のレシーバー構成を作成できます。その後文書マネージャーは、複数の Web、FTP、および POP メール・サーバー (内部ディレクトリーおよび JMS キューを含む) の文書リポジトリー・ロケーションをポーリングして着信文書を探します。

文書マネージャーが事前定義のレシーバーに基づいてロケーションから文書を検索すると、ルーティング・インフラストラクチャーがチャンネル構成に基づいて文書を処理できます。

WebSphere Data Interchange からの EDI トランザクションを受信するには、以下の手順で新しい JMS 受信側を作成します。

1. 「ハブ管理」>「ハブ構成」>「レシーバー」をクリックします。
2. 「レシーバーの作成」をクリックします。
3. 表 102 に示すプロパティを割り当てます。

表 102. JMS を通じて受信する場合の受信側プロパティ

レシーバー・プロパティ	値
レシーバー名	WdiJmsListener
トランスポート	JMS
宛先タイプ	実動
JMS プロバイダー URL	file:///opt/mqm/java/JNDI/WdiJms
JMS キュー名	EDI_IN_B
JMS ファクトリー名	HUB60_QM_QCF
JNDI ファクトリー名	com.sun.jndi.fscontext.ReffSContextFactory

パートナー 2 から AS2 を使用して EDI を受信するには、2 番目の受信側が必要です。この受信側を作成するには、以下の手順を実行します。

1. 「ハブ管理」>「ハブ構成」>「レシーバー」をクリックします。
2. 「レシーバーの作成」をクリックします。

3. 次の表 103 からプロパティを割り当てます。

表 103. AS2 を通じて受信する場合の受信側プロパティ

レシーバー名	HubAS2Listener
トランスポート	HTTP/S
宛先タイプ	実動
URI	/bcgreceiver/submit 注: HTTP/S の受信用の URI は、常に /bcgreceiver で始まる必要があります。

4. 「保管」をクリックします。

## WebSphere Partner Gateway - Express の構成

このセクションでは、外部パートナーの環境を構成する手順について説明します。ここでは、WebSphere Partner Gateway - Express システムを使用してこの環境を操作します。本章で説明するサンプル・シナリオでは、パートナー 2 は HTTP AS2 を使用した EDI の送受信に WebSphere Partner Gateway - Express を使用します。

HTTP AS2 を通じて EDI を正常に受信するには、以下の手順を実行します。

- 『ユーザー・プロファイルの構成』
- 227 ページの『パートナー 1 のパートナーの作成』
- 227 ページの『パートナー 1 のパートナーの構成』

### ユーザー・プロファイルの構成

#### このタスクについて

最初のステップとして、パートナー 2 のプロファイルを WebSphere Partner Gateway - Express で作成する必要があります。パートナー 2 のプロファイルを作成するには、以下の手順を実行します。

- メインメニューの「**構成 (Configuration)**」をクリックします。
- 水平ナビゲーション・バーの「**ユーザー・プロファイル**」をクリックします。
- 表 104 に示す詳細情報を入力します。

表 104. ユーザー・プロファイルの詳細

受信アドレス無保護ドメイン (Receipt Address Unsecure Domain)	IP_COMPUTER_C  IP_COMPUTER_C は、WebSphere Partner Gateway - Express が稼働しているコンピューター C のインターネット・プロトコル (IP) アドレスです。
受信アドレス無保護ポート (Receipt Address Unsecure Port)	80  80 は、WebSphere Partner Gateway - Express で使用するためにインストール時に割り当てられたポートです。
AS2 送信側 ID (AS2 Sender ID)	987654321

表 104. ユーザー・プロファイルの詳細 (続き)

ビジネス ID タイプ (Business ID Type)	DUNS
ビジネス ID (Business Identifier)	987654321

4. 「保管」をクリックします。

## パートナー 1 のパートナーの作成 このタスクについて

パートナー 1 は、WebSphere Partner Gateway - Express のパートナーとして認識される必要があります。パートナー 1 をパートナーとして作成するには、以下の手順を実行します。

1. メインメニューの「構成 (Configuration)」をクリックします。
2. 水平ナビゲーション・バーの「パートナー」をクリックします。
3. 「パートナーの作成 (Create Partners)」ボタンをクリックします。
4. 次の値を割り当てます。
  - a. パートナー名: **partnerOne**
  - b. AS2 パートナー ID (AS2 Partner ID): **123456789**
5. 「保管」をクリックします。

「パートナーの管理 (Manage Partners)」ビューで、パートナー 1 の詳細を表示できます。

## パートナー 1 のパートナーの構成 このタスクについて

パートナー 1 のパートナーが存在する場合は、パートナー 1 を AS2 および HTTP に対応するように構成する必要があります。この構成では、WebSphere Partner Gateway - Express が HTTP と AS2 をパートナー 1 に送受信する際に必要なパラメーターを識別します。

HTTP と AS2 に対応するようにパートナー 1 を構成するには、以下の手順を実行します。

1. メインメニューの「構成 (Configuration)」をクリックします。
2. 水平ナビゲーション・バーの「AS2」をクリックします。
3. 「選択済みパートナー (Selected Partner)」リストから「**partnerOne**」を選択して、「編集」をクリックします。
4. パートナー 1 のアウトバウンド宛先アドレスを次のように定義します。

```
http://IP_COMPUTER_B:7080/bcgreceiver/submit
```

ここで *IP\_COMPUTER\_B* はコンピューター B の IP アドレスです。

5. 「保管」をクリックします。
6. 水平ナビゲーション・バーの「HTTP」をクリックします (**partnerOne** は選択済みパートナーとしてそのまま表示されます)。
7. 「編集」をクリックします。
8. インバウンド・ユーザー名とパスワードを設定します。

ユーザー名: **partnerOne**

パスワード: **partnerOne**

これらはコンピューター B の WebSphere Partner Gateway Enterprise Edition にパートナー 2 のデフォルト宛先を作成するサンプル・ステップで以前に参照したものです。

9. アウトバウンド宛先アドレスを、次の値に設定します。

`http://IP_COMPUTER_B:7080/bcgreceiver/submit`

10. 「保管」をクリックします。

**重要:** WebSphere Partner Gateway - Express にこれらの変更を行った後、コンソールからログアウトして宛先を停止します。すべての変更内容を有効にするために宛先とコンソールを再始動します。

## 要約

本章では、WebSphere Partner Gateway が WebSphere Data Interchange と対話するプロセスを説明しました。また、208 ページの『本章で使用するサンプル・シナリオ』で説明するサンプル・シナリオのセットアップ手順も示しました。

本章の冒頭で説明したように、「*Integrating WebSphere Data Interchange V3.2 with WebSphere Business Integration Connect V4.2*」チュートリアルに従ってサンプル構成を実際に作成できます。チュートリアルには、環境の構成に役立つサンプル・スクリプトとマップが記載され、またサンプル交換のテスト方法も説明されています。このチュートリアルを使用するには、次の URL にアクセスしてください。

[www.ibm.com/developerworks/websphere/](http://www.ibm.com/developerworks/websphere/)

このサイトで、チュートリアルのタイトルを探してください。

パートナーの B2B 機能を設定する場合は、文書フローに「ISA (すべて)」(「すべて (すべて)」ではない) を選択する必要があります。セットアップするインタラクションおよびアクティブにする接続でも、文書フローは「ISA (すべて)」になります。

---

## 第 6 章 その他の製品との統合

---

### WebSphere Transformation Extender との統合

この章では、WebSphere Partner Gateway を WebSphere Transformation Extender と統合する方法を説明します。

#### 概要

#### WebSphere Transformation Extender とは

WebSphere Transformation Extender は、強力なトランザクション指向のデータ統合ソリューションで、ハンド・コーディングを行わなくても高容量の複雑なトランザクションを自動化できます。これにより、企業の投資収益率が向上します。この製品は、EDI、XML、SWIFT、HIPAA および他の標準ベースの B2B 統合や、複数のアプリケーション、データベース、メッセージング・ミドルウェアのデータのリアルタイム統合、全社的な通信テクノロジーをサポートします。

WebSphere Transformation Extender は、バッチおよびリアルタイム環境で、ソース・システムのデータを変換して、ターゲット・システムに送付します。ソースには、ファイル、リレーショナル・データベース、MOM (メッセージ指向ミドルウェア)、パッケージ化アプリケーション、または他の外部ソースが含まれます。ソースからデータを取り出した後で、WebSphere Transformation Extender 製品は、そのデータを変換し、データを必要とする任意の数のターゲットに各ターゲット・システムに適した内容および形式でそのデータを送付します。WebSphere Transformation Extender 製品で提供される機能は、以下のとおりです。

