

**IBM WebSphere Business Integration  
Adapters**



## **Adapter for Ariba Buyer ユーザーズ・ガイド**

*Adapter* バージョン 3.2.x

**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、95 ページの『付録 D. 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 Business Integration Adapter for Ariba Buyer バージョン 3.2.x、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。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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：	IBM WebSphere Business Integration Adapters Adapter for Ariba Buyer User Guide Adapter Version 3.2.x
発 行：	日本アイ・ビー・エム株式会社
担 当：	ナショナル・ランゲージ・サポート

第1刷 2004.1

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、  
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2002, 2003. All rights reserved.

© Copyright IBM Japan 2004

---

# 目次

本書について	v
対象読者	v
関連文書	v
表記上の規則	vi
<b>本リリースの新機能</b>	<b>vii</b>
リリース 3.2.x の新機能	vii
リリース 3.1.x の新機能	vii
リリース 3.0.x 文書パッチの新機能	vii
<b>第 1 章 アダプターの概要</b>	<b>1</b>
アダプターのアーキテクチャー	1
アダプター・コンポーネント	3
コネクターの動作方法	5
<b>第 2 章 アダプターのインストール</b>	<b>9</b>
アダプター環境	9
インストール・プロセスの概要	11
アダプターおよび関連ファイルのインストール	11
インストール済みファイルの構造	11
インストールの完了	12
<b>第 3 章 アダプターの構成</b>	<b>13</b>
Ariba Buyer のアダプター用の構成	13
アダプターの構成	14
統合チャネルの構成	16
複数のコネクター・インスタンスの作成	18
アダプターの始動	19
コネクターの停止	21
ログ・ファイルとトレース・ファイルの使用	22
<b>第 4 章 ビジネス・オブジェクトの理解</b>	<b>25</b>
メタデータの定義	25
ビジネス・オブジェクト構造の概要	26
Ariba Buyer のビジネス・オブジェクト	26
ビジネス・オブジェクトの生成	31
<b>第 5 章 ビジネス・オブジェクト定義の生成</b>	<b>33</b>
ODA for Ariba Buyer の概要	33
ビジネス・オブジェクト定義の生成	33
ファイルのアップロード	42
<b>第 6 章 トラブルシューティングとエラー処理</b>	<b>43</b>
チャネル・エラー処理	43
コネクター・エラー処理	45
ODA エラー処理	48
トレース・メッセージ	49
トラブルシューティングのヒント	50
<b>付録 A. コネクターの標準構成プロパティ</b>	<b>51</b>

新規プロパティと削除されたプロパティ	51
標準コネクタ・プロパティの構成	51
標準プロパティの要約	53
標準構成プロパティ	57
<b>付録 B. Connector Configurator</b>	<b>69</b>
Connector Configurator の概要	69
Connector Configurator の始動	70
System Manager からの Configurator の実行	71
コネクタ固有のプロパティ・テンプレートの作成	71
新規構成ファイルの作成	74
既存ファイルの使用	75
構成ファイルの完成	76
構成ファイル・プロパティの設定	77
構成ファイルの保管	84
構成ファイルの変更	84
構成の完了	85
グローバル化環境における Connector Configurator の使用	85
<b>付録 C. 統合チャネルのマイグレーション</b>	<b>87</b>
概要	87
マイグレーション・プロセス	87
WebSphere Business Integration Adapter for Ariba Buyer のインストール	87
Ariba Buyer のマイグレーション	89
統合チャネルのマイグレーション	89
<b>付録 D. 特記事項</b>	<b>95</b>
プログラミング・インターフェース情報	96
商標	97

---

## 本書について

IBM<sup>®</sup> WebSphere<sup>®</sup> Business Integration Adapter ポートフォリオは、主要な e-business テクノロジー、エンタープライズ・アプリケーション、およびレガシー・システムとメインフレーム・システムに、統合接続性を提供します。製品セットには、ビジネス・プロセスの統合に向けてコンポーネントをカスタマイズ、作成、および管理するためのツールとテンプレートが含まれています。

本書では、IBM WebSphere Business Integration Adapter for Ariba Buyer のインストール、構成、ビジネス・オブジェクト開発、およびトラブルシューティングについて説明します。

---

## 対象読者

本書は、お客様のサイトでアダプターを使用するコンサルタント、開発者、およびシステム管理者を対象としています。

---

## 関連文書

この製品に付属する資料の完全セットで、すべての WebSphere Business Integration Adapters のインストールに共通な機能とコンポーネントについて説明します。また、特定のコンポーネントに関する参考資料も含まれています。

以下のサイトから、関連資料をインストールすることができます。

- アダプターの一般情報、WebSphere Message Brokers (WebSphere MQ Integrator、WebSphere MQ Integrator Broker、WebSphere Business Integration Message Broker) でのアダプターの使用、WebSphere Application Server でのアダプターの使用については、次の IBM WebSphere Business Integration Adapters InfoCenter をご覧ください。  
<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- WebSphere InterChange Server でのアダプターの使用については、次の IBM WebSphere InterChange Server InfoCenter をご覧ください。  
<http://www.ibm.com/websphere/integration/wicserver/infocenter>  
<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>
- WebSphere Message Brokers の詳細については、以下をご覧ください。  
<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>
- WebSphere Application Server の詳細については、以下を参照してください。  
<http://www.ibm.com/software/webservers/appserv/library.html>

上記のサイトには資料のダウンロード、インストール、および表示に関する簡単な説明が記載されています。

## 表記上の規則

本書では、以下のような規則を使用しています。

---

Courier フォント	コマンド名、ファイル名、入力情報、システムが画面に出力した情報などのリテラル値を示します。
イタリック 青のアウトライン	初出語、変数名、または相互参照を示します。 オンラインで表示したときのみ見られる青のアウトラインは、相互参照用のハイパーリンクです。アウトラインの内側をクリックすると、参照先オブジェクトにジャンプします。
{ }	構文の記述行の場合、中括弧 {} で囲まれた部分は、選択対象のオプションです。1 つのオプションのみを選択する必要があります。
	構文の記述行の場合、パイプで区切られた部分は、選択対象のオプションです。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、オプションのパラメーターです。
...	構文の記述行の場合、省略符号 ... は直前のパラメーターが繰り返されることを示します。例えば、option[,...] は、複数のオプションをコンマで区切って指定できることを意味します。
< >	1 つの名前の個々のエレメントを互いに区別するために、不等号括弧によって個々のエレメントが囲まれます。例えば、<server_name><connector_name>tmp.log のように使用します。
/, ¥	本書では、ディレクトリー・パスの規則として円記号 (¥) を使用します。UNIX システムの場合には、円記号をスラッシュ (/) に置き換えてください。すべての製品パス名は、Ariba Buyer のコネクタがインストールされているディレクトリーを基準とした相対パス名です。
%text% および \$text	パーセント (%) 符号で囲まれたテキストは、Windows 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 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。
ProductDir	製品のインストール先ディレクトリーを表します。IBM WebSphere InterChange Server の場合、デフォルトの製品ディレクトリーは IBM¥WebSphereICS です。IBM WebSphere Business Integration Adapters の場合、デフォルトの製品ディレクトリーは WebSphereAdapters です。

---

---

## 本リリースの新機能

---

### リリース 3.2.x の新機能

アダプターのインストール情報は、本書から移動しました。この情報の新たな入手先については、第 2 章を参照してください。

アダプターが Ariba Buyer のバージョン 8.1 をサポートするようになりました。

バージョン 3.2 以降の Adapter for Ariba Buyer は Microsoft Windows NT ではサポートされません。

---

### リリース 3.1.x の新機能

アダプターは、WebSphere Application Server を統合ブローカーとして使用できるようになりました。詳細については、9 ページの『アダプター環境』を参照してください。

アダプターは、以下のプラットフォーム上で実行されます。

- Solaris 7、8
- AIX 5.1、5.2
- HP-UX 11i

アダプターには、新たにアダプター固有の構成プロパティー B0Locale が加わりました。詳細については、14 ページの『アプリケーション固有の構成プロパティー』を参照してください。

---

### リリース 3.0.x 文書パッチの新機能

システムをアップグレードする場合、マイグレーション情報については 87 ページの『付録 C. 統合チャネルのマイグレーション』を参照してください。

この付録では、Ariba Buyer 7.x から Ariba Buyer 8.x にアップグレードする際に、Ariba Buyer 統合チャネルを以前のバージョンのアダプター、WebSphere Adapter for Ariba Buyer (WAAB) から現行バージョン WebSphere Business Integration Adapter for Ariba Buyer にマイグレーションする方法について説明します。





---

## 第 1 章 アダプターの概要

この章では、IBM WebSphere Business Integration Adapter for Ariba(R) Buyer および関連するシステム・アーキテクチャーについて説明します。

Adapter for Ariba Buyer は、コネクター、アダプター・フレームワーク、および統合チャネルから構成されます。これらが一体となることにより、Ariba Buyer は WebSphere MQ Integrator Broker や WebSphere InterChange Server (ICS) などの統合ブローカーと情報を交換できるようになります。

Adapter for Ariba Buyer バージョン 3.2 により、Ariba Buyer バージョン 8.1 との統合が可能になります。このアダプターは前のバージョンのアプリケーションとは互換性がありません。チャネル・コンポーネントは、アダプター・フレームワークのサポートの有無にかかわらず、Ariba Buyer をサポートするすべてのプラットフォームで稼働します。コネクター・コンポーネントは、アダプター・フレームワークをサポートしているすべてのプラットフォームで稼働します。

統合ブローカーとアダプターの関係の詳細については、「*IBM WebSphere Business Integration システム管理ガイド*」を参照してください。

本章の内容は、次のとおりです。

- 1 ページの『アダプターのアーキテクチャー』
- 5 ページの『コネクターの動作方法』

---

## アダプターのアーキテクチャー

Adapter for Ariba Buyer により、Ariba Buyer は SAP、Oracle Financials、および PeopleSoft などの外部アプリケーションと通信できます。

### コンポーネント

このアダプターには、以下の 3 つの基本コンポーネントが含まれます。

- **統合チャネル**  
Ariba Buyer がターゲット・アプリケーションと通信するために必要です。Ariba Buyer のコンポーネントとして稼働し、コネクターとリモートで通信します。
- **コネクター**  
データの転送において、統合ブローカーとアプリケーションの間の中継を行います。コネクターはメタデータ主導型です。
- **Object Discovery Agent (ODA)**  
アプリケーションによってエクスポートされたメタデータを読み取り、ビジネス・オブジェクト定義に変換する設計時のツール。コネクターは、実行時にこれらの定義を使ってアプリケーション・データをビジネス・オブジェクトに変換します。

2 ページの図 1 は、アダプター・コンポーネントの連携方法を示します。詳細については、3 ページの『アダプター・コンポーネント』で説明します。

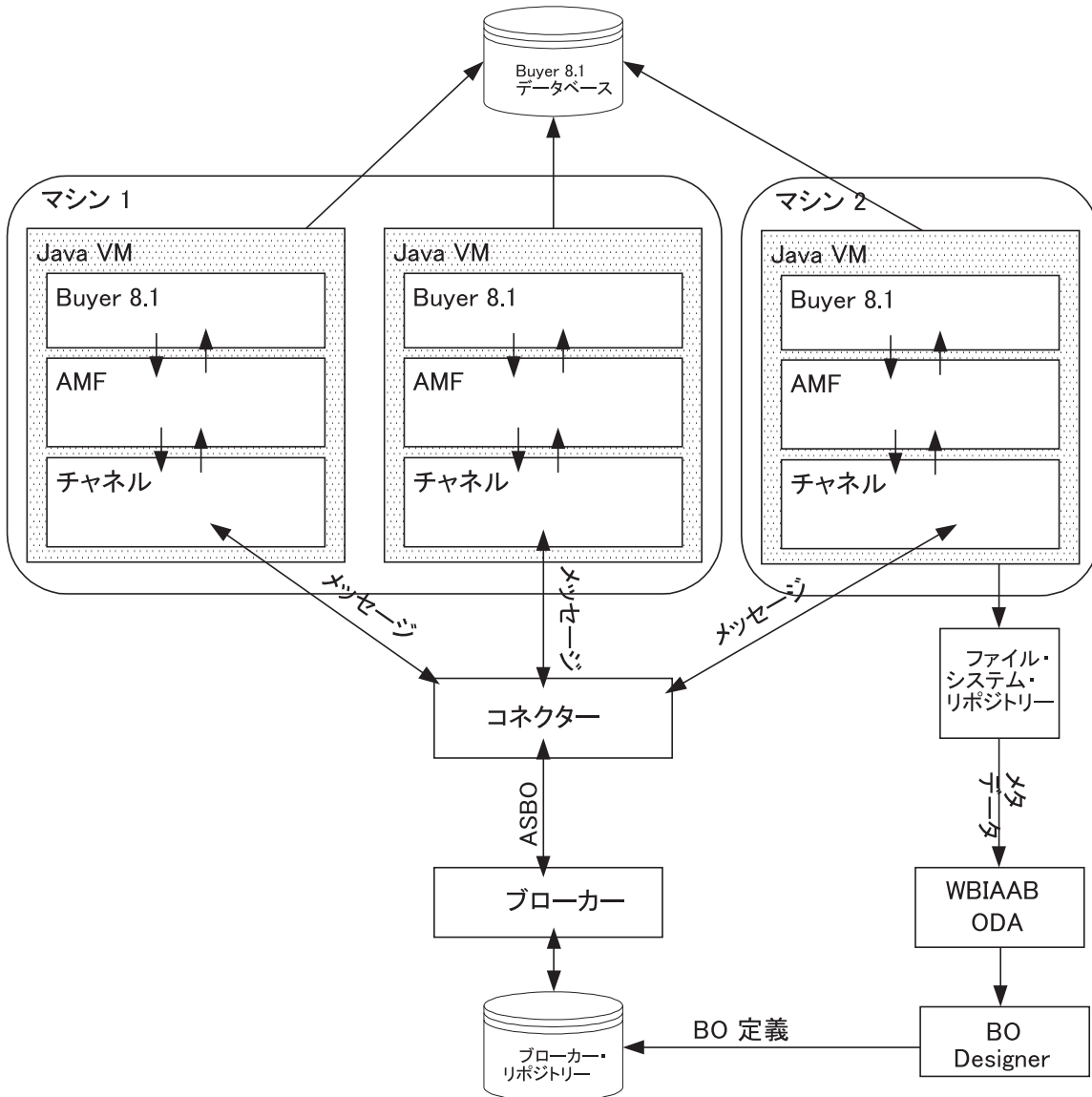


図1. アダプターのアーキテクチャー

## コネクターとノード

Ariba Buyer は、すべてのサーバーが同一のオペレーティング・システムを実行している場合には、すべてのノードが 1 つマシン上にあるマルチノード構成、あるいはマルチサーバー構成で多数のマシンに分散されるマルチノード構成での実行が可能です。マルチサーバー構成では、各マシンには少なくとも 1 つのノードがあります。ただし、ノードの数にかかわらず、コネクターのインスタンスは 1 つのみです。