- メインフレーム・アプリケーション、レガシー・アプリケーション、エンタープライズ・アプリケーション、データベース、メッセージング・システム、および外部情報源との幅広い接続性。
- 120 を超える事前構築関数の包括的なライブラリー。これにより、開発時間が短縮され、検証、変換、およびルーティングのための規則の指定が単純化されます。
- 適切なタイミングで適切な形式での変換 (バッチ、リアルタイム、組み込みを問わない) をサポートするための複数の実行オプション。
- 開発、デプロイメント、保守および高可用性プラットフォーム・サポートのためのエンタープライズ・クラス機能。これにより、継続的な管理および実装リスクが削減され、ハンド・コーディングの場合よりも結果を速く配信できます。

#### WebSphere Transformation Extender と WebSphere Partner Gateway の連携方法

WebSphere Transformation Extender は、文書の変換を実行するために WebSphere Partner Gateway 内で使用されます。WebSphere Transformation Extender との統合は、2 つの方法で実現できます。

まず、WebSphere Partner Gateway と WebSphere Transformation Extender が直接対話しない方法です。この方法では、WebSphere Transformation Extender が別のバツ

クエンド・アプリケーションとして扱われます。WebSphere Message Broker を使用することにより、WebSphere Transformation Extender が取り込まれます。この方法の場合、文書は WebSphere Partner Gateway 内では変換されず、WebSphere Message Broker にパススルーされ、そこで WebSphere Transformation Extender が使用されて変換が実行されます。

次に、WebSphere Partner Gateway と WebSphere Transformation Extender が直接対話する方法です。ユーザー出口をアクション・ステップ・ハンドラーとして使用し、WebSphere Partner Gateway 内で WebSphere Transformation Extender に対する呼び出しが行われます。変換を実行するために WebSphere Transformation Extender Java RMI API を使用して、WebSphere Transformation Extender を起動するユーザー出口がインプリメントされています。

## WebSphere Transformation Extender との統合の計画

前述のとおり、WebSphere Transformation Extender との統合は、2 つの方法で実現できます。WebSphere Message Broker を使用する方法と、ユーザー出口を使用する方法です。

### WebSphere Message Broker を使用した統合

**WebSphere Message Broker を使用するための要件:** WebSphere Message Broker を WebSphere Partner Gateway で使用する場合は、以下が必要です。

- WebSphere Partner Gateway によってサポートされる WebSphere Message Broker のバージョン
- WebSphere Message Broker によってサポートされる WebSphere Transformation Extender のバージョン
- WebSphere Transformation Extender for WebSphere Message Broker

**WebSphere Message Broker を使用するための計画:** WebSphere Partner Gateway は、WebSphere Message Broker を介して、WebSphere Transformation Extender と間接的に対話します。つまり、WebSphere Partner Gateway のトポロジーは WebSphere Message Broker に関連し、WebSphere Transformation Extender のトポロジーは WebSphere Message Broker に関連しています。WebSphere Transformation Extender マップの作成、構成、およびデプロイメントは、WebSphere Message Broker のコンテキストで行われます。

### WebSphere Partner Gateway ユーザー出口を使用した統合

**アクションから WebSphere Partner Gateway ユーザー出口を使用するための要件:** WebSphere Partner Gateway のバージョン 6.2 は、WebSphere Transformation Extender 8.1 (フィックスパック 1 以降) との統合をサポートします。フィックスパック 1 を使用している場合で、ローカル呼び出しを使用する場合は、WebSphere Transformation Extender サポートのパッチ番号 105385 が必要です。WebSphere Transformation Extender のこれ以降のフィックスパックには、このパッチが組み込まれている可能性があります。RMI Server を使用しない場合は、ローカル呼び出し (インプロセスとも呼ばれる) が、WebSphere Transformation Extender Java API または WebSphere Transformation Extender RMI API で使用されます。

**WebSphere Partner Gateway ユーザー出口を使用するための計画:** WebSphere Transformation Extender は、Java RMI API を使用して、WebSphere Partner Gateway 内からローカルにまたはリモート側で呼び出すことができます。使用方法は、デプロイメント環境によって異なります。リモート呼び出しでは、WebSphere Partner Gateway により大きな柔軟性と安全性が確保されます。

**ローカル呼び出し (インプロセス):** 以下は、WebSphere Transformation Extender をローカルに呼び出す場合の情報です。

- ローカル呼び出しを行うには、WebSphere Transformation Extender Java API を使用するか、WebSphere Transformation Extender RMI API をローカル・モード (ホスト名の値がヌル、ポート番号の値が 0) で使用します。
- WebSphere Transformation Extender は、WebSphere Partner Gateway と同じマシンで稼働している必要があります。つまり、WebSphere Partner Gateway の各インスタンスに、WebSphere Transformation Extender がインストールされている必要があります。
- 文書の変換は、WebSphere Transformation Extender API の呼び出し元と同じ WebSphere Partner Gateway のスレッドおよびメモリーで行われます。
- WebSphere Partner Gateway と WebSphere Transformation Extender は、両方の製品でサポートされるオペレーティング・システムにインストールする必要があります。
- WebSphere Partner Gateway の PATH 環境内にランタイム WebSphere Transformation Extender ライブラリーを置く必要があります。
- WebSphere Transformation Extender マップが、WebSphere Transformation Extender API にとって使用可能になっている必要があります。
- WebSphere Partner Gateway は、変換中の予期しないエラーによる影響を受ける場合があります。

**リモート呼び出し:** 以下は、WebSphere Transformation Extender をリモート側で呼び出す場合の情報です。

- WebSphere Partner Gateway は、WebSphere Transformation Extender に対してリモートのマシンに置くことができます。つまり、WebSphere Partner Gateway の複数のインスタンスが、同じ WebSphere Transformation Extender インスタンスを共用できます。
- WebSphere Partner Gateway は、Windows オペレーティング・システム上で稼働している必要はありません。
- WebSphere Transformation Extender RMI Server を実行しておく必要があります。
- 文書の変換は WebSphere Transformation Extender RMI Server で行われます。
- 入力文書および出力文書用に WebSphere Partner Gateway 共通ファイル・システムが、WebSphere Transformation Extender RMI Server で使用可能になっている必要があります。
- WebSphere Transformation Extender マップが、WebSphere Transformation Extender RMI Server で使用可能になっている必要があります。
- WebSphere Partner Gateway は、変換中の予期しないエラーによる悪影響を受けません。

## WebSphere Message Broker と WebSphere Transformation Extender の統合

このデプロイメントで期待される使い方では、変換が WebSphere Message Broker を使用して WebSphere Transformation Extender によって行われますが、既存の機能を使用して WebSphere Partner Gateway で変換を実行することもできます。WebSphere Message Broker の詳細については、187 ページの『WebSphere Message Broker との統合』を参照してください。WebSphere Message Broker との統合については、WebSphere Transformation Extender の資料を参照してください。

## WebSphere Transformation Extender ユーザー出口の統合

WebSphere Partner Gateway との統合を行う前に、この製品の要件および説明に従って、WebSphere Transformation Extender をインストールする必要があります。さらに、変換する入力文書を読み取り、変換から出力文書を書き出すには、WebSphere Transformation Extender が稼働しているマシンから、WebSphere Partner Gateway 共通ファイル・システムにアクセス可能でなければなりません。

構成の要件については、WebSphere Transformation Extender の製品資料を参照してください。WebSphere Transformation Extender RMI Server の構成に関する考慮事項については、235 ページの『WebSphere Transformation Extender - RMI Server 構成』を参照してください。

## WebSphere Transformation Extender 用の WebSphere Partner Gateway ユーザー出口の作成

WebSphere Transformation Extender ユーザー出口は、WebSphere Transformation Extender RMI API を使用する必要があります。これらの API により、WebSphere Transformation Extender のインストール先を柔軟に選択でき、WebSphere Partner Gateway の複数のインスタンスが同じ WebSphere Transformation Extender インストール環境を使用することが可能になります。また、ユーザー出口は、WebSphere Transformation Extender RMI Server が listen するホスト名およびポートに関して構成可能でなければなりません。

このユーザー出口を作成するために必要な特定のタスクについては、以下を参照してください。

- 可変ワークフロー・ハンドラーのインプリメントについては、「*WebSphere Partner Gateway Programmer Guide*」を参照してください。
- WebSphere Transformation Extender API の使用方法については、WebSphere Transformation Extender の資料を参照してください。
- このユーザー出口については、WebSphere Partner Gateway のサンプル・ユーザー出口を参照してください。

## ユーザー出口のための WebSphere Partner Gateway の構成

WebSphere Partner Gateway Console を使用して、WebSphere Transformation Extender ユーザー出口を使用するよう WebSphere Partner Gateway を構成する必要があります。まず、ユーザー出口を WebSphere Partner Gateway に対して定義し、このユーザー出口を使用するアクションを作成する必要があります。特に、以下のタスクを実行する必要があります。

1. ユーザー出口用に作成した XML ハンドラー定義をアップロードします。

2. ユーザー出口を使用するアクションを作成します。

WebSphere Transformation Extender ユーザー出口をアップロードしてアクションを作成したら、他の WebSphere Partner Gateway 文書フローの場合と同じように構成します。特に、以下のタスクを実行する必要があります。

1. 文書タイプを構成します。
2. 文書タイプの B2B 機能をセットアップします。
3. ユーザー出口が使用する WebSphere Partner Gateway マップをインポートします。

注: マップの形式は、ユーザー出口に定義した形式です。

4. インタラクションを構成します。
5. 接続を管理します。

これらのタスクを実行するための具体的な情報については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

## ユーザー出口のための WebSphere Partner Gateway ランタイム構成

232 ページの『ユーザー出口のための WebSphere Partner Gateway の構成』では、WebSphere Transformation Extender ユーザー出口を使用するよう WebSphere Partner Gateway を構成しました。WebSphere Partner Gateway で文書を処理できるようにするには、事前に Java クラス・ファイルをインストールして、ランタイム・コードを使用可能にする必要があります。

Java クラス・ファイルをインストールするには、以下を実行してください。

1. ユーザー出口 Java クラス・ファイルを {WPG Install}\%router%\lib%\userexits に置きます。
2. WebSphere Transformation Extender インストール・ディレクトリーから、開発キット jar ファイル dtxpi.jar をディレクトリー {WPG Install}\%router%\lib%\userexits にコピーします。この jar ファイルには、WebSphere Transformation Extender Map Transformation を起動するために必要な WebSphere Transformation Extender ランタイム・クラスが含まれています。
3. 新しい JAR ファイルを使用するには、WebSphere Partner Gateway を再始動する必要があります。
4. ローカルで実行している場合は、WebSphere Transformation Extender インストール・ディレクトリーをシステム PATH に追加する必要があります。新しいパス設定を使用するには、WebSphere Partner Gateway を再始動する必要があります。

## WebSphere Partner Gateway が提供するサンプルのユーザー出口

WebSphere Partner Gateway と WebSphere Transformation Extender の統合に役立つよう、サンプルのユーザー出口が提供されています。

**WebSphere Transformation Extender のサンプル・ユーザー出口の概要:** サンプル・ユーザー出口は、WebSphere Partner Gateway アクション内から WebSphere Transformation Extender を呼び出して変換を実行するための推奨事項を示しています。ユーザー出口の実際のインプリメンテーションは、入力カードおよび出力カー

ドを使用して WebSphere Transformation Extender マップを構成する方法によって異なります。これは、WebSphere Partner Gateway から実行できる実用的なサンプルです。サンプル、ソース・コード、入力ファイル、およびマップのインストール方法および使用方法に関する README は、製品 CD の Integration/WebSphereTransformationExtender ディレクトリーにあります。詳細な情報については、これを参照してください。

このサンプルでは、2 つのタイプのマップがあります。WebSphere Transformation Extender Map Designer を使用して作成される WebSphere Transformation Extender マップと、ユーザー出口が WebSphere Transformation Extender マップを実行するのに役立つメタデータがプロパティー・ファイルの形式で組み込まれている WebSphere Partner Gateway マップです。WebSphere Partner Gateway マップの形式は、ユーザー出口の実装に固有です。内容の詳細については、WebSphere Partner Gateway サンプル・マップを参照してください。

**サンプル・ユーザー出口クラスの説明:** これらのクラスは、アクション内の操作に必要な WebSphere Partner Gateway ユーザー出口インターフェース・コードで WebSphere Transformation Extender API をラッピングします。インプリメンテーションの詳細については、Java ソース・コードを参照してください。以下で、サンプル・ユーザー出口のクラスについて説明します。

#### **WTXTransformationFactory**

このクラスは、アクション・ユーザー出口ハンドラーです。ハンドラー属性に基づいて、使用する BusinessProcessInterface インスタンスを作成して返します。ハンドラー属性は、さらに構成を行うためにビジネス・プロセス・インスタンスで使用されます。

#### **WTXProcess**

このクラスは、WebSphere Transformation Extender とのインターフェースを提供する Business Process クラスのスーパー・クラスです。このクラスには、WebSphere Transformation Extender の非 API アスペクトの全体的なフロー・ロジックが含まれています。WebSphere Transformation Extender API 呼び出し特性は、callWTX() メソッドのサブクラスにカプセル化されています。setCfg() メソッドは、サブクラスによって使用され、そのサブクラスに関連するハンドラー属性の情報が設定されます。

#### **WTXProcessRMI**

これは WTXProcess のサブクラスです。この BusinessProcess クラスは、WebSphere Transformation Extender RMI Server を使用してリモート側で、WebSphere Transformation Extender 上の WebSphere Transformation Extender RMI API を使用してマップを呼び出します。または、WebSphere Transformation Extender RMI Server を使用せずにローカルでマップを処理する場合。特性は、以下のとおりです。

- WebSphere Transformation Extender マップへの入力として、ファイル入力が期待されます。WebSphere Partner Gateway マップからの情報を使用して、マップ名および他のランタイム呼び出し要件が決定されます。
- ユーザー出口ハンドラー構成の RMI Sever 値を使用して、WebSphere Transformation Extender RMI Server を使用するかどうかが決まります。WebSphere Transformation Extender RMI Server を使用する場合は、WebSphere Transformation Extender RMI Server へのアクセスで、ユーザ

ー出口ハンドラー構成のホスト名およびポート番号が使用されます。

WebSphere Transformation Extender RMI API は、マップをローカルに呼び出すために使用できるため、ユーザー出口ハンドラー構成がマップをローカルに呼び出すよう指示する場合は、同じコードを使用できます。マップをローカルに呼び出すためには、WebSphere Transformation Extender ランタイム dll がパス内で使用可能になっている必要があります。

- WebSphere Transformation Extender は Windows 上で稼働するため、WebSphere Partner Gateway が UNIX マシンに存在する場合は、リモート呼び出しを行う必要があります。

### WTXProcessLocal

これは WTXProcess のサブクラスです。BusinessProcess クラスは、WebSphere Transformation Extender 上のマップをローカルに呼び出します。特性は、以下のとおりです。WebSphere Transformation Extender マップへの入力として、ファイル入力が期待されます。WebSphere Partner Gateway マップからの情報を使用して、マップ名および他のランタイム呼び出し要件が決定されます。

この BusinessProcess クラスを使用するには、WebSphere Transformation Extender ランタイム dll がパス内で使用可能になっている必要があります。

注: WTXTransformationFactory クラスは、このサブクラスをインスタンス化しません。WTXProcessRMI 内の WebSphere Transformation Extender RMI API もローカル呼び出しに使用できるためです。このクラスは、WebSphere Transformation Extender Java API を使用するためのサンプルとして提供されています。

**サンプル・ユーザー出口の基本ランタイム操作:** 以下は、サンプル・ユーザー出口の基本ランタイム操作に関する情報です。

- WebSphere Partner Gateway が、外部パートナーから XML 文書を受信します。
- WebSphere Partner Gateway 接続が XML 形式および XML 文書から取得されたビジネス ID に基づいて検索され、その接続を使用して文書が処理されます。
- その接続のカスタム・アクションが、サンプル WebSphere Transformation Extender Transformation ユーザー出口を使用するよう構成されます。WebSphere Transformation Extender Transformation ユーザー出口が呼び出されると、以下の操作が実行されます。
  - WebSphere Transformation Extender RMI Server が稼働しているホスト名とポート番号が取得されます。
  - マップ入力および出力カード情報が WebSphere Partner Gateway Transformation Extender マップに基づいて設定されます。
  - WebSphere Transformation Extender マップが呼び出され、結果がビジネス文書オブジェクトに設定されます。

### WebSphere Transformation Extender - RMI Server 構成

このセクションでは、WebSphere Partner Gateway に関連する WebSphere Transformation Extender RMI Server の構成についての考慮事項を取り上げます。構成プロパティの詳細については、WebSphere Transformation Extender の資料を参照してください。

WebSphere Transformation Extender マップをローカルに呼び出す場合、RMI Server 構成は不要です。ローカル呼び出し (インプロセス) により、WebSphere Partner Gateway で実行される同じスレッド内でマップが呼び出されます。マップをリモート側で呼び出す場合は RMI Server を使用するため、このセクションの情報が適用されます。インプロセスの考慮事項 (ローカル呼び出し) については、プロパティ `server.mode.multi.process` で詳しく説明します。

RMI Server は、コマンド行から始動します。コマンド行パラメーター「`-verbose`」を指定することをお勧めします。このパラメーターにより、ランタイム構成特性の一部が標準出力にプリントされるためです。