各ノードは独自のプロセスで実行され、統合チャネルの独自のインスタンスと通信しますが、すべてのノードは同一の Ariba Buyer アプリケーション・データベースと通信します。

## 実行時のイベント・フロー

実行時のイベント・フローは双方向であり、方向はイベントが発生する場所によって決定します。

Ariba Buyer から発生するイベントの場合。

1. メッセージ・オブジェクトが、特定の Ariba Buyer ノードの統合チャンネルに送信されます。
2. 統合チャンネルがメッセージ・オブジェクトをコネクターに送信します。
3. コネクターは、メッセージ・オブジェクト内のビジネス・オブジェクト・データを、設計時に ODA によって作成されたビジネス・オブジェクト定義を使用して、アプリケーション固有のビジネス・オブジェクト (ASBO) にマップします。その後、ASBO を統合ブローカーに送信します。

統合ブローカーから発生するイベントの場合。

1. ブローカーはアプリケーション固有のビジネス・オブジェクトをコネクターに送信します。
2. コネクターは、オブジェクト内の応答キーの有無を調べることにより、アプリケーション固有のビジネス・オブジェクトが、以前に出された要求への応答であるかどうかを判断します。
3. アプリケーション固有のビジネス・オブジェクトは、メッセージ・オブジェクトに変換されます。
4. アプリケーション固有のビジネス・オブジェクトが前の要求に対する応答である場合は、メッセージ・オブジェクトが要求を発したノードにルーティングされます。それ以外の応答の場合、メッセージ・オブジェクトは使用可能な最初のノードにルーティングされます。

---

## アダプター・コンポーネント

3 つのアダプター・コンポーネントが、ブローカーとの間のデータ・フローを処理および管理します。以下に詳細を示します。

### 統合チャンネル

統合チャンネルは Ariba Buyer とコネクターとの間のリンクです。Ariba Buyer は Ariba Messaging Framework (AMF) を使用してチャンネルと通信します。アプリケーションは AMF を使用してメッセージ・オブジェクトをチャンネルに送信し、チャンネルは AMF を使用してオブジェクトをアプリケーションに送信します。

AMF とチャンネルとの間の通信は以下の場合に行われます。

- チャンネル上で定義された統合イベントが Ariba Buyer 内で起動された場合。AMF はイベントを同期または非同期で起動できます。
- 統合ブローカーからサービス呼び出しを受信した場合。呼び出しが使用可能な最初のチャンネル・ノードに送られます。
- Ariba Buyer 内でメタデータが変更された場合。これは 1 次ノード上で、開発モードにある時のみ発生します。

サービス呼び出しイベントは、コネクタによって最初に使用可能なチャネル・ノードに送られ、アプリケーションに中継されます。

詳細については、25 ページの『第 4 章 ビジネス・オブジェクトの理解』、および 33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。

## コネクタ

コネクタは、アプリケーション固有のコンポーネントとコントローラーおよび通信フレームワークの 2 つのパーツで構成されています。アプリケーション固有のコンポーネントには、特定のアプリケーションまたはテクノロジー（この場合は Ariba Buyer）用に調整されたコードが含まれます。コントローラーと通信フレームワークは、コードがすべてのコネクタに対して共通であるため、以下のサービスを提供することによって、統合ブローカーとアプリケーション固有のコンポーネントとの仲介役の機能を果たします。

- ビジネス・オブジェクトの受信と送信
- 始動メッセージや管理メッセージの交換の管理

Ariba Buyer 用のコネクタは統合チャネルを通じてアプリケーションと通信します。

統合チャネルとコネクタは実行時に相互に通信します。メッセージ・オブジェクトは Ariba Buyer ノードから発信され、AMF を通じて統合チャネルに渡されます。統合チャネルがメッセージ・オブジェクトをコネクタに送信します。

コネクタは、メッセージ・オブジェクト内のビジネス・オブジェクト・データを、ブローカー・リポジトリ内に保管されている定義を使用してアプリケーション固有のビジネス・オブジェクトにマップし、ブローカーに送信します。また Ariba Buyer に配送するために、ブローカーからビジネス・オブジェクトを受信することもできます。これらのビジネス・オブジェクトは、チャネル・インスタンスを通じて、交換を開始した Ariba Buyer ノードに返送されます。

詳細については、25 ページの『第 4 章 ビジネス・オブジェクトの理解』、および 33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。

## Object Discovery Agent (ODA)

Object Discovery Agent (ODA) は、Ariba Buyer 固有のメタデータからビジネス・オブジェクト定義を作成するために使用される設計ツールです。ODA は、そのメタデータを使用してビジネス・オブジェクト定義を作成します。

統合チャネルによって、次の 2 タイプのメタデータがエクスポートされます。

- **クラス・メタデータ**。これは Ariba Buyer オブジェクト・モデル内に定義されたビジネス・オブジェクトを表します。
- **メッセージ・メタデータ**。これは Ariba Buyer 内に定義された、統合メッセージを表します。これらのメッセージは Buyer ビジネス・オブジェクトを参照します。すなわち、メッセージ・メタデータがクラス・メタデータを参照します。

メタデータは開発モードで実行されている Ariba Buyer インスタンスの 1 次ノードからエクスポートできます。ODA はこれらファイルを読み取り、その情報を使用し

て、統合ブローカー・リポジトリ内のビジネス・オブジェクト定義を作成します。実行時に、コネクタはこれらの定義を使用して、メッセージ・オブジェクトをアプリケーション固有のビジネス・オブジェクトに変換することができます。

Ariba Buyer はメタデータに対するあらゆる変更を統合チャンネルに通知するインターフェースを提供します。チャンネルはこの情報をリスナーを通じて AMF から取得し、自動的にメタデータ・リポジトリに保管します。

**注:** Ariba Buyer サーバーをシャットダウンし、`initdb -reshapedb` コマンドを実行すると、リポジトリを再生成できます。メタデータを再生成するには、システムを開発モードにする必要があります。詳細については、Ariba Buyer 8.x のドキュメンテーションを参照してください。

詳細については、25 ページの『第 4 章 ビジネス・オブジェクトの理解』、および 33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。

---

## コネクタの動作方法

Ariba Buyer 用のコネクタは双方向です。ブローカーからアプリケーションに送信された要求だけでなく、Ariba Buyer アプリケーションから発信されたイベントも処理できます。

イベント通知のために、Ariba Buyer は統合チャンネル内のメッセージ・ハンドラーを使用するメカニズムを提供します。イベントを送信するために、Ariba Buyer ノードは関連するチャンネル・インスタンスでメッセージ・ハンドラーを呼び出し、そのメッセージ・ハンドラーがコネクタと通信します。すべてのイベントは、ポーリングなどの他の通知形式を必要とせずに、コネクタに送られます。

要求処理においては、Ariba Buyer 用のコネクタはビジネス・オブジェクトの形で統合ブローカーから受信した要求を処理します。コネクタから統合チャンネル内のサブスクライブ・メッセージ・ハンドラーに送信されたメッセージ・オブジェクトは、サブスクライブ・メッセージ・ハンドラーからアプリケーションに送信されます。

Ariba Buyer 用コネクタは、WebSphere Business Integration Adapter コネクタ用のメタデータ設計規則に従います。これは、コネクタ・コード・レベルでのコードの追加やカスタマイズを行わなくても新規のビジネス・オブジェクトを定義できるということを意味します。詳細については、25 ページの『第 4 章 ビジネス・オブジェクトの理解』を参照してください。

## イベントの処理

以下のセクションでは、コネクタがアプリケーション・イベントを処理する方法について説明します。

### イベントの検出と通知

Ariba Buyer ではイベントの検出は必要ありません。

アプリケーションは、イベント・データを含むメッセージ・オブジェクトを送信することにより、イベントを統合チャンネルに通知します。チャンネルはメッセージ・オブジェクトを受信すると、コネクタに送ります。コネクタでは、ビジネス・オ

プロジェクト・ハンドラーがメッセージをアプリケーション固有のビジネス・オブジェクトに変換し、統合ブローカーへ送信します。チャンネルのメッセージ・ハンドラーは、複数のイベントを同時に受信および転送でき、各イベントは異なる Buyer スレッドによって送信されます。

Ariba Buyer のイベントは非同期または同期です。アプリケーションはいずれかのタイプのイベントをチャンネルに送信する際に、応答を配信する戻りビジネス・オブジェクトを要求します。

### 同期イベント

Ariba Buyer が同期イベントを統合チャンネル送信する時には、メッセージ・オブジェクトをコネクターに送信し、応答を待つ間 Buyer スレッドのコントロールを保持します。コネクターは応答としてアプリケーション固有のビジネス・オブジェクトを受信すると、このオブジェクトをメッセージ・オブジェクトに変換し、待機中のチャンネル・ノードに返送します。

同期応答がチャンネルに返される前にコネクターが停止すると、イベントはエラーを生成します。詳細については、43 ページの『第 6 章 トラブルシューティングとエラー処理』のタイムアウトを参照してください。

### 非同期イベント

Ariba Buyer が非同期イベントを統合チャンネルに送信すると、メッセージにタイムアウトが指定されている場合は、チャンネルがタイムアウト・プロセスを開始します。続いてチャンネルがメッセージ・オブジェクトをコネクターに送信し、呼び出し元の Buyer スレッドにコントロールを戻します。コネクターは応答としてアプリケーション固有のビジネス・オブジェクトを受信すると、このオブジェクトをメッセージ・オブジェクトに変換し、発信元のチャンネル・ノードに返送します。

非同期イベントに応答が送信されない場合、または応答の送信が非常に遅い場合には、イベントはタイムアウトになります。詳細については、43 ページの『第 6 章 トラブルシューティングとエラー処理』のタイムアウトを参照してください。

### サービス呼び出し要求の処理

Ariba Buyer でサービス呼び出しが開始されると、コネクターがアプリケーション固有のビジネス・オブジェクトを統合ブローカーから受信し、メッセージ・オブジェクトに変換し、使用可能な最初のノードに渡します。受信ノード用のチャンネル・インスタンスはメッセージ・オブジェクトを受信し、Ariba Buyer で定義された API を使ってデータを Ariba Buyer に渡します。

### ステータスの更新

一般に、エラーは、障害のあるコンポーネントによって記録され、発信元のコンポーネントに通知されます。

統合チャンネル内で発生したエラーは、メッセージとともにアプリケーション API によって記録され、Ariba Buyer ログ・ファイル (AribaBuyerNodexLog.txt) に表示されます。

コネクタ内で発生したエラーは、メッセージとともにコネクタ・ログ・ファイルに表示され、またコネクタ・フレームワークに通知されます。コネクタ・ログ・ファイルの名前は、システムのセットアップ時に Connector Configurator 内で指定します。

詳細については、43 ページの『第 6 章 トラブルシューティングとエラー処理』を参照してください。





---

## 第 2 章 アダプターのインストール

本章では、IBM WebSphere Business Integration Adapter for Ariba Buyer のインストール方法を説明します。本章の内容は、次のとおりです。

- 『アダプター環境』
- 11 ページの『インストール・プロセスの概要』
- 11 ページの『アダプターおよび関連ファイルのインストール』
- 11 ページの『インストール済みファイルの構造』
- 12 ページの『インストールの完了』

---

### アダプター環境

アダプターをインストール、構成、使用する前に、環境要件を理解しておく必要があります。環境要件は、以下のセクションでリストされています。

- 『ブローカーの互換性』
- 10 ページの『アダプターのプラットフォーム』
- 10 ページの『アダプターの依存関係』
- 11 ページの『グローバリゼーション』

### ブローカーの互換性

アダプターが使用するアダプター・フレームワークは、アダプターと通信する統合ブローカーのバージョンとの互換性を備えている必要があります。Adapter for Ariba Buyer バージョン 3.2 は、以下のアダプター・フレームワークと統合ブローカーでサポートされています。

- **アダプター・フレームワーク:**  
WebSphere Business Integration Adapter Framework バージョン 2.3.1、および 2.4
- **統合ブローカー:**
  - WebSphere InterChange Server、バージョン 4.1.1、4.2、4.2.1、4.2.2
  - WebSphere MQ Integrator、バージョン 2.1.0
  - WebSphere MQ Integrator Broker、バージョン 2.1.0
  - WebSphere Business Integration Message Broker、バージョン 5.0
  - WebSphere Application Server Enterprise、バージョン 5.0.2  
(Studio Application Developer Integration Edition、バージョン 5.0.1 と併用)

例外については、『リリース情報』を参照してください。

**注:** 統合ブローカーおよびその前提条件のインストールに関する説明については、以下のガイドを参照してください。

WebSphere InterChange Server (ICS) については、「*IBM WebSphere InterChange Server システム・インストール・ガイド (UNIX 版)*」または「*IBM WebSphere InterChange Server システム・インストール・ガイド (Windows 版)*」を参照してください。

WebSphere Message Brokers については、「*WebSphere Message Brokers 使用ア*

「アダプター・インプリメンテーション・ガイド」を参照してください。  
WebSphere Application Server については、「アダプター実装ガイド (WebSphere Application Server)」を参照してください。

## アダプターのプラットフォーム

アダプターをインストールする前に、システムに以下のソフトウェアがインストールおよび構成されている必要があります。

### オペレーティング・システム:

以下のアプリケーション・プラットフォームのいずれか:

- AIX 4.3.3、AIX 5.1、AIX 5.2 (64 ビット)
- Solaris 7.0、Solaris 8.0
- HP UX 11.0、HP UX 11i
- Windows NT (SP6a)、Windows 2000 (SP3)

### データベース:

- DB2 7.2 (FP9)
- Oracle 8.1.7.4、Oracle 9.2
- MS SQL Server 7 (SP4)、MS SQL Server 2000

### サード・パーティー・ソフトウェア:

- Ariba Buyer 8.1 (このアダプターのバージョン 3.2 は Ariba Buyer 8.0 と互換性がありません)

**注:** アダプターのインストールは、Ariba Buyer 8.1 のインストールの前後いずれでも構いません。ただし、アダプターはアプリケーションを構成する前にインストールされている必要があります。Ariba Buyer の構成プロセスでは、アダプターのインストール・ディレクトリーから Ariba Buyer のインストール・ディレクトリーにファイルをコピーする必要があります。

アダプターをインストールすると、Ariba Buyer 8.1 を構成できるようになります。詳細については、Ariba Buyer 8.1 に関する文書を参照してください。