RMI Server が始動すると、セットアップ構成パラメーターがプロパティ・ファイルから読み取られます。WebSphere Partner Gateway に関連してこれらのパラメーターを構成するための考慮事項について、以下で取り上げます。

**server.mode.multi.process: 推奨設定: true**

**説明:** このプロパティは、RMI Server をマルチプロセス構成オプションで実行します。RMI Server に対する各呼び出しは別々のプロセスで実行されるため、マップ例外が発生する場合に最大限の保護が提供されます。マップ例外が発生しても、RMI Server は継続して実行されます。単一プロセス構成 (`server.mode.multi.process=false`) を使用する場合、RMI Server に対する各呼び出しは同じプロセスで実行されますが、スレッドは異なります。ただし、マップ例外が発生する場合、RMI Server は実行を停止する可能性があります。

#### マルチプロセスの考慮事項:

- 各マップ呼び出しは、独自のプロセスで実行されます。
- 各プロセスには、独自の Java メモリー・ヒープがあります。
- 強化されたセキュリティー。あるマップで例外が発生しても RMI Server は停止しません。
- 各プロセスで WebSphere Transformation Extender インスタンスがメモリーにロードされます (最大で約 30 MB)。
- プロセスが使用できない場合、マップ呼び出しがタイムアウトになるまでの待ち時間が RMI Server プロパティ `pool.acquire.process.timeout` によって決定されます。
- 使用可能な十分の数のプロセスを確保するには、少なくとも、稼働中の WebSphere Partner Gateway 文書マネージャー数と文書マネージャー内のスレッド数を乗算した数のプロセスを構成する必要があります。文書マネージャーは、複数のコンポーネントで構成されます。ユーザー出口を実行するコンポーネントは、ビジネス・プロセス・エンジン (BPE) です。これは、デフォルトで 2 つのスレッドに構成されています。したがって、デフォルトの構成で 2 つの文書マネージャー・インスタンスがインストールされている場合は、4 つのスレッド (2 つのインスタンス X 2 つのスレッド) が実行されるため、RMI Server は最低 4 つのプロセスに対応するよう構成する必要があります。プロセスの数を制御する方法については、RMI Server 構成プロパティ `pool.max.process.count` を参照してください。

#### シングル・プロセスの考慮事項:

- 各マップ呼び出しは、独自のスレッドで実行されますが、他のマップ・スレッドと同じプロセスで実行されます。
- マップ例外が発生する場合、RMI Server は停止する可能性があります。
- ロードされる WebSphere Transformation Extender インスタンスは 1 つだけであるため、メモリー・オーバーヘッドが減少します。

#### インプロセスの考慮事項:

- これは、RMI Server なしでローカルに実行されます。サーバー環境の WebSphere Transformation Extender では推奨されません。マップが呼び出し側スレッド (この場合は WebSphere Partner Gateway) で実行されるためです。異常な例外により、WebSphere Partner Gateway 内で問題が発生する可能性があります。
- WebSphere Transformation Extender は、WebSphere Partner Gateway と同じマシンにインストールする必要があります。
- Windows オペレーティング・システムに限定されます。

#### **pool.mode.managed:** 推奨設定: true

**説明:** このプロパティは、一度に使用できるプロセスの数を管理します。false の場合、制限はありません。プロセスが際限なく増え続ける状態を避けるために推奨設定を使用してください。

#### **pool.max.process.count:** 推奨設定: (WebSphere Partner Gateway インスタンスの合計数) X (そのスレッド構成)

**説明:** WebSphere Partner Gateway 文書マネージャーは、複数のコンポーネントで構成されます。ユーザー出口を実行するコンポーネントは、ビジネス・プロセス・エンジン (BPE) です。これは、デフォルトで 2 つのスレッドに構成されています。したがって、デフォルトの構成で 2 つの文書マネージャー・インスタンスがインストールされている場合は、4 つのスレッド (2 つのインスタンス X 2 つのスレッド) が実行されるため、RMI Server は最低 4 つのプロセスに対応するよう構成する必要があります。このプロパティにより、RMI Server に対するユーザー出口呼び出しを処理するために十分な数のプロセスが確保されます。プロセス数が WebSphere Partner Gateway スレッド数より少ない場合、RMI Server に対するユーザー出口呼び出しは、他の WebSphere Partner Gateway スレッドがプロセスを使用している場合に、プロセスが使用可能になるまで待機する必要があります。

**pool.max.keep.idle.count:** 推奨設定: 特定の設定はありませんが、プロセスを絶えず作成しなくても済むよう (パフォーマンスに影響する)、十分高い値に設定してください。設定の単位は秒です。

**説明:** このプロパティは、プロセスが再利用されるまでの、プロセスのアイドル時間を制御します。pool.max.process.count に達する前にプロセスが再度必要となる場合は、プロセスが再度作成されます。

#### **pool.acquire.process.timeout:** 推奨設定: 10 (秒)

**説明:** このプロパティは、マップに対する呼び出しが、マップを実行できるようプロセスを取得するまで待機する時間を決定します。pool.max.process.count が正しく構成されていれば、タイムアウトが問題になることはありません。

**pool.map.auto.unload.timeout:** 推奨設定: 120 (秒)

**説明:** RMI Server がマップをメモリー内のキャッシュに入れます。このプロパティは、マップをプールから削除する前に、RMI Server メモリー・プールにマップを保持する時間を制御します。マップがプールから削除された後でマップが再度必要となる場合、マップはファイル・システムから読み取られます。WebSphere Partner Gateway マップを実動状態にしておくために、長い時間キャッシュに入れておくことが有利です。最適の設定は、経験を通して決定する必要があります。

---

## WebSphere Transformation Extender Trading Manager との統合

この章では、WebSphere Partner Gateway を WebSphere Transformation Extender Trading Manager と統合する方法を説明します。

### 概要

#### WebSphere Transformation Extender Trading Manager とは

WebSphere Transformation Extender Trading Manager は、電子商取引データを管理および処理するクライアント/サーバー製品で、パートナー関係とメッセージ・フローの Business-to-Business (B2B) 統合を管理および制御するための機能を備えています。Trading Manager のユーザーは、拡張エンタープライズに渡る B2B 統合環境全体を監査、制御、モニター、および表示でき、バックエンド・システムと完全に統合された安全なデータ交換を実現できます。

Trading Manager とそのコンポーネントについては詳しくは、Trading Manager の資料を参照してください。

#### WebSphere Transformation Extender Trading Manager と WebSphere Partner Gateway の連携方法

Trading Manager との統合は、他のバックエンド・アプリケーションの場合と同様です。Trading Manager の標準的な使い方は、EDI 文書の処理です。例えば、EDI ペイロードを持つ AS2 文書が外部パートナーから受信されると、EDI ペイロードが Trading Manager に送信され、さらに処理されます。WebSphere Partner Gateway (宛先) と Trading Manager (ポスト・オフィス) の間で使用できる互換インターフェースは、File、HTTP、または JMS です。

統合シナリオは、以下のとおりです。

- 外部パートナー → WebSphere Partner Gateway → Trading Manager (内部パートナー) → バックエンド・アプリケーション外部パートナー
- WebSphere Partner Gateway → Trading Manager (内部パートナー) → バックエンド・アプリケーション

#### 該当資料

WebSphere Partner Gateway の場合、該当する資料は「*WebSphere Partner Gateway* ハブ構成ガイド」です。Trading Manager の場合は、Trading Manager の資料です。

## WebSphere Transformation Extender Trading Manager との統合の計画

WebSphere Partner Gateway バージョン 6.2 では、WebSphere Transformation Extender Trading Manager バージョン 7.8 以降との統合がサポートされます。Trading Manager との間で File インターフェースを使用する場合、WebSphere Partner Gateway と Trading Manager は両方とも、文書の交換に使用する同じファイル・システム・ディレクトリーにアクセス可能でなければなりません。

## WebSphere Partner Gateway と WebSphere Transformation Extender Trading Manager 間の共通構成

WebSphere Partner Gateway と Trading Manager の構成は、一部共通です。以下の表に、2 つの製品の間の相互参照を示します。

表 105. WebSphere Transformation Extender Trading Manager の構成に関する用語

WebSphere Partner Gateway	Trading Manager	備考
内部パートナー	内部取引パートナー	
外部パートナー	外部取引パートナー	
パートナー・ビジネス ID。EDI 交換修飾子と ID で構成される EDI の場合。	パートナー・ビジネス ID。交換修飾子と ID が別々に構成されている EDI の場合。	
宛先	ゲット・ポスト・オフィス	
レシーバー	プット・ポスト・オフィス	
接続 - 外部パートナー・ソースから内部パートナー・ターゲット	インバウンド取引リンク	EDI グループ構成に基づいて、複数のインバウンド取引リンクが定義されている可能性があるため、正確に 1 対 1 の関係はありません。
接続 - 内部パートナー・ソースから外部パートナー・ターゲット	アウトバウンド取引リンク	EDI グループ構成に基づいて、複数のアウトバウンド取引リンクが定義されている可能性があるため、正確に 1 対 1 の関係はありません。