## アダプターの依存関係

Adapter for Ariba Buyer は表 1に示すように、クライアント・ライブラリーまたは API に依存しています。

表 1. アダプターの依存関係

ライブラリー/API	バージョン	オペレーティング・システム
ariba.util.core.zip	Ariba Buyer 8.1	サポートされるすべての OS
ariba.util.messaging.zip	Ariba Buyer 8.1	サポートされるすべての OS
ariba.amf.zip	Ariba Buyer 8.1	サポートされるすべての OS

表 1. アダプターの依存関係 (続き)

ライブラリー/API	バージョン	オペレーティング・システム
ariba.base.zip	Ariba Buyer 8.1	サポートされるすべての OS
ariba.tools.migration.zip	Ariba Buyer 8.1	サポートされるすべての OS
ariba.tools.taskharness.zip	Ariba Buyer 8.1	サポートされるすべての OS
jakarta-oro-2.0.jar	2.0	サポートされるすべての OS

## グローバル化

このアダプターは DBCS (2 バイト文字セット) で使用可能であり、翻訳されています。

---

## インストール・プロセスの概要

Adapter for Ariba Buyer をインストールするには、次のステップに従います。

1. 統合ブローカーのインストール。
2. Ariba Buyer のインストール。

注: インストールの最後に、Ariba Buyer の構成プログラムに進むかどうかをたずねるプロンプトが出された場合は、「いいえ」を選択してください。

3. Ariba Buyer 用アダプターをインストール。
4. Ariba Buyer の構成および初期化。

ここでは、ステップ 1 および 2 は完了していることを前提とします。本書のインストールの説明では、ステップ 3 および 4 を対象としています。

---

## アダプターおよび関連ファイルのインストール

WebSphere Business Integration アダプター製品のインストールについては、次のサイトで WebSphere Business Integration Adapters Infocenter にある「*WebSphere Business Integration Adapters* インストール・ガイド」を参照してください。

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

---

## インストール済みファイルの構造

12 ページの表 2 には、Adapter for Ariba Buyer が使用するファイル構造が示されており、また、システムにインストールされたファイルがリストされています。

注:

1. 本書では、ディレクトリー・パスとして円記号 (¥) を使用しています。UNIX システムの場合には、円記号をスラッシュ (/) に置き換えてください。

- すべての製品パス名は、システム上の、製品のインストール先ディレクトリーを基準とした相対パス名です。

表 2. システムにインストールされたアダプターのファイル

%Prod_Dir% のサブディレクトリー	説明
_uninstall	アダプターをアンインストールするためのファイルが入っています。
connectors¥Buyer	内容: <ul style="list-style-type: none"> <li>コネクタ・コードを含むアダプターの CWBuyer.jar ファイル、バージョン 3.2.0</li> <li>start_Buyer.bat ファイル (WIN) または start_Buyer.sh ファイル (UNIX)</li> <li>Ariba Buyer のインストール・ディレクトリーにコピーされる、¥Buyer ディレクトリー</li> </ul>
connectors¥messages	エラー・メッセージなどのメッセージがリストされた、BuyerConnector.txt ファイルが入っています。
legal¥license	アダプター使用許諾契約条件を記載したテキスト・ファイルが入っています。
ODA¥Buyer	内容: <ul style="list-style-type: none"> <li>ODA コードを含む、Object Discovery Agent BuyerODA.jar ファイル、バージョン 3.2.0</li> <li>start_BuyerODA.bat ファイル (WIN) または start_BuyerODA.sh ファイル (UNIX)</li> </ul>
ODA¥messages	エラー・メッセージなどのメッセージがリストされた、BuyerODAAgent.txt ファイルが入っています。
repository¥Buyer	コネクタ構成プロパティーを指定する、CN_Buyer.txt ファイルが入っています。

## インストールの完了

¥Buyer サブディレクトリーは、¥connectors¥Buyer ディレクトリーの下にあります (表 2 を参照)。このサブディレクトリー内のファイルには、アダプター・チャネルに関する特定の構成情報が含まれています。

インストールを完了するには、このサブディレクトリーを Ariba Buyer 8 のホーム・ディレクトリーの上部にコピーし、以下のようにクラス・パスを更新します。

- ¥Buyer サブディレクトリーをソース・サブディレクトリーから <Ariba Buyer Server root>¥、例えば ¥Ariba¥app¥Buyer¥ などの上部にコピーします。
- 以下のエントリーを追加して /Ariba/app/Buyer/classes/classpath.txt を変更します。

```
#IBM ERP Channel インプリメンテーション
```

```
classes/IBMChannel.jar
```

```
classes/ABCommon.jar
```

```
classes/BIA_IBMMigration.jar(WAAB 1.x または 2.x からマイグレーションしている場合にのみ必要。詳細は、87 ページの『付録 C. 統合チャネルのマイグレーション』を参照。)
```

- ファイルを保管します。これで Ariba Buyer を構成する準備が完了しました。

---

## 第 3 章 アダプターの構成

この章では、IBM WebSphere Business Integration Adapter for Ariba Buyer の構成の方法、およびアダプターと連携させるためのアプリケーションの構成方法を説明します。本章の内容は、次のとおりです。

- 『Ariba Buyer のアダプター用の構成』
- 14 ページの『アダプターの構成』
- 16 ページの『統合チャネルの構成』
- 19 ページの『アダプターの始動』
- 21 ページの『コネクターの停止』
- 22 ページの『ログ・ファイルとトレース・ファイルの使用』

---

### Ariba Buyer のアダプター用の構成

コネクターを使用するには、まず Ariba Buyer を構成してアダプターを認識させる必要があります。Ariba Buyer Configure ツールを使用して、次のように正しい Enterprise Application Integration (EAI) チャネルを Ariba Buyer で設定します。

1. 「Ariba Buyer 構成 (Ariba Buyer Configuration)」ウィンドウで「**EAI チャネル (EAI Channel)**」をチェックして、「次へ」をクリックします。
2. お使いのデータベース、例えば IBM DB2 UDB などを選択し、「次へ」をクリックします。
3. お使いのデータベースのデータベース・プロパティを入力し、「次へ」をクリックします。
4. Ariba Buyer 8 Hot Fix Pack 1 がインストール済みであれば、EAI チャネルとして「**IBM WBI Adapter**」を選択し、「構成ファイルのロケーション (Configuration File Location)」を以下のように設定します。  
<Ariba Buyer Server root>/channels/IBM
5. Ariba Buyer 8 Hot Fix Pack 1 をインストールしていない場合 EAI チャネルとして「**その他 (Other)**」を選択します。  
「構成ファイルのロケーション (Configuration File Location)」を以下のように設定して、チャネル名として **IBM** を設定します。  
<Ariba Buyer Server root>/channels/IBM
6. Ariba Buyer 8.x の構成を続けます。

注: このアダプターは、構成時に「**基本 CSV 構成 (Basic CSV configuration)**」以外の ERP タイプを選択した場合にのみ使用されます。

## アダプターの構成

アダプターのコネクタ・コンポーネントには、標準構成プロパティとアプリケーション固有の構成プロパティの、2つのタイプの構成プロパティがあります。コネクタを実行する前に、これらのプロパティの値を設定する必要があります。

### 標準構成プロパティ

標準コネクタ構成プロパティを構成するには、Connector Configurator ツールを使用します。詳細については、69 ページの『付録 B. Connector Configurator』に記載されています。このツールはコネクタ構成用のグラフィカル・ユーザー・インターフェースを提供します。「標準構成プロパティ (Standard Config Properties)」タブをクリックして、構成プロパティを追加または変更します。

コネクタの構成プロパティへの値の指定が終了すると、Connector Configurator は値をアダプタ・リポジトリに保管する (ICS の場合) か、構成ファイルを生成してアダプタのローカル・リポジトリに配置 (WebSphere MQ Integrator Broker または WebSphere Application Server の場合) します。

コネクタは、始動時に構成値を取得します。実行時のセッション中に、1 つ以上のコネクタ・プロパティの値を変更できます。

- AgentTraceLevel など、一部の構成プロパティへの変更は動的であり、即時に有効になります。
- それ以外のコネクタ・プロパティへの変更は静的であり、変更後にコンポーネントまたはシステムの再始動が必要です。

プロパティが動的と静的のいずれであるかを判別するには、Connector Configurator の「更新メソッド」欄を参照してください。

### アプリケーション固有の構成プロパティ

アプリケーション固有のコネクタ構成プロパティは、コネクタが実行時に必要とする情報を提供します。また、コネクタ内の静的情報やロジックを、再コーディングや再ビルドを行わずに変更する方法も提供します。

これらのプロパティを構成するには、Connector Configurator を使用します。「アプリケーション構成プロパティ」タブをクリックして、構成プロパティの追加または変更を行います。詳細については、69 ページの『付録 B. Connector Configurator』を参照してください。

表 3 には、コネクタのアプリケーション固有の構成プロパティが、説明および指定可能な値とともにリストされています。

表 3. Ariba Buyer 用のアプリケーション固有の構成プロパティ

プロパティ	説明	指定可能な値	デフォルト値	必要
Nodes	Ariba Buyer ノード名および完全修飾マシン名	<url1>:Node1; <url2>:Node2;... <urln>:NodeN	なし	はい

表 3. Ariba Buyer 用のアプリケーション固有の構成プロパティ (続き)

プロパティ	説明	指定可能な値	デフォルト値	必要
ConnectorPort	コネクタの RMI ポート	任意のオープン・ポート - Parameters.table ファイルで指定された値と一致する必要があります。	2226	はい
ChannelPort	チャネルの RMI ポート	任意のオープン・ポート - Parameters.table ファイルで指定された値と一致する必要があります。	2225	はい
DateFormats	有効な日付形式を指定。詳細については、以下の『DateFormat』を参照してください。	<format>;<format>;...	EEE MMM dd HH:mm:ss z yyyy;M/d/yy; M/d/yyyy; M-d-yy; M-d yyyy; yyyyMMdd	いいえ
VariantMappings	ユーザー定義のバリエーション名を特定の ERP タイプと関連させます。バリエーションおよび区画固有のビジネス・オブジェクト名を、より一般的なビジネス・オブジェクト名にマップします。	<variant name>=<variant type prefix>;<variant name>=<variant type prefix>;...	なし	いいえ
BOLocale	ビジネス・オブジェクトのロケールを指定します。詳細については、以下の『BOLocale』を参照してください。	有効なロケール指定、例えば en_US、ja_JP、または de_DE です。所定の値が無効である場合は、システム・デフォルト・ロケールが使用されます。	なし	いいえ

### DateFormat

Java API によって決定される、有効な日付形式の詳細については、Web サイト <http://java.sun.com/j2se/1.3/docs/api/java/text/SimpleDateFormat.html> を参照してください。

### BOLocale

これは階層プロパティです。以下のように使用することができます。

- 親レベルでのみ BOLocale を定義することにより、すべてのビジネス・オブジェクトに対して単一のロケールを指定します。
- 親レベルで BOLocale を定義し、さらに BOLocale で子のプロパティも宣言することにより、単一の区画に対してロケールを指定します。子のプロパティの名前は、ビジネス・オブジェクトに指定された区画名と一致する必要があります。



## 統合チャネルの構成

統合チャネルは次の 2 つの目的で構成する必要があります。

- コネクターとの通信
- 統合イベントの処理

### チャネル・プロパティの構成

統合チャネル構成プロパティは、実行時にチャネルがコネクターとの通信に必要とする情報を提供します。これらのプロパティは、以下の Ariba Buyer インストール・ディレクトリーにある `Parameters.table` テキスト・ファイルで設定されています。

```
¥Ariba¥app¥Buyer¥Server¥config
```

統合チャネル・プロパティを構成するには、`Parameters.table` ファイルを開いてプロパティ値を入力します。

表 4 には、必要な構成プロパティが、説明および指定可能な値とともにリストされています。

表 4. 統合チャネル用のアプリケーション固有の構成プロパティ

プロパティ	説明	指定可能な値	デフォルト値	必要
ConnectorURL	コネクターが実行しているマシンのホスト名	完全修飾マシン名または IP アドレス	localhost	はい
ConnectorPort	コネクターの RMI ポート	任意のオープン・ポート - コネクターに設定された値と一致する必要があります。	2226	はい
ChannelPort	チャネルの RMI ポート	任意のオープン・ポート - コネクターに設定された値と一致する必要があります。	2225	はい
MetadataRepos	生成されたメタデータが保管されるロケーション	¥Ariba¥app¥Buyer¥Server からの相対パスまたは完全修飾ディレクトリー・パス	"/channels/ IBM/metadata";	はい

### 統合イベントの構成

各統合イベントを統合チャネルを通じて送信できるようにするには、事前に構成する必要があります。イベントは、`MessageDefinition.table` および `MessageConfiguration.table` という 2 つの構成ファイルで構成されます。

Ariba Buyer は数多くの創造的な統合イベントを提供しており、これらの統合イベントは「*Ariba Buyer Configuration Reference Guide*」にリストされています。ただし、統合チャネル上で構成されていないイベントは、Ariba Buyer 用アダプターを介して送信されません。

**MessageDefinition.table** ファイルには、任意の ERP バリエーション・タイプの各統合イベントに関する一般情報が含まれています。Ariba Buyer は、これらのバリエーションそれぞれの創造的な統合イベントの基本セットに、テーブル・エントリーを提供します。



**MessageConfiguration.table** ファイルには、チャンネル固有データなど、任意の区画の各統合イベントに関するより特定された情報が含まれています。各イベントのエントリーは、このテーブルで作成する必要があります。

MessageConfiguration.table のエントリー例を以下に示します。

```
PurchaseOrderPush={
  Channel={
    Name=IBM;
    Operation="Update Elements Only";
    Verb=Retrieve;
    Timeout=300000;
  };
  LoggingName=PurchaseOrderEvent;
  MessageParameters={
    Request={
      MergedSchemaName="ariba.integration.param.PurchaseOrderPush";
      Parameters={EventSource="ibmcsvp:PO:SAP3.1H"};
      SchemaName=ariba.integration.param.SimpleParams;
    };
  };
  TopicName=PurchaseOrderPush;
  ExecutionNode="Node2";
};
```

すべての MessageConfiguration.table エントリーには、チャンネル・パラメーター用のセクション (上記の例では最初のサブエントリーとして示されています) が含まれています。表 5 は、統合チャンネル用の有効なパラメーターと値をリストしています。

表 5. 統合チャンネル用のイベント構成パラメーター

パラメーター	説明	指定可能な値
Name	ERP 統合チャンネル名を指定します。	IBM
Operation	Ariba Buyer が実行できるオペレーションを指定します。詳細については、28 ページの『ビジネス・オブジェクト動詞』を参照してください。	Create、Delete、Load、“Load And Delete”、Update、“Update And Delete”、“Update Elements Only”
Verb	宛先のアプリケーションが実行可能なオペレーションを指定します。詳細については、28 ページの『ビジネス・オブジェクト動詞』を参照してください。	Create、Delete、Retrieve、Update、ABUpdate、Load、Update_And_Delete、Update_Elements_Only
Timeout	Ariba Buyer イベント・プロセスのタイムアウトを指定します。値はミリ秒単位で指定されます。	整数 >=0 デフォルト値は 0 で、そのイベントにはタイムアウトがないことを意味します。
EventSource	このイベントに関連したデータを一意的に識別します。このパラメーターが必要となるのはサブスクライプ・イベントの場合のみです。	任意のストリング値。最長 50 文字。それぞれの区画ごとに固有である必要があります。また、統合イベントの合間に変更することはできません。

統合イベントのその他のパラメーターの構成に関する完全な詳細については、「*Ariba Buyer Configuration Reference Guide*」を参照してください。

## 複数のコネクタ・インスタンスの作成

コネクタの複数のインスタンスを作成する作業は、いろいろな意味で、カスタム・コネクタの作成と同じです。以下に示すステップを実行することによって、コネクタの複数のインスタンスを作成して実行するように、ご使用のシステムを設定することができます。次のようにする必要があります。

- コネクタ・インスタンス用に新規ディレクトリを作成します。
- 必要なビジネス・オブジェクト定義が設定されていることを確認します。
- 新規コネクタ定義ファイルを作成します。
- 新規始動スクリプトを作成します。

## 新規ディレクトリの作成

それぞれのコネクタ・インスタンスごとにコネクタ・ディレクトリを作成する必要があります。このコネクタ・ディレクトリには、次の名前を付けなければなりません。

```
ProductDir¥connectors¥connectorInstance
```

ここで `connectorInstance` は、コネクタ・インスタンスを一意的に示します。

コネクタに、コネクタ固有のメタオブジェクトがある場合、コネクタ・インスタンス用のメタオブジェクトを作成する必要があります。メタオブジェクトをファイルとして保管する場合は、次のディレクトリを作成して、ファイルをそこに格納します。

```
ProductDir¥repository¥connectorInstance
```

## ビジネス・オブジェクト定義の作成

各コネクタ・インスタンスのビジネス・オブジェクト定義がプロジェクト内にまだ存在しない場合は、それらを作成する必要があります。

1. 初期コネクタに関連付けられているビジネス・オブジェクト定義を変更する必要がある場合は、適切なファイルをコピーし、**Business Object Designer** を使用してそれらのファイルをインポートします。初期コネクタの任意のファイルをコピーできます。変更を加えた場合は、名前を変更してください。
2. 初期コネクタのファイルは、次のディレクトリに入っていないければなりません。

```
ProductDir¥repository¥initialConnectorInstance
```

作成した追加ファイルは、`ProductDir¥repository` の適切な `connectorInstance` サブディレクトリ内に存在している必要があります。

## コネクタ定義の作成

**Connector Configurator** 内で、コネクタ・インスタンスの構成ファイル (コネクタ定義) を作成します。これを行うには、以下のステップを実行します。

1. 初期コネクタの構成ファイル (コネクタ定義) をコピーし、名前変更します。
2. 各コネクタ・インスタンスが、サポートされるビジネス・オブジェクト (および関連メタオブジェクト) を正しくリストしていることを確認します。
3. 必要に応じて、コネクタ・プロパティをカスタマイズします。

## 始動スクリプトの作成

始動スクリプトは以下のように作成します。

1. 初期コネクターの始動スクリプトをコピーし、コネクター・ディレクトリーの名前を含む名前を付けます。  
`dirname`
2. この始動スクリプトを、18ページの『新規ディレクトリーの作成』で作成したコネクター・ディレクトリーに格納します。
3. 始動スクリプトのショートカットを作成します (Windows のみ)。
4. 初期コネクターのショートカット・テキストをコピーし、新規コネクター・インスタンスの名前に一致するように (コマンド行で) 初期コネクターの名前を変更します。

これで、ご使用の統合サーバー上でコネクターの両方のインスタンスを同時に実行することができます。

カスタム・コネクター作成の詳細については、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

---

## アダプターの始動

Ariba Buyer 用のアダプターを始動するには、以下の手順を実行する必要があります。

1. Ariba Buyer 8.x を始動します。これにより、統合チャンネルが開始されます。
2. コネクターを始動します。

**注:** ビジネス・オブジェクト定義を追加または再ロードする場合は、第 3 のステップとして ODA の始動が必要です。

### Ariba Buyer の始動

Ariba Buyer を始動するための手順は、システムのノード構成に応じて異なります。詳細については、Ariba Buyer 8.x のドキュメンテーションを参照してください。

各 Buyer ノードは統合チャンネルのインスタンスを開始します。チャンネルが開始されると、チャンネルはコネクターへの接続を試行します。チャンネルは、コネクターが実行されているか否かにかかわらず開始されます。

### コネクターの始動

コネクターは、**コネクター始動スクリプト**を使用して明示的に始動する必要があります。始動スクリプトは、次に示すようなコネクターのランタイム・ディレクトリーに存在していなければなりません。

`ProductDir¥connectors¥connName`

ここで、`connName` はコネクターを示します。始動スクリプトの名前は、表 6 に示すように、オペレーティング・システム・プラットフォームによって異なります。

表 6. コネクターの始動スクリプト

オペレーティング・システム	始動スクリプト
UNIX ベースのシステム	connector_manager_connName
Windows	start_connName.bat

コネクタ始動スクリプトは、以下に示すいずれかの方法で起動することができます。

- Windows システムで「スタート」メニューから。  
「プログラム」>「IBM WebSphere Business Integration Adapters」>「アダプター」>「コネクタ」を選択します。デフォルトでは、プログラム名は「IBM WebSphere Business Integration Adapters」となっています。ただし、これはカスタマイズすることができます。あるいは、ご使用のコネクタへのデスクトップ・ショートカットを作成することもできます。
- コマンド行から。
  - Windows システム:  
start\_connName connName brokerName [-cconfigFile ]
  - UNIX ベースのシステム:  
connector\_manager\_connName -start

ここで、*connName* はコネクタの名前であり、*brokerName* は以下のご使用の統合ブローカーを表します。

- WebSphere InterChange Server の場合は、*brokerName* に ICS インスタンスの名前を指定します。
- WebSphere Message Brokers (WebSphere MQ Integrator、WebSphere MQ Integrator Broker、または WebSphere Business Integration Message Broker) または WebSphere Application Server の場合は、*brokerName* にブローカーを示すストリングを指定します。

**注:** Windows システム上の WebSphere Message Broker または WebSphere Application Server の場合は、-c オプションに続いてコネクタ構成ファイルの名前を指定しなければなりません。ICS の場合は、-c はオプションです。

- Adapter Monitor から (WebSphere Business Integration Adapters 製品のみ)。  
Adapter Monitor は System Manager 始動時に起動されます。  
このツールを使用して、コネクタのロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除を行うことができます。
- System Monitor から (WebSphere InterChange Server 製品のみ)。  
このツールを使用して、コネクタのロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除を行うことができます。
- Windows システムでは、Windows サービスとして始動するようにコネクタを構成することができます。この場合、Windows システムがブートしたとき (自動サービスの場合)、または Windows サービス・ウィンドウを通じてサービスを始動したとき (手動サービスの場合) に、コネクタが始動します。

コマンド行の始動オプションなどのコネクターの始動方法の詳細については、以下の資料のいずれかを参照してください。

- WebSphere InterChange Server については、「システム管理ガイド」を参照してください。
- WebSphere Message Brokers については、「*WebSphere Message Brokers 使用アダプター・インプリメンテーション・ガイド*」を参照してください。
- WebSphere Application Server については、「*アダプター実装ガイド (WebSphere Application Server)*」を参照してください。

## ODA の始動

ODA は、start\_BuyerODA.bat または start\_BuyerODA.sh スクリプトを実行すると始動します。Ariba Buyer サーバーを参照するパスを、次のように指定する必要があります。

- スクリプト内で BUYERPATH 変数を指定する。または
- Ariba Buyer サーバー・パスを、以下のようにコマンド行パラメーターとして指定する。

```
C:%Ariba%app%Buyer%Server
```

**注:** Ariba Buyer アプリケーションとコネクターが別々のマシンにインストールされている場合、Ariba Buyer マシンのファイル・システムをコネクターがインストールされているマシンのファイル・システムにマップするか、コネクターがインストールされているマシンのファイル・システムから Ariba Buyer マシンのファイル・システムにアクセスできるようにして、BUYERPATH を設定します。

ODA は、Business Object Designer が使用します。Business Object Designer が ODA に接続すると、ODA は Ariba Buyer メタデータ定義をアダプター・リポジトリから読み取って処理し、ビジネス・オブジェクト定義を構成します。

---

## コネクターの停止

コネクターを停止する方法は、以下に示すように、コネクターが始動された方法によって異なります。

- コマンド行からコネクターを始動した場合は、コネクター始動スクリプトを用いて、以下の操作を実行します。
  - Windows システムでは、始動スクリプトを起動すると、そのコネクター用の別個の「コンソール」ウィンドウが作成されます。このウィンドウで、「Q」と入力して Enter キーを押すと、コネクターが停止します。
  - UNIX ベースのシステムでは、コネクターはバックグラウンドで実行されるため、別ウィンドウはありません。代わりに、次のコマンドを実行してコネクターを停止します。

```
connector_manager_connName -stop
```

ここで、*connName* はコネクターの名前です。

- Adapter Monitor から (WebSphere Business Integration Adapters 製品のみ)。Adapter Monitor は System Manager 始動時に起動されます。

このツールを使用して、コネクタのロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除を行うことができます。

- **System Monitor から (WebSphere InterChange Server 製品のみ)**  
このツールを使用して、コネクタのロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除を行うことができます。
- **Windows システムでは、Windows サービスとして始動するようにコネクタを構成することができます。**この場合、Windows システムのシャットダウン時に、コネクタは停止します。

## ログ・ファイルとトレース・ファイルの使用

アダプターのコンポーネントは、複数レベルのメッセージ・ロギングとトレースを提供します。

**統合チャネル**は、情報メッセージ、デバッグ・メッセージ、およびエラー・メッセージを記録します。これらは、Ariba Buyer 管理者コンソール、およびメッセージ発信元の Ariba Buyer ノードのログ・ファイルに出力されます。表 7 にはメッセージのタイプとデフォルト設定がリストされています。

表 7. 統合チャネルのログおよびトレース・メッセージのレベル

メッセージ・タイプ	説明	デフォルト
情報	キー初期化および終了ステップ、メッセージ・デリバリーおよび受信イベントを記録します。	オン
デバッグ	メソッドの呼び出しおよび各呼び出し内のクリティカル・ポイントを記録します。	オフ
エラー	チャネル内のエラー状態、またはチャネルの検出したのエラー状態をすべて記録します。	オン

**コネクタ**はエラー・メッセージ、通知メッセージ、およびトレース・メッセージを記録するために、アダプター・フレームワークを使用します。エラー・メッセージと通知メッセージはログ・ファイルに記録され、トレース・メッセージとトレース・レベル (0 から 5) はトレース・ファイルに記録されます。

表 8 には、各トレース・レベルで記録されるメッセージのタイプが記載されています。

表 8. トレース・ファイルのトレース・レベル

トレース・レベル	説明
0	<ul style="list-style-type: none"> <li>• Ariba Buyer ODA からのエラーと致命的エラーを記録します。</li> <li>• システム管理者の対応が必要な警告を記録します。</li> <li>• アプリケーションからの重要なメッセージを記録します。</li> </ul>
1	メソッドのすべての開始メッセージおよび終了メッセージをトレースします。
2	ODA のプロパティとそれらの値をトレースします。
3	すべてのビジネス・オブジェクトの名前をトレースします。
4	ビジネス・オブジェクトのプロパティと、受信した値をトレースします。

表 8. トレース・ファイルのトレース・レベル (続き)

トレース・レベル	説明
5	<ul style="list-style-type: none"> <li>• すべての ODA プロパティの ODA 初期化値を示します。</li> <li>• ビジネス・オブジェクト定義のダンプをトレースします。</li> </ul>

ログ・ファイルとトレース・ファイルの名前、およびトレース・レベルをConnector Configurator で構成してください。詳細については、69 ページの『付録 B. Connector Configurator』を参照してください。

**ODA** にはロギング機能はなく、エラー・メッセージは直接ユーザー・インターフェースに送られます。トレース・ファイルとトレース・レベルは Business Object Designer 内で構成します。構成の手順は 35 ページの『エージェントの構成』に記載されています。ODA トレース・レベルはコネクタ・トレース・レベルと同一であり、上記の 22 ページの表 8 に記載されています。

エラー処理とトラブルシューティングの詳細については、43 ページの『第 6 章 トラブルシューティングとエラー処理』を参照してください。





---

## 第 4 章 ビジネス・オブジェクトの理解

本章では、ビジネス・オブジェクトの構造、アダプターによるビジネス・オブジェクトの処理方法、およびアダプターがそれらを行うにあたっての前提条件について説明します。

本章の内容は、次のとおりです。

- 25 ページの『メタデータの定義』
- 26 ページの『ビジネス・オブジェクト構造の概要』
- 26 ページの『Ariba Buyer のビジネス・オブジェクト』
- 31 ページの『ビジネス・オブジェクトの生成』

---

### メタデータの定義

Ariba Buyer のアダプターはメタデータ主導型です。WebSphere Business Integration システム内では、メタデータは Ariba Buyer によってエクスポートされた、データ構造を記述するアプリケーション固有の情報であると定義されます。メタデータは、コネクタが実行時にビジネス・オブジェクトの構築に使用する、ビジネス・オブジェクト定義の構成に使用されます。

メタデータ主導型アダプターは、サポートする各ビジネス・オブジェクトを処理する際に、ビジネス・オブジェクト定義にエンコードされたメタデータに従って処理を行います。これによりアダプターは、コードを変更することなく、新規または変更されたビジネス・オブジェクト定義を処理できます。あらゆる変更は Business Object Designer 内の Object Discovery Agent を通じて行われます。

アプリケーション固有のメタデータには、ビジネス・オブジェクトの構造と、その属性プロパティの設定が含まれています。各ビジネス・オブジェクトの実際のデータ値は、実行時にメッセージ・オブジェクトによって伝達されます。

アダプターには、サポートするビジネス・オブジェクトの構造、親と子のビジネス・オブジェクト間の関係、およびデータのフォーマットに関する前提事項があります。したがって、ビジネス・オブジェクトの構造が Ariba Buyer 内の対応するオブジェクトに定義されている構造と正確に一致することは重要であり、一致しない場合はアダプターがビジネス・オブジェクトを正しく処理できません。