## WebSphere Partner Gateway の構成

このセクションでは、WebSphere Transformation Extender Trading Manager インターフェースに関する項目について説明します。

文書フローは、どの統合でも通常どおりに定義されます。パートナーは、WebSphere Partner Gateway と Trading Manager の両方で定義する必要がありますが、WebSphere Partner Gateway の構成では、文書のパススルー操作に必要な事柄にのみ制限することができます。

内部パートナー、外部パートナー、文書定義、インタラクション、接続、およびレシーバーに関連した Trading Manager インターフェースに固有の論点について取り上げます。

パートナー、宛先、および文書フローの追加情報については、「*WebSphere Partner Gateway ハブ構成ガイド*」を参照してください。

## 内部パートナーについて

内部パートナーは、Trading Manager に使用することが期待されており、Trading Manager 内部取引パートナーに相当します。

表 106. *WebSphere Transformation Extender Trading Manager* の内部パートナーの構成について

項目	特別な考慮事項
ビジネス ID	EDI 交換の内部パートナーを表すビジネス ID を構成する必要があります。文書が Trading Manager に送信される場合、これは交換受信側修飾子および ID になります。文書が Trading Manager から送信される場合、これは交換送信側修飾子および ID になります。詳しくは、243 ページの『EDI 文書のルーティング』を参照してください。
宛先	<p>トランスポート・タイプおよび構成は、使用されている Trading Manager ゲット・ポスト・オフィスに相当します (File、HTTP、または JMS など)。</p> <ul style="list-style-type: none"> <li>File を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、共用するディレクトリーにアクセス可能でなければなりません。</li> <li>JMS を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、同じ JMS キューを使用するよう構成する必要があります。</li> </ul>
B2B 機能	<p>内部パートナーの B2B 機能は、Trading Manager で使用されるソース文書タイプおよびターゲット文書タイプで有効にする必要があります。EDI 文書の場合、以下のとおりです。</p> <ul style="list-style-type: none"> <li>パッケージ化 - なし。</li> <li>プロトコル - EDI-EDIFACT または EDI-X12。</li> <li>文書タイプ - UNB (EDIFACT) または ISA (X12)。</li> </ul>

## 外部パートナーについて

外部パートナーは、Trading Manager 外部取引パートナーに相当します。

表 107. *WebSphere Transformation Extender Trading Manager* の外部パートナーの構成について

項目	特別な考慮事項
ビジネス ID	EDI 交換の外部パートナーを表すビジネス ID を構成する必要があります。文書が Trading Manager に送信される場合、これは交換送信側修飾子および ID になります。文書が Trading Manager から送信される場合、これは交換受信側修飾子および ID になります。詳しくは、243 ページの『EDI 文書のルーティング』を参照してください。
宛先	外部パートナーの宛先は、Trading Manager 用に特別な方法で構成する必要はありません。

表 107. WebSphere Transformation Extender Trading Manager の外部パートナーの構成について (続き)

項目	特別な考慮事項
B2B 機能	<p>外部パートナーの B2B 機能は、外部パートナーで使用されるソース文書タイプおよびターゲット文書タイプで有効にする必要があります、Trading Manager に固有ではありません。例えば、AS2 パッケージを使用する EDI 文書の場合、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• パッケージ化 - AS</li> <li>• プロトコル - EDI-EDIFACT または EDI-X12。</li> <li>• 文書タイプ - UNB (EDIFACT) または ISA (X12)。</li> </ul>

## 文書定義について

Trading Manager との間で送受信を行う場合、文書定義にパッケージ化なしを使用してください。文書が EDI の場合、使用する文書定義は、以下のとおりです。

- パッケージ化 - なし
- プロトコル - EDI-EDIFACT または EDI-X12
- 文書タイプ - UNB (EDIFACT) または ISA (X12)

## インタラクションについて

Trading Manager への送信時のターゲット文書定義は、文書定義のセクションで説明されているものになります。Trading Manager からの受信時のソース文書定義は、文書定義のセクションで説明されているものになります。使用するアクションは、パススルーです。

## 接続について

接続のための Trading Manager 構成設定は特にありません。

## レシーバーについて

レシーバーについて、トランスポート・タイプおよび構成は、使用されている Trading Manager プット・ポスト・オフィスに相当します (File、HTTP、または JMS など)。

- File を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、共用するディレクトリーにアクセス可能でなければなりません。
- JMS を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、同じ JMS キューを使用するよう構成する必要があります。

## WebSphere Transformation Extender Trading Manager の構成

Trading Manager は、通常どおりに構成されます。パートナーが WebSphere Partner Gateway で定義されているとしても、Trading Manager でも定義する必要があります。

す。パートナーの名前は、WebSphere Partner Gateway と Trading Manager で同一である必要はありませんが、類似の名前にしておけば 2 つの製品を管理する際に役立ちます。

WebSphere Partner Gateway インターフェースに固有の項目は、以下のとおりです。

表 108. WebSphere Transformation Extender Trading Manager の構成上の考慮事項

項目	特別な考慮事項
内部取引パートナー	<p>内部取引パートナーは、WebSphere Partner Gateway の内部パートナーに相当します。</p> <p>EDI 交換の内部取引パートナーを表すビジネス ID を構成する必要があります。文書が WebSphere Partner Gateway に送信される場合、これは交換ソース修飾子および ID になります。文書が WebSphere Partner Gateway から送信される場合、これは交換受信側修飾子および ID になります。</p>
外部取引パートナー	<p>外部取引パートナーは、WebSphere Partner Gateway の外部パートナーに相当します。</p> <p>EDI 交換の外部取引パートナーを表すビジネス ID を構成する必要があります。文書が WebSphere Partner Gateway に送信される場合、これは交換受信側修飾子および ID になります。文書が WebSphere Partner Gateway から送信される場合、これは交換送信側修飾子および ID になります。</p>
ゲット・ポスト・オフィス	<p>トランスポート・タイプおよび構成は、Trading Manager とのインターフェースに使用されている WebSphere Partner Gateway の内部パートナー宛先に相当します (File、HTTP、または JMS など)。</p> <ul style="list-style-type: none"> <li>File を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、共用するディレクトリーにアクセス可能でなければなりません。</li> <li>JMS を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、同じ JMS キューを使用するよう構成する必要があります。</li> </ul>
プット・ポスト・オフィス	<p>トランスポート・タイプおよび構成は、使用されている WebSphere Partner Gateway レシーバーに相当します (File、HTTP、または JMS など)。</p> <ul style="list-style-type: none"> <li>File を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、共用するディレクトリーにアクセス可能でなければなりません。</li> <li>JMS を使用している場合、WebSphere Partner Gateway と Trading Manager は両方とも、同じ JMS キューを使用するよう構成する必要があります。</li> </ul>
インバウンド取引リンク	特定の構成はありません。
アウトバウンド取引リンク	特定の構成はありません。

構成の詳細については、WebSphere Transformation Extender Trading Manager の資料を参照してください。

## 第 7 章 その他のビジネス・プロトコル固有の情報

以下の章では、ルーティング EDI 文書などのビジネス・プロトコル固有の情報について説明し、AS パッケージ化、RosettaNet パッケージ化、および ebMS パッケージ化の場合に続いて行われる追加処理のガイドラインについても説明します。

### EDI 文書のルーティング

このセクションでは、WebSphere Partner Gateway が送受信する電子データ交換 (EDI) 文書のルーティング情報を決定するプロセスについて説明します。この処理の一般的なフロー (『EDI ルーティングの概要』を参照) について説明します。

EDI 文書のルーティング時にファイル・ベースの統合をどのように使用できるかの詳細については、48 ページの『ファイル・システム・プロトコル』を参照してください。AS パッケージ化に関する特別な考慮事項については、245 ページの『AS パッケージ化に関する特別な考慮事項』を参照してください。

### EDI ルーティングの概要

EDI 文書には、その文書の送信側と受信側の情報が格納されています。WebSphere Partner Gateway は、EDI 文書を送付する場合、この情報を使用します。一般的なフローは、次のとおりです。

1. WebSphere Partner Gateway は、文書の最初の 3 文字を調べて、使用されているプロトコルを判別します。表 109 には、各コードに関連した文書タイプ・プロトコルを示します。

表 109. EDI コードと関連の文書タイプおよびプロトコル

コード	文書タイプ	文書タイプ・プロトコル	コンテンツ・タイプとしてのアウトバウンド
ISA	X12	EDI-X12	application/EDI-X12
GS	X12	EDI-X12	application/EDI-X12
UNB	Edifact	EDI-EDIFACT	application/EDIFACT
UNA	Edifact	EDI-EDIFACT	application/EDIFACT
ICS	ICS	EDI-X12	application/EDI-X12
STX	UNTDI	EDI-Consent	application/edi-consent
BG	UCS	EDI-Consent	application/edi-consent