**注:** Business Object Designer を使用してビジネス・オブジェクトの構造を変更してはいけません。

ビジネス・オブジェクトの構造を変更する必要がある場合は、Ariba Buyer 内の対応するオブジェクトを変更してから、その変更をファイル・システム・リポジトリにエクスポートし、ODA に入力します。詳細については、33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。

ビジネス・オブジェクト定義の変更の詳細については、「*WebSphere Business Integration Adapters* ビジネス・オブジェクト開発ガイド」を参照してください。

---

## ビジネス・オブジェクト構造の概要

WebSphere Business Integration システムでは、ビジネス・オブジェクト定義は以下のものから構成されます。

- タイプ名
- サポートされる動詞
- 属性

アプリケーション固有のビジネス・オブジェクトはビジネス・オブジェクト定義の特別なインスタンスであり、特定のアプリケーションのデータ構造および属性プロパティを反映します。

一部の属性 (含まれるデータではなく) は、そのオブジェクトのデータを含む、子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を指します。親レコードと子レコード間のデータはキーにより関連付けられます。

アダプターのビジネス・オブジェクトにはフラットなものと同層構造のものがあります。フラットなビジネス・オブジェクトは単純属性、すなわち単一の値 (ストリングなど) を表す属性のみを含み、子ビジネス・オブジェクトを参照することはありません。階層ビジネス・オブジェクトは単純属性、および属性値を含む子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を含みます。

カーディナリティー 1 コンテナ・オブジェクト、または単一カーディナリティー関係は、親ビジネス・オブジェクトの属性が単一の子ビジネス・オブジェクトを含む場合に生じます。この場合、子ビジネス・オブジェクトは単一のレコードのみを含むことができるコレクションを表します。属性タイプは子ビジネス・オブジェクトのものと同一です。

カーディナリティー n コンテナ・オブジェクト、または複数カーディナリティー関係は、親ビジネス・オブジェクトの属性に子ビジネス・オブジェクトの配列が含まれる場合に生じます。この場合、子ビジネス・オブジェクトは複数のレコードを含むことができるコレクションを表します。属性タイプは、子ビジネス・オブジェクトの配列の属性タイプと同じになります。

---

## Ariba Buyer のビジネス・オブジェクト

WebSphere Business Integration システム内のアプリケーション固有ビジネス・オブジェクトの構造は、Ariba Buyer によって設定され、Ariba Buyer メッセージ・オブジェクトの構造を反映します。Ariba Buyer は現在、以下の 2 タイプのメッセージ・オブジェクトをサポートしています。

- **要求 - 応答:** ERP アプリケーションへのデータの送信、または ERP アプリケーションのデータを要求する場合に使用
- **サブスクリプト:** ERP アプリケーションの非送信請求データの受信に使用

これらのメッセージ・タイプは、あるタイプのビジネス・オブジェクトから別のタイプのビジネス・オブジェクトに変換されます。ほとんどの Ariba Buyer イベントは要求 - 応答タイプです。

ビジネス・オブジェクトは、一致する Ariba Buyer メッセージ・オブジェクトから名前を継承します。Ariba バリエントと区画名、およびトピック名を使用して、継承した名前で構造上の各部分を結合します。例えば、SAP\_NA\_PurchaseOrderPush のようになります。

ビジネス・オブジェクト名の長さは 80 文字を超えることはできません。ビジネス・オブジェクト名が 80 文字よりも長い場合、ODA はエラー・メッセージを表示して、ビジネス・オブジェクト生成中に中断します。これを回避するには、バリエント名および区画名を短くしておいてください。

このエラーが発生した場合は、MessageConfiguration.table ファイル、および MessageDefinition.table ファイル中の TopicName の長さを短くし、ビジネス・オブジェクト定義を再生成してください。詳細については、33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。

## ビジネス・オブジェクト用の必要フィールド

ビジネス・オブジェクトには、次の 2 セットのフィールドが含まれます。

- すべてに共通する必要データ・フィールドのセット
- メッセージ・タイプによって異なるフィールドの追加セット

共通フィールドは 表 9 に、追加フィールドは 表 10 および 28 ページの表 11 に記載されています。

**注:** これら 2 つの表の追加フィールドは互いに排他的です。いずれか一方のみを使用し、両方を使用することはできません。

表 9. すべてのビジネス・オブジェクト用の必要フィールド

必要フィールド	タイプ	説明
Variant	ストリング	このメッセージを送信または受信するバリエントの名前。
Partition	ストリング	このメッセージを送信または受信する区画名。
TopicName	ストリング	イベントを一意的に識別。ビジネス・オブジェクト定義作成時のデフォルト・キー値。
ReplyKey	ストリング	メッセージを一意的に識別。非同期イベントのためにのみ統合チャネルにより生成。

表 10 には、**要求 - 応答** タイプのすべてのビジネス・オブジェクトに必要な追加フィールドのみがリストされています。

表 10. 要求 - 応答ビジネス・オブジェクト用の必要フィールド

必要フィールド	カーディナリティー	タイプ	説明
RequestSchema	1	MessageConfiguration.table 内の MergedSchemaName (あれば) または SchemaName によって指定	要求データ・タイプを指定
PullSchema_<n>	N	MessageDefinition.table 内で定義された各 Signature.Pull.Schema タイプごとに 1 つの PullSchema が存在	応答データ・タイプを指定

表 11 には、サブスクライブ・タイプすべてのビジネス・オブジェクトに必要な追加フィールドのみがリストされています。

表 11. サブスクライブ・ビジネス・オブジェクト用の必要フィールド

必要フィールド	カーディナリティー	説明
SubscribeSchema	N	MessageDefinition.table で定義された各スキーマのインバウンド・オブジェクト・タイプを指定する属性

## ビジネス・オブジェクト動詞

すべてのビジネス・オブジェクトは動詞を含んでおり、この動詞は、アプリケーション固有のビジネス・オブジェクト内のデータを受信側のアプリケーションが処理する方法を記述しています。Ariba Buyer は、動詞に類似しているが、Ariba Buyer データベースへの着信データのロード方法のみを記述するオペレーションのセットを定義します。

動詞およびオペレーションは次のように処理されます。

### • 同期要求 - 応答イベントの場合:

要求アプリケーション固有のビジネス・オブジェクト動詞は、MessageConfiguration.table のチャンネル・セクションで定義される Verb パラメーターの値に設定されます。この動詞は、宛先アプリケーションで意図されたアクションを記述します。応答アプリケーション固有のビジネス・オブジェクトに戻されるデータに対して行われるアクションは、MessageConfiguration.table のチャンネル・セクションで定義された Operation パラメーターの値に設定されます。このシーケンスは、図 2 で説明されています。

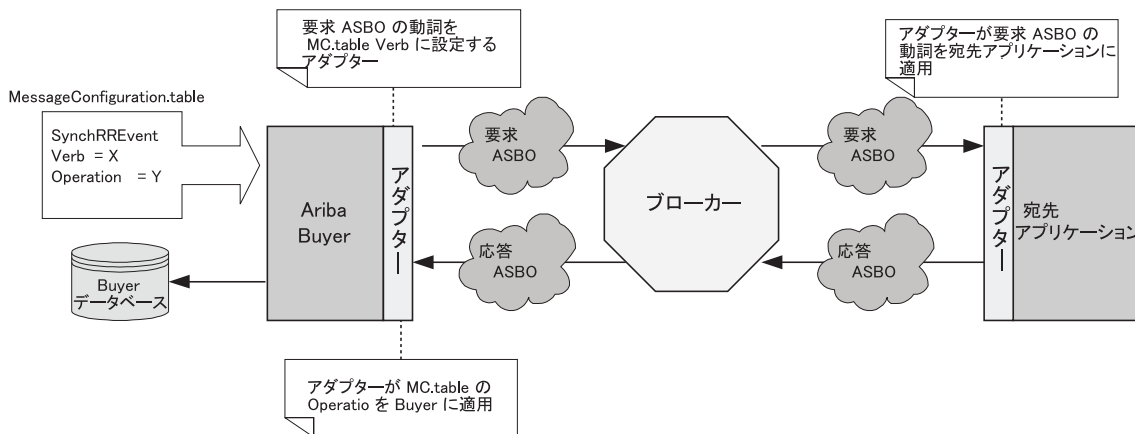


図 2. 同期要求 - 応答イベント

### • 非同期要求 - 応答イベントの場合:

要求アプリケーション固有のビジネス・オブジェクト動詞は、MessageConfiguration.table のチャンネル・セクションで定義される Verb パラメーターの値に設定されます。この動詞は、宛先アプリケーションで意図されたアクションを記述します。宛先アプリケーションは、応答アプリケーション固有のビジネス・オブジェクトの動詞を設定します。応答アプリケーション固有のビジネス・オブジェクトに戻されるデータに対して行われるアクションは、このアプリ

ケーション固有のビジネス・オブジェクト内の動詞を 表 12 で指定されたオペレーションの値に変換することにより決定されます。このシーケンスは、図 3 で説明されています。

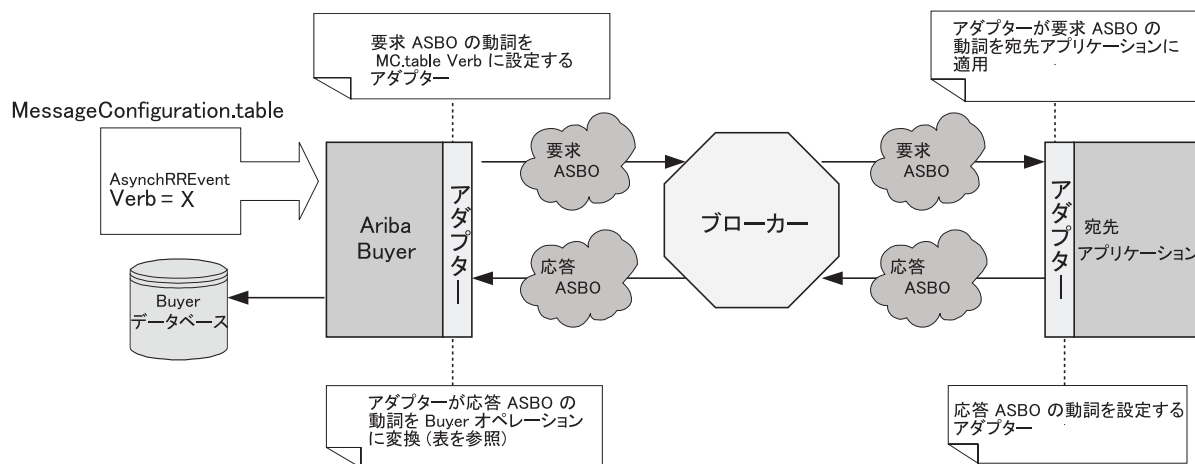


図 3. 非同期要求 - 応答イベント

・ サブスクライブ・イベントの場合:

ソース・アプリケーションはビジネス・オブジェクト動詞を設定します。受信されたデータに対して行われるアクションは、ビジネス・オブジェクト動詞を表 12 で指定されたオペレーションの値に変換することによって決定されます。このシーケンスは、図 4 で説明されています。

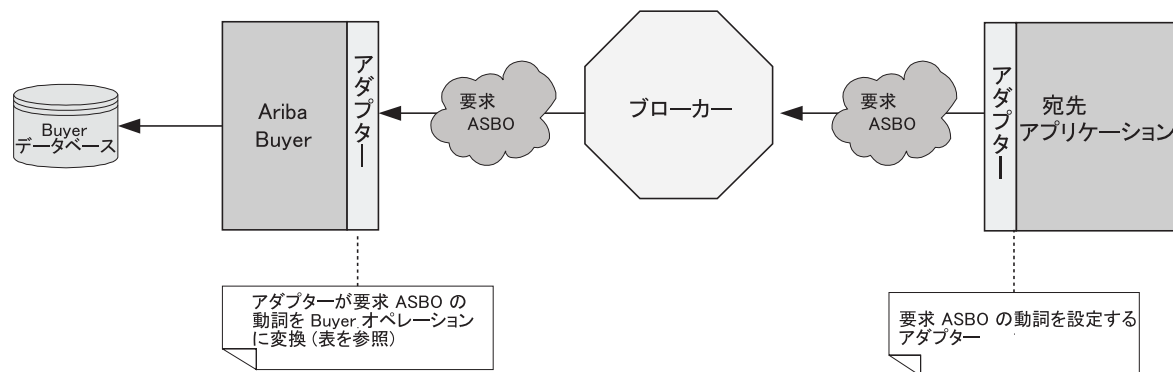


図 4. サブスクライブ・イベント

表 12 は、着信アプリケーション固有のビジネス・オブジェクト動詞と Ariba Buyer オペレーションとの間の相関を示しています。

表 12. ASBO 動詞と Ariba Buyer オペレーションの間の相関

動詞 (WBI)	オペレーション (Ariba Buyer)	処置
Create	Create	任意の新規オブジェクトを Ariba Buyer 内に作成します。
Delete	Delete	メッセージ内の任意のオブジェクトを Ariba Buyer から削除します。

表 12. ASBO 動詞と Ariba Buyer オペレーションの間の相関 (続き)

動詞 (WBI)	オペレーション (Ariba Buyer)	処置
Retrieve	N/A	Ariba Buyer は検索を実行できません。
Update	“Load And Delete”	既存オブジェクトの新規データ値を更新し、新規オブジェクトを作成します。また、着信メッセージに存在しないオブジェクトを Ariba Buyer から削除します。
ABUpdate	Update	既存オブジェクトの新規データ値を更新します。新規データの作成や、着信メッセージに存在しない Ariba Buyer 内のオブジェクトの削除は行いません。
Load	Load	既存オブジェクトの新規データ値を更新、および新規オブジェクトを作成します。
Update_And_Delete	“Update And Delete”	既存オブジェクトの新規データ値を更新します。オブジェクトが着信メッセージに存在しない場合には、そのオブジェクトを Ariba Buyer から削除します。
Update_Elements_Only	“Update Elements Only”	ベクトル・オブジェクトを更新します。

## 属性プロパティ

コネクタには、ビジネス・オブジェクト属性に設定可能な、各種のプロパティがあります。以下の表に属性とその値を示します。

表 13 には単純属性をリストしています。

表 13. 単純属性

属性	プロパティ
Name	ビジネス・オブジェクトのフィールド名を指定します。
Type	ビジネス・オブジェクトのフィールド・タイプを指定します。
MaxLength	デフォルトでは 255。
IsKey	各ビジネス・オブジェクトは、少なくとも 1 つのキー属性を持つ必要があります。キー属性は、属性のキー・プロパティを true に設定することで指定します。
IsForeignKey	使用しません。
Is Required	使用しません。
AppSpecInfo	オリジナル Java タイプを保持します。
DefaultValue	必要な属性が設定されていない場合に、コネクタがインバウンド・ビジネス・オブジェクト内の単純属性用に使用するデフォルト値を指定します。

表 14 には、子オブジェクト属性用のプロパティを示します。

表 14. 子オブジェクト属性

属性	説明
Name	ビジネス・オブジェクトのフィールド名を指定します。
Type	ビジネス・オブジェクトのフィールド・タイプを指定します。
Contained ObjectVersion	このプロパティは、子ビジネス・オブジェクトを表すすべての属性に、子のビジネス・オブジェクトのバージョン番号を指定します。
Relationship	子がコンテナ属性の場合、Containment に設定します。



表 14. 子オブジェクト属性 (続き)

属性	説明
IsKey	属性名が UniqueName である場合は true に設定し、それ以外の場合は false に設定します。
IsForeignKey	使用しません。
Is Required	使用しません。
AppSpecInfo	オリジナルの Ariba Buyer フィールド名を保持します。
Cardinality	タイプが配列またはベクトルを表す場合は N に設定し、それ以外の場合には 1 に設定します。

## アプリケーション固有の情報

アプリケーション固有の情報により、ビジネス・オブジェクトの処理方法に関する、アプリケーション依存の指示が、コネクターに提供されます。ビジネス・オブジェクト定義を拡張または変更する場合には、定義内のアプリケーション固有の情報と、コネクターが予期する構文とを必ず一致させる必要があります。

アプリケーション固有情報は、ビジネス・オブジェクトや、各ビジネス・オブジェクトの属性にも指定できます。

ビジネス・オブジェクトのアプリケーション固有の情報は、Adapter for Ariba Buyer の 2 つのキー・プロパティを処理します。これらのキー・プロパティは、次のとおりです。

- オリジナルのオブジェクト・タイプ名
- ビジネス・オブジェクトの使用するコラボレーションの名前

**注:** システム・インテグレーターは、コラボレーション名を記入する必要があります。この値は WebSphere MQ Integrator がブローカーの場合は使用されません。ただし IBM では、この値を WebSphere MQ Integrator Broker 内でこのトランザクションを処理するブローカー・フローの名前と一致させるよう設定することを推奨します。

ビジネス・オブジェクト属性のアプリケーション固有の情報は、オリジナルの Ariba Buyer 固有 Java タイプを保持します。

## ビジネス・オブジェクトの生成

Adapter for Ariba Buyer の統合チャネル・コンポーネントは、Ariba Buyer 自体で実行されます。イベントが統合チャネル用に構成されると、Ariba Buyer はチャネルを介してファイル・システム・リポジトリにアプリケーション固有のメタデータをエクスポートします。Object Discovery Agent は、このメタデータを使用して、統合ブローカーのリポジトリにエクスポートされるビジネス・オブジェクト定義を作成します。

Ariba Buyer は、実行時にイベントが発生するたびに、オブジェクト・レベルのデータとトランザクションのタイプに関する情報を含むメッセージ・オブジェクトを送信します。コネクターは、このデータを対応するビジネス・オブジェクト定義にマップして、アプリケーション固有のビジネス・オブジェクトを作成します。コネク

ターは、これらのビジネス・オブジェクトを統合ブローカーに送って処理します。また、統合ブローカーから送り返されたビジネス・オブジェクトを受信し、Ariba Buyer に戻します。

**注:** Ariba Buyer 内のオブジェクト・モデルが変更された場合は、ODA を使用して更新を反映するようにオリジナルのビジネス・オブジェクト定義を変更するか、新規の定義を作成します。統合ブローカー・リポジトリ内のビジネス・オブジェクト定義が、Ariba Buyer がエクスポートしたデータと正確に一致していない場合には、コネクタはビジネス・オブジェクトを作成できず、トランザクションは失敗します。

Business Object Designer の提供するグラフィカル・インターフェースを使用すると、実行時に使用するビジネス・オブジェクト定義を生成できます。詳細については、33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。



---

## 第 5 章 ビジネス・オブジェクト定義の生成

この章では、Object Discovery Agent (ODA) for Ariba Buyer の詳細と、それを使用して IBM WebSphere Business Integration Adapter for Ariba Buyer のビジネス・オブジェクト定義を生成する方法について説明します。

本章の内容は、次のとおりです。

- 33 ページの『ODA for Ariba Buyer の概要』
- 33 ページの『ビジネス・オブジェクト定義の生成』
- 42 ページの『ファイルのアップロード』

---

### ODA for Ariba Buyer の概要

ODA (Object Discovery Agent) を使用すると、アプリケーション固有のビジネス・オブジェクト定義を生成できます。ビジネス・オブジェクト定義はビジネス・オブジェクトのテンプレートです。ODA は指定されたアプリケーション・オブジェクトを検査し、ビジネス・オブジェクト属性に対応するそれらのオブジェクトのエレメントを「発見」し、その情報を表現するビジネス・オブジェクト定義を生成します。Business Object Designer を使用すると、グラフィカル・インターフェースを使用して Object Discovery Agent にアクセスし、対話式に操作できます。

Object Discovery Agent (ODA) for Ariba Buyer は、初期化時に Ariba Buyer によってエクスポートされたメタデータから、ビジネス・オブジェクト定義を生成します。Business Object Designer ウィザードにより、これらの定義の生成プロセスが自動化されます。ビジネス・オブジェクト定義はサーバーに保管する前に表示または修正できます。

ビジネス・オブジェクト定義を生成するために、ODA を次の 2 つの段階で使用できます。

1. **システムを初めてセットアップする時:** 統合チャネル上で定義されたすべての統合イベントに対して、ビジネス・オブジェクト定義のセットを作成する必要があります。ビジネス・オブジェクト定義およびブローカー・コラボレーションおよびワークフローを、データベース初期化の一部として実行されるすべての統合チャネル・イベントに対して定義する必要があります。
2. **Ariba Buyer 内で、統合チャネル・イベントで使用するオブジェクトを変更する時:** Business Object Designer で ODA を実行し、そのオブジェクトの変更されたメタデータをピックアップして、新しいビジネス・オブジェクト定義をブローカー・リポジトリにエクスポートする必要があります。

---

### ビジネス・オブジェクト定義の生成

このセクションでは、Business Object Designer の Ariba Buyer ODA を使用して、ビジネス・オブジェクト定義を生成する方法について説明します。Business Object Designer の起動と使用については、「*IBM WebSphere Business Integration Adapters* ビジネス・オブジェクト開発ガイド」を参照してください。

## ODA の始動

ODA はメタデータ・リポジトリがあるファイル・システムをマウントできる任意のマシンから、start\_BuyerODA.bat (WIN) または start\_BuyerODA.sh (UNIX) 開始ファイルを使用して実行できます。このファイルには、Ariba Buyer とコネクタに必須の .jar ファイルを参照するパスを含む、開始パラメーターが含まれます。これらの .jar ファイルにも ODA を実行しているマシンからアクセス可能でなければなりません。

ODA for Ariba Buyer には BuyerODA というデフォルト名が付けられています。この名前は、開始スクリプト内の AGENTNAME 変数の値を変えることで変更できます。

ODA を開始するには、次のコマンドを実行します。

```
start_BuyerODA <Ariba server root path>
```

例えば、次のようになります。

```
start_BuyerODA C:%Ariba%app%Buyer%Server
```

## Business Object Designer の実行

Business Object Designer は、ODA を使用してビジネス・オブジェクト定義を生成するステップを、段階的にガイドするウィザードを提供します。以下のステップがあります。

### エージェントの選択

1. Business Object Designer を開始します。
2. 「ファイル」>「**ODA を使用して新規作成**」を選択します。「ビジネス・オブジェクト・ウィザード - ステップ 1/6 - エージェントの選択」画面が表示されます。
3. 「検索されたエージェント」リストで (start\_BuyerODA スクリプトの) ODA/AGENTNAME を選択し、「次へ」をクリックします。

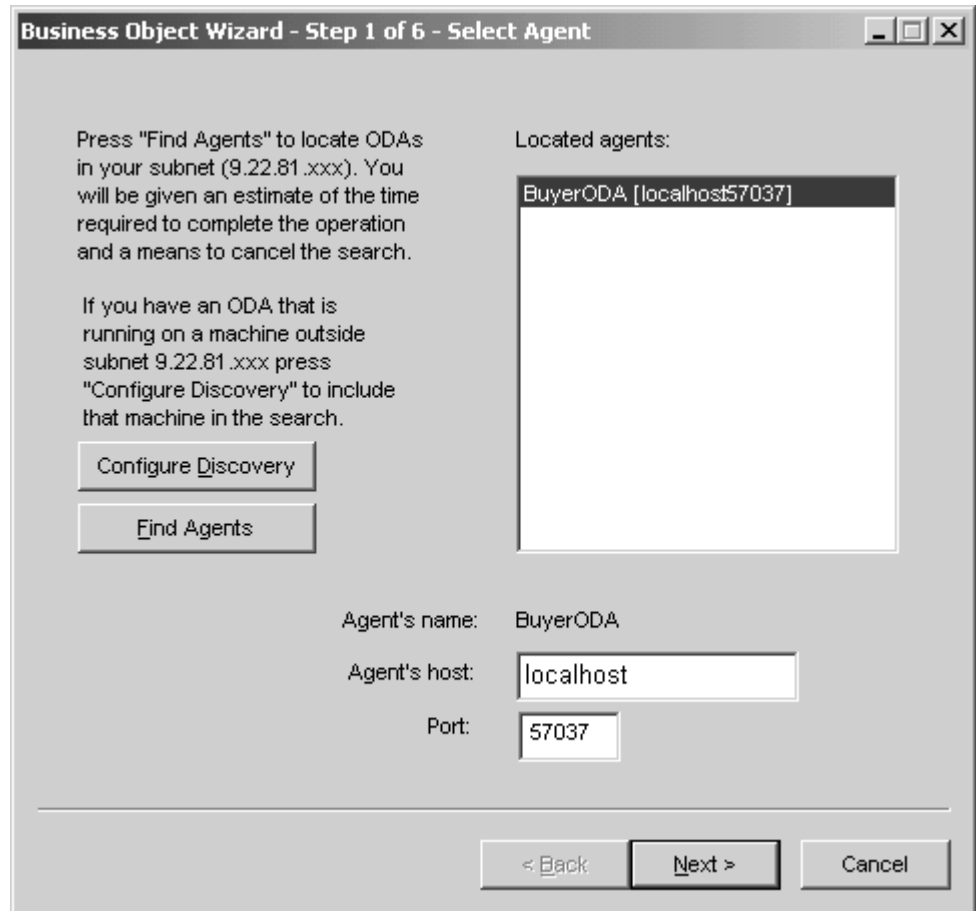


図 5. 「エージェントの選択」画面

## エージェントの構成

「ビジネス・オブジェクト・ウィザード - ステップ 2/6 - エージェントの構成」画面が表示されます。この画面は、最初に開いた時には、内容が空白になっています。

この画面で、次のことを行います。

- ODA がメタデータ・リポジトリとメッセージ・ファイルを見つけるために必要とするパス情報の入力。
- トレース・メッセージ・ファイルの指定 (例えば、BuyerODATrace.txt)。
- トレース・レベルの指定 (0 から 5)。

**注:** メッセージ・ファイルの名前が正しく指定されていない場合には、ODA はメッセージなしで実行されます。

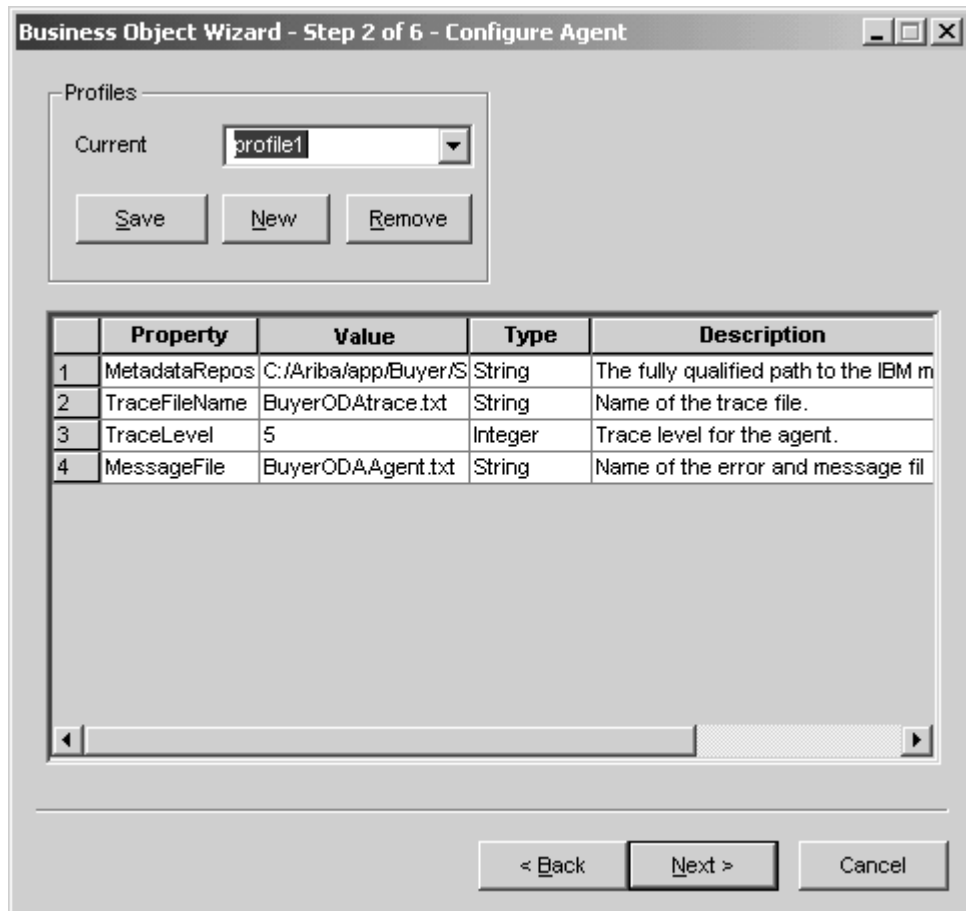


図6. 「エージェントの構成」画面

Buyer ODA によって出力されるメッセージはすべて BuyerODAAgent.txt ファイルに標準メッセージ・ファイル・フォーマットで表示されます。トレース・レベルの範囲とその意味の詳細については、22 ページの『ログ・ファイルとトレース・ファイルの使用』を参照してください。

この画面に入力した値すべてをプロファイルに保管することもできます。ODA を次回に実行する際に、ドロップダウン・メニューからプロファイルを選択して保管しておいた値を使用すれば、すべてのデータを再入力する必要はありません。