2. WebSphere Partner Gateway は、244 ページの表 110 に説明されているように、EDI 文書の特定の文書タイプに対応するエレメントおよび位置に基づいて、EDI 文書から送信側の情報を抽出します。

表 110. EDI コードおよび送信側情報と受信側情報の場所

コード	送信側修飾子	送信側 ID	受信側修飾子	受信側 ID	EDI コンポーネントのサポート
ISA	位置 5 にある エレメント 105	位置 6 にある エレメント 107	位置 7 にある エレメント 105	位置 8 にある エレメント 106	サポート
GS	N/A	位置 2 にある エレメント 142	N/A	位置 3 にある エレメント 124	「GS のみ」アドレッシングは非サポート
UNB UNA	UNB セグメントの位置 20 (2 番目の複合エレメント) にある複合エレメント S002 の位置 2 にあるサブエレメント 0007	UNB セグメントの位置 20 (2 番目の複合エレメント) にある複合エレメント S002 の位置 2 にあるサブエレメント 0004	UNB セグメントの位置 30 (3 番目の複合エレメント) にある複合エレメント S003 の位置 2 にあるサブエレメント 0007	UNB セグメントの位置 30 (3 番目の複合エレメント) にある複合エレメント S003 の位置 1 にあるサブエレメント 0010	サポート
ICS	位置 4 にある エレメント X05	位置 5 にある エレメント X06	位置 6 にある エレメント X05	位置 7 にある エレメント X08	ISA の先行版 - 非サポート
STX	位置 3 にある エレメント FROM1	位置 3 にある エレメント FROM2	位置 4 にある エレメント UNT1	位置 4 にある エレメント UNT2	このリリースでは非サポート
BG	N/A	位置 3 にある エレメント BG03	N/A	位置 4 にある エレメント BG04	サポート

3. WebSphere Partner Gateway は、EDI 文書の送信側 ID と修飾子を基にして送信側 ID を判別します。

一部の EDI エンベロープ (GS など) には、修飾子の概念がありません。この場合、WebSphere Partner Gateway は、ID のみを使用します。

4. WebSphere Partner Gateway は、修飾子とダッシュ (-) 文字付きの ID を連結して、WebSphere Partner Gateway プロファイル・リポジトリから送信側 ID を検索します。例えば、送信側の EDI メッセージでは、修飾子が AB で ID が 1234567 の場合、WebSphere Partner Gateway は、プロファイル・リポジトリから AB-1234567 という ID を持つ外部パートナーを検索するものと認識します。WebSphere Partner Gateway がこの ID を検出できない場合、EDI 文書は送付されません。
5. 受信側のパートナーを検索するため、WebSphere Partner Gateway は、EDI メッセージを基にして受信側の修飾子と ID を調べます。
6. WebSphere Partner Gateway は、修飾子とダッシュ (-) 文字付きの ID を連結して、プロファイル・リポジトリから受信側 ID を検索します。
7. WebSphere Partner Gateway は、目的の宛先に文書を送付します。

## AS パッケージ化に関する特別な考慮事項

このセクションでは、AS パッケージ化を指定してある場合に必要な追加の処理について説明します。

文書のパッケージ化を AS と指定した場合、WebSphere Partner Gateway は、何らかの追加処理を実行します。

バックエンド・パッケージ化および AS について詳しくは、29 ページの『バックエンド統合パッケージ化』を参照してください。

## インバウンド文書の処理方法

文書が外部パートナーから受信されると、以下のようになります。

1. WebSphere Partner Gateway は、最初に AS1 または AS2 のヘッダー情報を検査します。特に、送信側と受信側の情報を検査して、これらが有効な外部パートナーの ID と一致するかどうかを判別します。
  - AS1 の場合は、「Subject」ヘッダー・フィールドを使用します。このフィールドの形式は、「ToID;FromID」です。
  - AS2 の場合は、「AS2-From」ヘッダー・フィールドおよび「AS2-To」ヘッダー・フィールドです。

ヘッダー・フィールドの値が有効な ID と一致しない場合、WebSphere Partner Gateway は文書を送付しません。

2. 次に、WebSphere Partner Gateway は、ペイロードの通常の (プロトコルおよび文書タイプを判別し、ビジネス ID を取り出すなどの) ステップを実行します。EDI ペイロードについて詳しくは、243 ページの『EDI ルーティングの概要』を参照してください。

## アウトバウンド文書の処理方法

バックエンド・システムから文書を受信すると、WebSphere Partner Gateway は、AS BusinessID 属性がソース・パッケージ化 (なし) と受信側パッケージ化 (AS) の両方に指定されているかどうかを調べます。

- AS BusinessID 属性が指定されている場合、WebSphere Partner Gateway は、この情報を使用して、AS1 ヘッダーまたは AS2 ヘッダーに「From ID」および「To ID」を生成します。
- この属性が指定されていない場合、WebSphere Partner Gateway は、文書の送信側および受信側 ID を使用します。EDI の場合、送信側および受信側 ID 情報と修飾子が連結されます (243 ページの『EDI ルーティングの概要』を参照)。

## パートナー・プロファイルの 2 つの ID の設定

WebSphere Partner Gateway は、EDI 文書から得られた情報だけでなく、AS1 と AS2 の 2 つのヘッダー情報を使用するため、同じパートナーの ID が異なる形式で表現されることがあります。例えば、送信側の AS ヘッダー情報は 123456789 でも、EDI 文書から得られる情報は AB-12345678 になります。

外部パートナーのプロファイルに両方の ID が登録されていることを確認してください。情報については、「*WebSphere Partner Gateway* ハブ構成ガイド」を参照してください。

## AS3 MDN 処理

一般に、WebSphere Partner Gateway が MDN をパートナーに送信する場合は、元の AS3 要求文書の「Disposition-notification-to」ヘッダー属性の IP アドレス値を使用して MDN をパートナーに送信します。具体的な動作は、戻り宛先 (パートナーからの元の AS3 要求文書に使用された接続で構成されている) によって決定されません。

パートナーからの元の AS3 要求文書に使用された接続には、戻り宛先構成がありません。この戻り宛先構成の値は、以下のように機能します。

- **FTP 宛先** - FTP 宛先用に構成された値ではなく、元の AS3 要求の「Disposition-notification-to」ヘッダー属性の IP アドレス値が使用されます。
- **FTP スクリプト宛先** - 元の AS3 要求の「Disposition-notification-to」ヘッダー属性の IP アドレス値は使用されません。FTP スクリプト宛先に構成されている値が使用されます。
- **別の宛先トランスポート・タイプ** - MDN 文書に障害が発生します。

## AS に関する他のリファレンス

バックエンド統合および AS について詳しくは、29 ページの『バックエンド統合パッケージ化』を参照してください。

---

## RosettaNet パッケージ化に関する特別な考慮事項

### RosettaNet に関する他のリファレンス

バックエンド統合および RosettaNet について詳しくは、以下を参照してください。

- 18 ページの『RosettaNet』
- 29 ページの『バックエンド統合パッケージ化』

WebSphere Process Server および RosettaNet について詳しくは、63 ページの『WebSphere Process Server との統合の計画』を参照してください。

---

## ebMS パッケージ化に関する特別な考慮事項

### ebMS に関する他のリファレンス

バックエンド統合および ebMS について詳しくは、以下を参照してください。

- 20 ページの『ebMS』
- 29 ページの『バックエンド統合パッケージ化』

---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502  
神奈川県大和市下鶴間1623番14号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Burlingame Laboratory Director  
IBM Burlingame Laboratory  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。

す。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

本「プログラム」は、IBM 社およびその他の著作権により保護されています。Copyright (c) 1995-2008  
All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

---

## プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

**重要:** 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

---

## 商標

以下は、世界の多くの国で登録された International Business Machines Corp. の商標です。

IBM	DB2	IMS	MQIntegrator	Tivoli
IBM logo	DB2 Universal Database	Informix	MVS	WebSphere
AIX	IBMLink	iSeries	OS/400	z/OS
CICS	i5/OS	Lotus Domino	Passport Advantage	
CrossWorlds		Lotus Notes	SupportPac	

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

WebSphere Partner Gateway Enterprise Edition および Advanced Edition には、Eclipse Project ([www.eclipse.org](http://www.eclipse.org)) により開発されたソフトウェアが含まれています。