それぞれが異なる指定値のセットを含む、複数のプロファイルを保管することができます。

1. ODA の初回実行時には、「プロファイル」メニューを使用して新規プロファイルを作成します。2 回目以降に使用する際には、既存のプロファイルを選択できます。
2. 各プロパティの名前、値、タイプおよび説明を入力します。

**注:** プロファイルを使用すると、プロパティ値が入力されます。プロパティと、各プロパティに指定可能な値を、37 ページの表 15 にリストしています。

表 15. ODA 構成用のプロパティ値

番号	プロパティ名	プロパティ・タイプ	説明	必要
1	MetadataRepos	ストリング	メタデータ・ファイル・リポジトリが含まれるディレクトリー。例えば、C:\temp\metadata。	はい
2	TraceFileName	ストリング	トレース・ファイルの名前	いいえ
3	TraceLevel	整数	ODA のトレース・レベル	はい
4	MessageFile	ストリング	メッセージ・ファイルを参照するパス	はい

**注:** メタデータ・リポジトリは <Ariba\_server\_root>/config/Parameters.table 内の、MetadataRepos プロパティ用として指定されたロケーションに常に存在します。MetadataRepos は、この値に設定することも、%metadata ディレクトリーを他のマシンにコピーすることで MetadataRepos 値をリセットすることもできます。これは、ビジネス・オブジェクト定義生成プロセス中には ODA とアプリケーション間に対話がないために可能になります。

入力が完了したら、「次へ」をクリックします。

### ビジネス・オブジェクトの選択

「ビジネス・オブジェクト・ウィザード - ステップ 3/6 - ソースの選択」画面が表示されます。この画面を使用して、ODA がビジネス・オブジェクト定義を生成する MessageConfigurations を任意の数選択します。

画面には Ariba Buyer 内に定義済みのバリエーション (メタデータ・リポジトリの構造から派生) がリストされます。Ariba Buyer が初期化されていない場合には、リストは空になります。

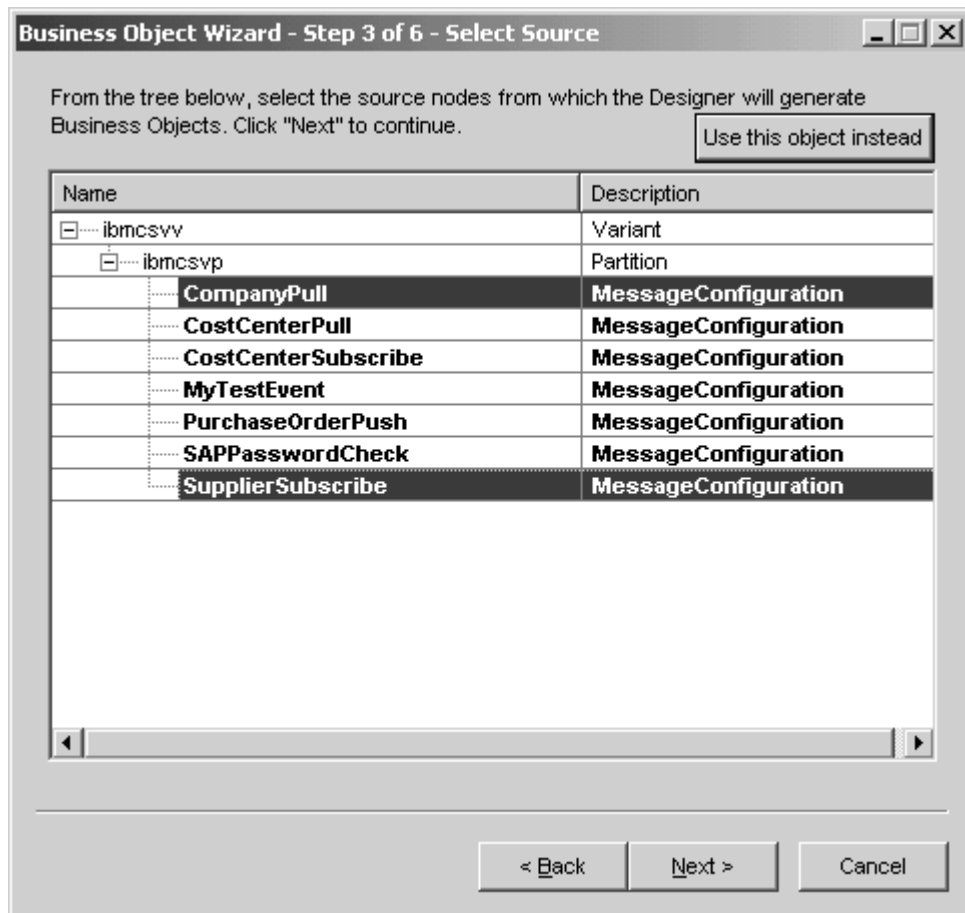


図7. 「ソースの選択」画面

- バリエントを展開して、その区画のリストを表示します。
  - 区画を展開して MessageConfigurations のリストを表示します。
1. 使用する MessageConfigurations を選択します。
  2. 「次へ」をクリックします。

### 選択の確認

「ビジネス・オブジェクト・ウィザード - ステップ 4/6 - ビジネス・オブジェクトのソース・ノードの確認」画面が表示されます。この画面には、選択した MessageConfigurations が表示されています。

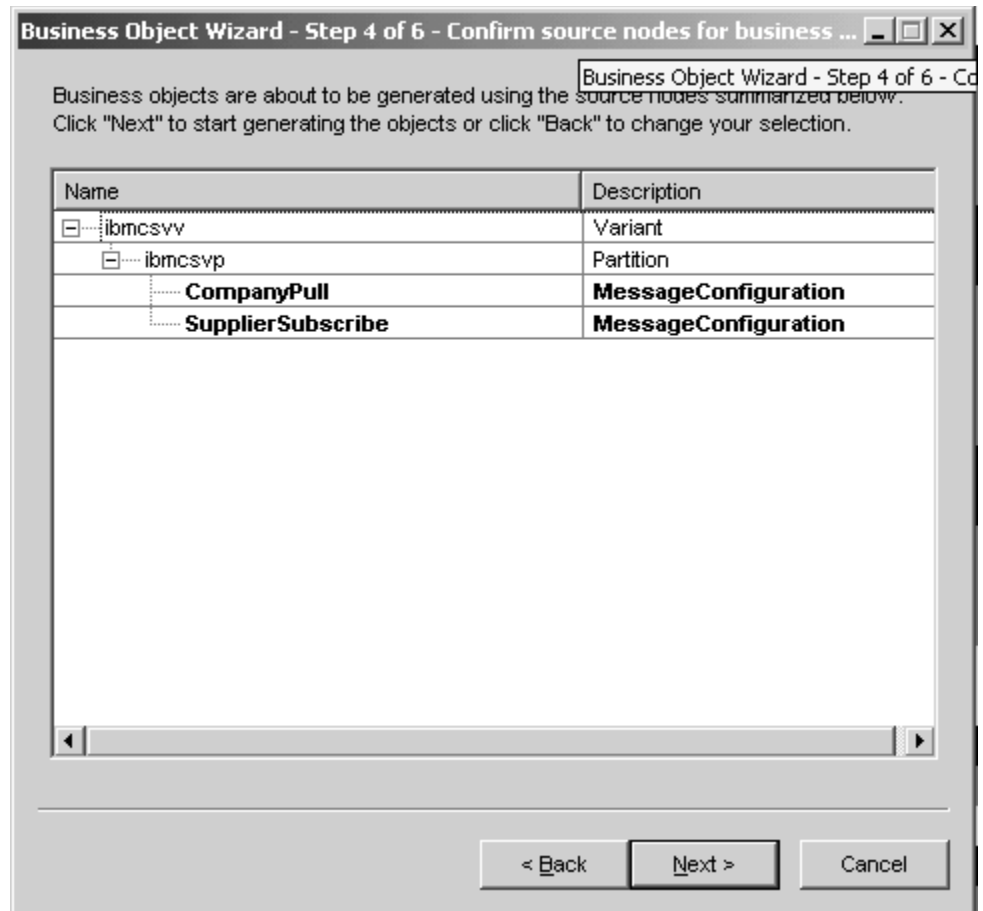


図 8. 「ソース・ノードの確認」画面

「戻る」をクリックして変更するか、または「次へ」をクリックしてリストが正しいことを確認します。

### ビジネス・オブジェクトの生成

「ビジネス・オブジェクト・ウィザード - ステップ 5/6 - ビジネス・オブジェクトの生成中...」画面が表示されます。

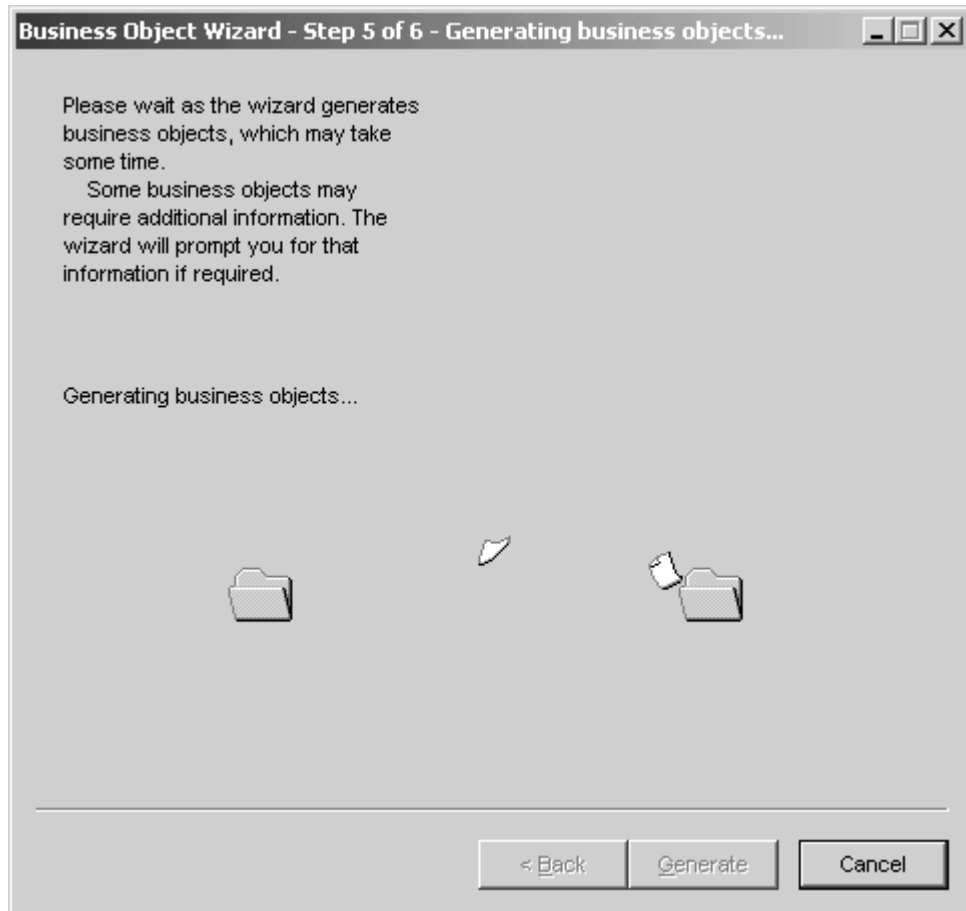


図9. 「ビジネス・オブジェクトの生成中」画面

ウィザードがビジネス・オブジェクトを生成中であることを示すメッセージが表示されます。

### ビジネス・オブジェクトのコピーを別ファイルに保管

「ビジネス・オブジェクト・ウィザード - ステップ 6/6 - ビジネス・オブジェクトの保管」画面が表示されます。



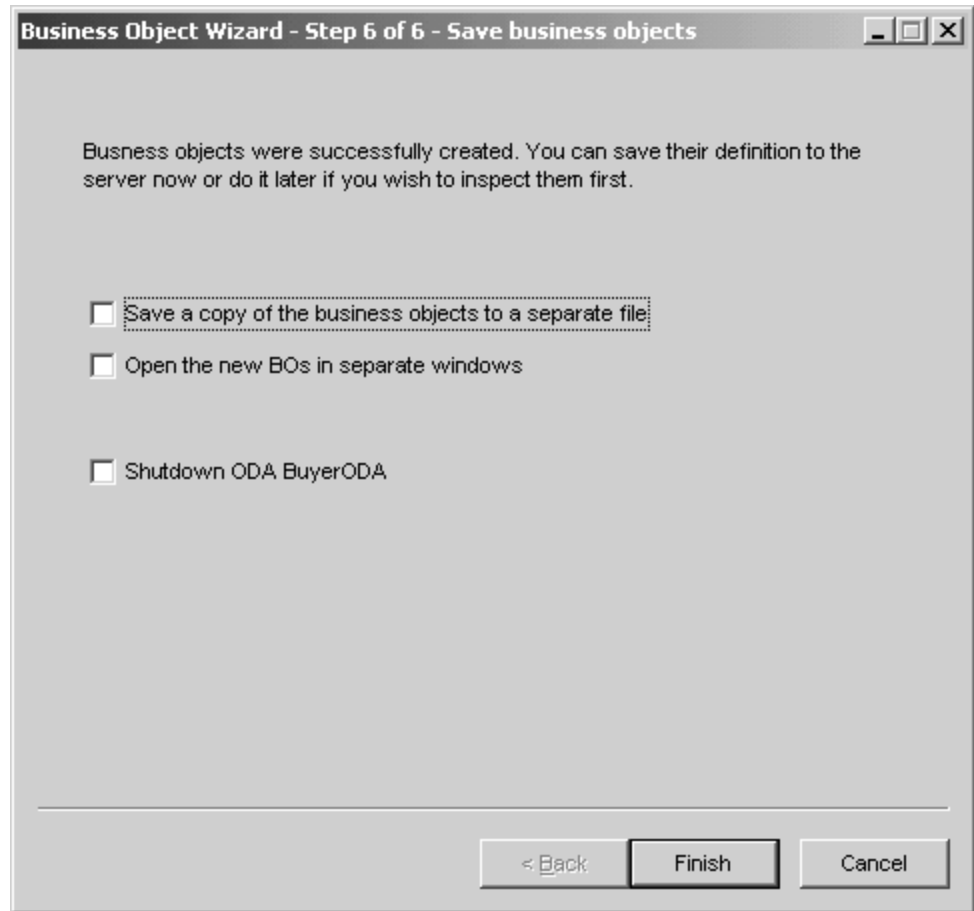


図 10. 「ビジネス・オブジェクトの保管」画面

生成されたビジネス・オブジェクト定義をオプションでファイルに保存できます。これを行うには、以下のステップを実行します。

1. 「ビジネス・オブジェクトをファイルに保管」をチェックします。ダイアログ・ボックスが表示されます。
2. 新規ビジネス・オブジェクト定義のコピーの保管先のロケーションを入力します。