# 索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

## [ア行]

宛先  
説明 7, 53  
定義 53  
イベント通知 19, 22, 23, 24  
イベント通知メッセージ  
フィールド 25  
例 26  
インタラクション  
作成 55, 58  
説明 9  
バックエンド・システムへの送信用  
55  
インポート・バインディング  
説明 63  
JMS 69, 99  
SCA 93  
Web サービス 111  
エクスポート・バインディング  
説明 62  
JMS 69, 98  
SCA 90  
Web サービス 110  
エンコード属性 39  
エンドポイント URL、変更 113  
エンベロープ・フラグ 38  
オンライン/オフライン、宛先 50

## [カ行]

キュー、メッセージ配信 50  
キュー・マネージャー 47  
グループ、EDI 13  
グローバル固有 ID (GUID) 51  
交換、EDI 13  
コラボレーション  
説明 130  
Adapter for HTTP 167  
Adapter for JMS 184

## [サ行]

サブレット、WebSphere Process Server  
用、HTTP 90, 95

再試行カウント、宛先 50  
再試行間隔、宛先 50  
実施権、特許権 247  
使用許諾  
住所 247  
スキーマ、XML  
イベント通知ペイロード 24  
トランスポート・エンベロープ 38  
スレッド数、宛先 50

## [タ行]

タイム・スタンプ・フィールド 26  
タグ  
添付ファイル 38  
トランスポート・エンベロープ 38  
ペイロード 38  
知的所有権 247  
重複メッセージ処理 51  
データ・バインディング  
概要 68  
バックエンド統合パッケージ化 68  
HTTP トランスポート 69  
JMS トランスポート 69  
データ・ハンドラー  
子メタオブジェクト 140  
添付ファイル 131  
トップレベル・メタオブジェクト 144  
添付ファイル  
エンコード 39  
コンテンツ・タイプ 39  
説明 38  
タグ 38  
タグ属性 38  
データ・ハンドラー 131  
InterChange Server 147  
WebSphere Process Server 72  
添付ファイル・コンテナ・ビジネス・オブジェクト  
InterChange Server 149  
WebSphere Process Server 72  
動的メタオブジェクト  
HTTP 165  
JMS 183  
特許権 247  
トップレベル・オブジェクト  
説明 70  
属性 70  
デフォルト 76  
トップレベル・ビジネス・オブジェクト  
InterChange Server 160

トップレベル・ビジネス・オブジェクト  
(続き)

WebSphere Process Server 70  
トランザクション、EDI 13  
トランスポート・エンベロープ  
タグ 38  
XML スキーマ 38  
トランスポート・エンベロープ資料  
および添付ファイル 38  
トランスポート・プロトコル  
バックエンド統合パッケージ化 40  
リスト 40  
ebMS に必要 20  
RosettaNet に必要 18  
XMLEvent に必要 23  
トランスポート・レベルのヘッダー  
内容 29  
AS1 ソース 34  
AS2 ソース 33  
ebMS ソース 35  
InterChange Server 163, 180  
RosettaNet ソース 32  
WebSphere Process Server 73

## [ナ行]

内部パートナー 6

## [ハ行]

バイナリー文書  
サポートされるトランスポート・プロ  
トコル 41, 42  
パッケージ化の制限 44, 46  
バックエンド統合の概要 5  
バックエンド統合パッケージ化  
エンベロープ・フラグ 38  
説明 28  
データ・バインディング 68  
トランスポート・プロトコル 40  
必要時 39, 43, 46  
例 40  
HTTP トランスポート・プロトコル  
43  
JMS トランスポート・プロトコル 46  
バックエンド文書 57  
バックエンド・システム  
文書の受信 56  
文書の送信 52  
パッケージ化  
説明 8, 28

## パッケージ化 (続き)

- なし 28
- バックエンド統合 28
- AS 245
- ebMS 246
- RosettaNet 246
- パッケージ化なし
- 説明 28
- 必要時 39, 44, 46
- HTTP トランスポート・プロトコル 44
- パッケージ化ヘッダー・ビジネス・オブジェクト 73
- ハブ管理者 6
- ビジネス・オブジェクト
  - InterChange Server
    - 添付ファイル 149
    - 動的メタオブジェクト 165, 183
    - トップレベル 160
    - ペイロード 127, 147, 150
    - ユーザー定義プロパティ 164
    - HTTP 構成メタオブジェクト 165
    - JMS プロパティ 181
  - WebSphere Process Server
    - 添付ファイル・コンテナ 72
    - トップレベル 70
    - パッケージ化ヘッダー 73
    - ペイロード・コンテナ 71
  - ビジネス・オブジェクト ID フィールド 26
  - ビジネス・オブジェクト定義、InterChange Server
    - 作成 126
    - Attachment Data Handler 145
    - HTTP 158
    - JMS 179
  - ビジネス・プロトコル 11
  - ファイル・ディレクトリー・トランスポート・プロトコル
    - 説明 48
    - ディレクトリー構造 48
    - WebSphere Process Server 114
- 文書
  - AS1 34
  - AS2 33
  - cXML 12, 91, 94
  - ebMS 20, 35
  - EDI 243
  - RosettaNet 18, 32
  - SOAP 12, 110, 167, 197
  - XMLEvent 23
- 文書処理の概要 5
- 文書フロー定義
  - 説明 8
  - 定義 54

## ペイロード

- イベント通知 24
- イベント通知メッセージ 24
- エンコード 39
- コンテンツ・タイプ 39
- 説明 37
- タグ 38
- タグ属性 38
- ペイロード・コンテナ・ビジネス・オブジェクト 71
- ペイロード・データ・ハンドラー
  - Adapter for HTTP 156
  - Adapter for JMS 177
- ペイロード・ビジネス・オブジェクト
  - InterChange Server 150, 159, 179
  - WebSphere Process Server 71

## [マ行]

### メソッド

- getAsByteArray 80, 83
- getAsString 80, 83
- getDataObject 83
- getMessageType 88
- getxAuxHeader 83
- isBusinessException 88
- read 75, 83, 88
- setBOPrefix 76, 84
- setBusinessException 88
- setDataObject 84
- setDebugLevel 84
- setFromByteArray 75, 84
- setFromString 75, 85
- setOptions 80, 85
- setPackagingSchema 80, 86
- setTLOType 76, 86
- setxAuxHeader 77, 87
- write 80, 87, 89
- メッセージ・フロー
  - 説明 193
  - HTTP トランスポート用の作成 195
  - JMS トランスポート用の作成 204

## [ヤ行]

- ユーザー定義プロパティ・ビジネス・オブジェクト 164

## [ラ行]

- レシーバー
  - 説明 6
  - 定義 57

## [数字]

- 0A1 メッセージ 19

## A

### Adapter for HTTP

- インストール 156
- 構成 155
- コラボレーションのバインド 167
- ビジネス・オブジェクト構造 159
- プロトコル・ハンドラー 157
- ペイロード・データ・ハンドラー 156

### Adapter for JMS

- 構成 177
- コラボレーションのバインド 184
- 入力キューの設定 178
- ビジネス・オブジェクト構造 179
- ペイロード・データ・ハンドラー 177

### Adapter for Web Services 168

### AS パッケージ化

- 特別な考慮事項 245

### AS1 文書とトランスポート・レベル・ヘッダー 34

### AS2 文書とトランスポート・レベル・ヘッダー 33

### Attachment Data Handler

- 概要 131
- 構成 140
- 添付ファイルの表現 148
- ビジネス・オブジェクト定義 145
- リポジトリ・ファイル 140
- ロケーション 140

## B

### B2B 機能

- 設定 54, 58
- 説明 9

### BCGBackendIntegration DataBindingUtil クラス

- 概要 82
- コード例 87
- メソッド 82
- getAsByteArray メソッド 83
- getAsString メソッド 83
- getDataObject メソッド 83
- getxAuxHeader メソッド 83
- read メソッド 83
- setBOPrefix メソッド 84
- setDataObject メソッド 84
- setDebugLevel メソッド 84
- setFromByteArray メソッド 84
- setFromString メソッド 85
- setOptions メソッド 85
- setPackagingSchema メソッド 86

BCGBackendIntegration DataBindingUtil クラス (続き)

setTLOTypeName メソッド 86  
setxAuxHeader メソッド 87  
write メソッド 87

BCGBackendIntegration

JMSDataBindingImpl クラス

概要 87  
メソッド 88  
getMessageType メソッド 88  
isBusinessException メソッド 88  
read メソッド 88  
setBusinessException メソッド 88  
write メソッド 89

## C

Compute ノード 196, 198, 204, 205

content-length attribute 30

content-type attribute 30, 39

cXML 文書

サポートされるトランスポート・プロトコル 41, 42

説明 12

データ・ハンドラー 128

必要なパッケージ化 44

WebSphere Process Server 91, 94

## D

Data Interchange Services クライアント 8, 13

Documents ディレクトリー 48

## E

ebMS サービス・コンテンツ

説明 20

ebMS パッケージ化

特別な考慮事項 246

ebMS 文書

概要 20

サービス・コンテンツ 20

サポートされるトランスポート・プロトコル 41

パッケージ化 41

ebMS 文書とトランスポート・レベル・ヘッダー 35

EDI 文書

サポートされるトランスポート・プロトコル 42

パッケージ化の選択 44

ペイロードのロケーション 37

ルーティング 243

WebSphere Data Interchange 206

EventMessageID フィールド 25

## G

getAsByteArray メソッド 80, 83

getAsString メソッド 80, 83

getDataObject メソッド 83

getMethodType メソッド 88

getxAuxHeader メソッド 83

GlobalMessageID フィールド 26

## H

HTTP トランスポート・プロトコル

説明 43

ペイロードのロケーション 37

InterChange Server と 121, 152, 167

WebSphere Message Broker 189, 193

WebSphere Process Server と 90

WebSphere Process Server 用サブレット 90, 95

HTTP プロトコル構成メタオブジェクト 165

HTTP プロトコル・ハンドラー 157

HTTPInput ノード 196, 198

HTTPReply ノード 196, 198

HTTPRequest ノード 196, 198

## I

InterChange Server

概要 119

構成 126, 152, 166, 184

サポートされるトランスポート・プロトコル 121

サポートされるバージョン 120

受信に必要なコンポーネント

HTTP を通じた 153

JMS を通じた 170

成果物の作成 166, 184

送信に必要なコンポーネント

HTTP を通じた 153

JMS を通じた 170

統合 170

統合の計画 120

統合のサンプル 122

要求処理 156

HTTP トランスポート・プロトコル 152, 167

JMS トランスポート・プロトコル 170

isBusinessException メソッド 88

## J

JMS トランスポート・プロトコル

キュー・マネージャー 47

説明 45

ペイロードのロケーション 37

ヘッダー情報の作成 180

InterChange Server と 122, 170

InterChange Server ビジネス・オブジェクト構造 179

WebSphere Message Broker 189, 198

WebSphere Process Server と 98

JMS プロパティ・ビジネス・オブジェクト 181

## M

MO\_DataHandler\_Default トップレベル・メタオブジェクト 144

MQInput ノード 196, 204, 205

MQOutput ノード 196, 204, 205

## R

read メソッド 75, 83, 88

ROD/フラット文書

サポートされるトランスポート・プロトコル 41, 42

ペイロードのロケーション 37

RosettaNet サービス・コンテンツ

説明 18

ビジネス・オブジェクトの作成 74

RosettaNet パッケージ化

特別な考慮事項 246

RosettaNet 文書

概要 18

サービス・コンテンツ 18

サポートされるトランスポート・プロトコル 41

トランスポート・レベル・ヘッダーのソースとして 32

パッケージ化 41, 43

ペイロードのロケーション 37

OA1 メッセージ 19

InterChange Server 統合のサンプル 122

WebSphere Process Server 統合のサンプル 65

## S

setBOPrefix メソッド 76, 84

setBusinessException メソッド 88

setDataObject メソッド 84

setDebugLevel メソッド 84

setFromArray メソッド 75, 84  
 setFromString メソッド 75, 85  
 setOptions メソッド 80, 85  
 setPackagingSchema メソッド 80, 86  
 setTLOTypeName メソッド 76, 86  
 setxAuxHeader メソッド 77, 87  
 SOAP 文書  
 概要 12  
 サポートされるトランスポート・プロ  
 トコル 41, 42  
 必要なパッケージ化 44  
 HTTP による InterChange Server 167  
 HTTP による WebSphere Message  
 Broker 197  
 HTTP による WebSphere Process  
 Server 110  
 StatusCode フィールド 25  
 StatusMessage フィールド 25

## W

Web サービス  
 エンドポイント URL、変更 113  
 受信側パートナーの指定 12, 112  
 内部パートナーが提供 12, 110  
 認証 112, 113  
 パートナーが提供 12, 111  
 WebSphere Process Server 110  
 Web サービス上の受信側パートナー宛先  
 112  
 Web サービス用の認証 112, 113  
 Web サービス・インポート・バインディ  
 ング 111  
 Web サービス・エクスポート・バインデ  
 ング 110  
 WebSphere Data Interchange 205  
 WebSphere Message Broker  
 概要 187  
 構成 192  
 サポートされるトランスポート・プロ  
 トコル 188  
 サポートされるバージョン 188  
 受信に必要なコンポーネント  
 HTTP を通じた 193  
 JMS を通じた 198  
 送信に必要なコンポーネント  
 HTTP を通じた 193  
 JMS を通じた 198  
 統合の計画 188  
 統合のサンプル 189  
 HTTP トランスポート・プロトコル  
 193  
 JMS トランスポート・プロトコル  
 198  
 WebSphere Partner Gateway  
 構成 51

WebSphere Partner Gateway (続き)  
 統合  
 InterChange Server との 119  
 WebSphere Data Interchange との  
 205  
 WebSphere Message Broker との  
 187  
 WebSphere Process Server との 61  
 WebSphere Transformation Extender  
 Trading Manager との 238  
 WebSphere Transformation Extender  
 との 229  
 文書の受信 56  
 文書の送信 52  
 WebSphere Process Server とのインス  
 トール 64  
 WebSphere Partner Gateway - Express 6  
 WebSphere Process Server  
 概要 61  
 サポートされるバージョン 63  
 データ・バインディング 68  
 添付ファイル・コンテナ・ビジネ  
 ス・オブジェクト 72  
 統合の計画 63  
 統合のサンプル 65  
 トップレベル・オブジェクト 70  
 バックエンド統合パッケージ化デー  
 タ・バインディング 68  
 パッケージ化ヘッダー・ビジネス・オブ  
 ジェクト 73  
 ファイル・ディレクトリー・トランス  
 ポート・プロトコル 114  
 ペイロード・コンテナ・ビジネス・  
 オブジェクト 71  
 HTTP トランスポート・プロトコル  
 90  
 JMS トランスポート・プロトコル 98  
 Web サービス 110  
 WebSphere Transformation Extender 229  
 WebSphere Transformation Extender  
 Trading Manager 238  
 write メソッド 80, 87, 89

## X

XML スキーマ  
 イベント通知ペイロード 24  
 トランスポート・エンベロープ 38  
 XML 文書  
 サポートされるトランスポート・プロ  
 トコル 41, 42  
 パッケージ化の選択 44  
 ペイロードのロケーション 37  
 XMLEvent サービス・コンテンツ  
 説明 23

XMLEvent 文書  
 概要 23  
 サービス・コンテンツ 23  
 x-aux-IntelligibleCheckRequired 31  
 ebMS ソース 37  
 x-aux-create-datetime 29  
 x-aux-event-status-code 30  
 x-aux-in-file-name 30  
 x-aux-msg-id  
 説明 29  
 AS1 ソース 35  
 AS2 ソース 34  
 ebMS ソース 37  
 RosettaNet ソース 33  
 x-aux-payload-root-tag  
 説明 30  
 AS1 ソース 35  
 AS2 ソース 34  
 RosettaNet ソース 32  
 x-aux-process-instance-id  
 説明 30  
 ebMS ソース 36  
 RosettaNet ソース 33  
 x-aux-process-type  
 説明 29  
 AS1 ソース 35  
 AS2 ソース 34  
 ebMS ソース 36  
 RosettaNet ソース 32  
 x-aux-process-version  
 説明 29  
 AS1 ソース 35  
 AS2 ソース 34  
 ebMS ソース 36  
 RosettaNet ソース 32  
 x-aux-production  
 説明 30  
 RosettaNet ソース 33  
 x-aux-protocol  
 説明 29  
 AS1 ソース 35  
 AS2 ソース 33  
 ebMS ソース 36  
 RosettaNet ソース 32  
 x-aux-protocol-version  
 説明 29  
 AS1 ソース 35  
 AS2 ソース 33  
 ebMS ソース 36  
 RosettaNet ソース 32  
 x-aux-receiver-id  
 説明 29  
 AS1 ソース 34  
 AS2 ソース 33  
 ebMS ソース 36  
 RosettaNet ソース 32

x-aux-sender-id  
説明 29  
AS1 ソース 34  
AS2 ソース 33  
ebMS ソース 36  
RosettaNet ソース 32  
x-aux-SyncResponse 31  
ebMS ソース 36  
x-aux-system-msg-id  
説明 30  
AS1 ソース 35  
AS2 ソース 34  
ebMS ソース 36  
x-aux-third-party-bus-id 30  
ebMS ソース 36  
x-aux-TimeToAccept 31  
ebMS ソース 36  
x-aux-transport-retry-count 30  
x-out-filename 30







Printed in Japan