Business Object Designer は、ファイルを指定されたロケーションに保管します。

**注:** Ariba Buyer 内で新規のカスタム・オブジェクトを作成した場合には、本章で説明されているように、ODA を使用してそのオブジェクトに新規のビジネス・オブジェクト定義を追加する必要があります。

ODA での作業を終了したら、「終了」をクリックする前に、「ODA Buyer ODA をシャットダウンする (Shutdown ODA Buyer ODA)」をチェックして、ODA をシャットダウンすることができます。

---

## ファイルのアップロード

新規に作成したビジネス・オブジェクト定義ファイルは、作成が完了したら統合ブローカーにアップロードする必要があります。アップロード・プロセスは、実行している統合ブローカー (ICS、MQ Integrator Broker、または WAS) によって異なります。

ICS の場合: ビジネス・オブジェクト定義ファイルをローカル・マシンに保管しており、サーバーのリポジトリにアップロードする必要がある場合は、「*WebSphere InterChange Server インプリメンテーション・ガイド*」を参照してください。

WebSphere MQ Integrator Broker の場合: Business Object Designer から統合ブローカーにビジネス・オブジェクト定義をエクスポートする必要があります。詳細については、「*WebSphere MQ Integrator Broker 用インプリメンテーション・ガイド*」を参照してください。

WAS の場合: Business Object Designer から統合ブローカーにビジネス・オブジェクト定義をエクスポートする必要があります。詳細については、「*アダプター実装ガイド (WebSphere Application Server)*」を参照してください。

## 第 6 章 トラブルシューティングとエラー処理

本章では、Adapter for Ariba Buyer によるエラー処理の方法について説明します。このアダプターの 3 つのコンポーネントは、ロギング・メッセージとトレース・メッセージを生成します。本章では、コンポーネントごとにこれらのメッセージを説明します。以下のセクションから構成されています。

- 43 ページの『チャンネル・エラー処理』
- 49 ページの『トレース・メッセージ』
- 50 ページの『トラブルシューティングのヒント』

### チャンネル・エラー処理

統合チャンネルは Ariba Buyer API を使用してエラー・メッセージをログに記録します。統合チャンネルは、トレース・レベルにかかわらず、処理中に発生した異常条件をすべてログに記録します。<Ariba server root>%logs にある、AribaBuyer<nodename>Log.txt (例えば、AribaBuyerNode1Log.txt) という名前の Ariba Buyer ログ・ファイルにテキストを書き込みます。

メッセージは、条件と結果の詳細記述を含み、ビジネス・オブジェクト・ダンプや(例外的) スタック・トレースなど、デバッグに役立つ可能性のある追加の情報を含む場合もあります。

チャンネル・エラー・メッセージには次の 2 つのタイプがあります。

- 初期化エラー
- 実行時エラー

チャンネル初期化エラーは通常致命的であり、Ariba Buyer ノードの開始を妨げます。表 16 には、起こりうるエラーと各エラー処理のために取りうる処置をリストしています。

表 16. チャンネル初期化エラー

エラー・メッセージ	説明	修正処置
Error accessing RMI registry on port xxx	致命的エラー。ChannelRemote が開始できない場合、コネクタはチャンネルと通信できません。	Parameters.table に指定された ChannelPort パラメーターを検査し、必要に応じて変更します。
Error binding ChannelRemote object at registry: <url> Exception: <message text>	致命的エラー。ChannelRemote が開始できない場合、コネクタはチャンネルと通信できません。	Parameters.table に指定された ChannelPort パラメーターを検査します。詳しくは、例外メッセージを参照してください。
Error initializing ChannelRemote object. Exception: <xxx>	統合チャンネルのリモート・オブジェクトの開始時にエラーが発生しました。	Parameters.table に指定された ChannelPort パラメーターを検査します。詳しくは、例外メッセージを参照してください。

表 16. チャネル初期化エラー (続き)

エラー・メッセージ	説明	修正処置
Error creating file: xxx	指定されたディレクトリーまたはファイルを作成しようとする、チャネルがエラーを受信します。	以下を検証してください。 <ul style="list-style-type: none"> <li>• ファイルを保管するのに十分なディスク・スペースがある。</li> <li>• ユーザーがファイルとディレクトリーを作成する権限を持っている。</li> </ul>

表 17 には、起こりうる実行時エラーと各エラー処理のために取りうる処置をリストしています。

表 17. チャネル実行時エラー

エラー・メッセージ	説明	修正処置
Exception processing Message: <topic name> Exception <error text>	メッセージ・ハンドラーが、メッセージをコネクターに送信しようとする、エラーを受信します。	詳しくは、エラー・テキストを参照してください。
The specified reply key was not found in the table. Key = xxx	メッセージ・ハンドラーは、応答表内に検出されなかったキーについてのメッセージを受信します。	指定された応答キーが別のノードで検出され、処理されたかどうかをログ・ファイルでチェックします。応答キーが別のノードになかった場合は、イベントを戻します。ノードが停止またはシャットダウンした場合、応答キーは無効になります。
Received Exception connecting to Connector remote object. Exception <error text>	メッセージ・ハンドラーのコネクターへの参照が欠落しているか、または無効です。	以下を検証してください。 <ul style="list-style-type: none"> <li>• Parameters.table 内の ConnectorPort パラメーターと ConnectorURL パラメーターが正しい。</li> <li>• Ariba Buyer コネクターが開始されている。</li> </ul>
The xxx parameter is not specified.	指定されたパラメーターがメッセージ構成またはメッセージ定義で検出されません。	メッセージ構成またはメッセージ定義を適切に編集します。
xxx has a null value	コネクターから受信した応答のパラメーターがヌル値です。	このパラメーターが、マップ、アダプター、またはコラボレーションによって正しく設定されていることを検証します。
Message Configuration not found for <topic name> in partition: <partition name>	指定された区画内で指定されたトピック名に対するメッセージ構成項目が検出されません。	指定された区画が、統合チャネル・ディレクトリー内の MessageConfiguration.table に存在することを検証します。
Channel parameter <xxx> not set	指定されたパラメーターが Parameters.table ファイルの統合チャネル・セクションに設定されていません。	パラメーターを正しい値で Parameters.table に追加します。
Processing for the Message with TopicName <xxx> has exceeded the timeout interval specified	指定されたトピック名を持つメッセージの応答が、割り当てられた時間内に受信されませんでした。	指定されたトピック名のタイムアウト・パラメーターが、MessageConfiguration.txt に正しく設定されていることを検証します。関連エラー・メッセージの統合チャネル・ファイルとコネクター・ログ・ファイルをチェックします。

表 17. チャネル実行時エラー (続き)

エラー・メッセージ	説明	修正処置
ABRequestMessage Handler not found in table for: <partition - topic name>	指定されたトピック名と区画を持つメッセージに用のメッセージ・ハンドラーが検出できません。	MessageConfiguration.table に指定されたトピック名および区画に関するエントリを追加します。
Reply for Message with TopicName <xxx> has no data for defined PullSchema objects	統合チャネルが、指定されたメッセージに関する欠落データのある応答を受信しました。	以下を検証してください。 <ul style="list-style-type: none"> <li>• PullSchema が MessageDefinition.table に定義されている。</li> <li>• PullSchema パラメーターが、マップ、コラボレーション、またはアダプターによって正しく設定されている。</li> </ul>

## コネクター・エラー処理

コネクターは、トレース・レベルにかかわらず、処理中に発生した異常条件をすべてログに記録します。コネクターは、コネクター・ログ・ファイルにテキストを書き込みます。

メッセージは、条件と結果の詳細記述を含み、ビジネス・オブジェクト・ダンプや(例外的) スタック・トレースなど、デバッグに役立つ可能性のある追加の情報を含む場合もあります。

コネクター・エラー・メッセージには次の 2 つのタイプがあります。

- 初期化エラー
- 実行時エラー

表 18 には、起こりうる初期化エラーと各エラー処理のために取りうる処置をリストしています。

表 18. コネクター初期化エラー

エラー・メッセージ	説明	修正処置
An Exception occurred attempting to bind the ABConnectorRemote object to the RMI Registry using name <name> Exception: <error text>	致命的エラー。ConnectorRemote が開始できない場合、チャネルはコネクターと通信できません。	コネクター・アプリケーション構成プロパティの ConnectorPort プロパティを正しく設定します。詳しくは、エラー・テキストを参照してください。
Could not connect to the remote object of the ABChannel	コネクターが初期化時にチャネル・ノードに接続できません。	以下を検証してください。 <ul style="list-style-type: none"> <li>• Ariba Buyer サーバーが開始されている。</li> <li>• コネクター・アプリケーション構成プロパティ・ノードが正しく設定されている。</li> </ul>
Error connecting to remote object <node name>, using URL: <url>	コネクターが、指定された Ariba Buyer ノード上のチャネルのリモート・オブジェクトに接続できません。	コネクター・アプリケーション構成プロパティのノード・プロパティを正しく設定します。

表 18. コネクター初期化エラー (続き)

エラー・メッセージ	説明	修正処置
Connector property not specified: <name>	指定されたプロパティが設定されていません。	コネクター・アプリケーション構成プロパティに指定されたプロパティを正しく設定します。

表 19 には、起こりうる実行時エラーと各エラー処理のために取りうる処置をリストしています。

表 19. コネクター実行時エラー

エラー・メッセージ	説明	修正処置
Collaboration name not found in Business Object: <name>	コネクターが、コラボレーション・アプリケーション固有のプロパティ・セットを持たないビジネス・オブジェクトを送信しようとしています。	指定されたビジネス・オブジェクト定義内のビジネス・オブジェクト・レベルのアプリケーション固有の情報に、コラボレーション・プロパティを正しく設定します。
Connector failed to deliver business object: <BO name>	gotAppEvent がエラーを戻します。	詳しくはログ・ファイルをチェックしてください。
Failed to deliver business object: <BO name> because the connector is not active.	gotAppEvent がエラーを戻します。ビジネス・オブジェクトを非アクティブなコネクターに配信しようとしたため。	コネクターがアクティブになっていることを検証します。
Connector failed to deliver business object: <BO name> because no subscription was found.	gotAppEvent がエラーを戻します。	ビジネス・オブジェクトがコラボレーションによってサブスクライブされていることを検証します。
Invalid verb specified: <verb> for business object: <BO name>	コネクターに送信されたメッセージ中の動詞が、ビジネス・オブジェクトによってサポートされていません。	以下を検証してください。 <ul style="list-style-type: none"> <li>MessageConfiguration.table で動詞パラメーターが正しく設定されている。</li> <li>ビジネス・オブジェクト定義で、サポートされているすべての動詞が正しく設定されている。</li> </ul>
The attribute <xxx> was not found in the Business Object <BO name>	コネクターが、ビジネス・オブジェクト定義内に対応する属性がないフィールドを含むメッセージを処理しています。	ビジネス・オブジェクト定義が正しくセットアップされていることを検証します。
The Buyer Connector does not poll. Set the Connector property PollFrequency to "No" to avoid this message.	Ariba Buyer コネクターがポーリングしません。	コネクター標準プロパティ PollFrequency を「いいえ」に設定します。
The business object <name> is not defined	指定された名前を持つビジネス・オブジェクト定義が検出できません。	ODA を使用して、指定されたビジネス・オブジェクトを作成します。
The value specified for attribute <name> is not a valid type	メッセージ・オブジェクトに指定された属性が、ビジネス・オブジェクト定義の属性と同じタイプではありません。	ビジネス・オブジェクト定義を検証し、必要に応じて ODA を使用して再作成します。

表 19. コネクター実行時エラー (続き)

エラー・メッセージ	説明	修正処置
The attribute: <name> was not found in the business object definition: <BO name>	メッセージ・オブジェクトで指定された属性が、指定されたビジネス・オブジェクト定義にありません。	ビジネス・オブジェクト定義を検証し、必要に応じて ODA を使用して再作成します。
An error occurred during the processing of the business object: <BO name>. Exception: <error text>	コネクターが指定されたビジネス・オブジェクトを送信した後にエラーが発生しました。	詳しくはエラー・テキストをチェックしてください。
An exception occurred unbinding the remote object from the RMI Registry.Name <name> Exception: <error text>	レジストリーからコネクターのリモート・オブジェクトをアンバインドする時に問題がありました。これは、次回のコネクター開始時に、コネクターによる RMI レジストリーへのリモート・オブジェクトのバインドに支障がある可能性があります。	コネクター・アプリケーション構成プロパティの ConnectorPort プロパティを検証します。詳しくはエラー・テキストをチェックしてください。コネクターを開始する前に、システムを再始動しなければならない場合があります。
A SpecNameNotFound exception occurred when adding the child business object: <child BO name> to the business object: <parent BO name>	コネクターが、指定された子ビジネス・オブジェクトを、指定された親ビジネス・オブジェクトの n- カーディナリティー属性に追加しようとしています。	カーディナリティーが親ビジネス・オブジェクトに正しく設定されていることを検証します。
Exception occurred instantiating class for type: <name>. Exception: <error text>	属性 App Spec Info に指定されたクラスがインスタンス化できませんでした。	指定されたクラスがクラスパスにあることを検証します。詳しくは、エラー・テキストを参照してください。
No String constructor found for type: <class name>	属性 App Spec Info に指定されたクラスに、特定のストリング・パラメーターを受け入れるコンストラクターがありません。	ストリング・コンストラクターがない場合は、メタデータで指定されたクラスを使用できません。
An error occurred on sending object to Channel on node: <name>. Exception: <error text>	オブジェクトのチャンネルへの送信中に、指定されたエラーが発生しました。	詳しくは、エラー・テキストを参照してください。Ariba Buyer ログで詳細をチェックしてください。
Reply key: <key> for object not found in Channel on Node: <node name>	コネクターが、チャンネルの表にない応答キーでオブジェクトをチャンネルに送信しました。	応答キーが無効であるか、別のノード上に存在している可能性があります。ログをチェックして、別のノードがそのオブジェクトを処理したかどうかを調べてください。要求がタイムアウトになり、応答キーが表から除去された可能性があります。検証のためログをチェックしてください。
An error occurred on sending the business object: <BO name> to Channel. Check the logs for more information.	指定されたビジネス・オブジェクトをチャンネルに送信中にエラーが発生しました。	詳しくは、ログをチェックしてください。



表 19. コネクター実行時エラー (続き)

エラー・メッセージ	説明	修正処置
An error occurred on creating a java.util.Date object from the String "xxx". Locale: <locale>	日付ストリングが、デフォルトのフォーマット、またはコネクター・アプリケーション構成プロパティの DateFormats パラメーターで指定されたフォーマットと一致しません。	コネクター・アプリケーション構成プロパティで正しいフォーマット・ストリングを指定します。日付フォーマット・ストリングの指定については、java.text.SimpleDateFormat の API を参照してください。
The business object for TopicName: <topic name>, variant: <variant name>, partition: <partition name> is not defined	指定されたトピック名、バリエーションおよび区画に一致するビジネス・オブジェクト定義が検出できませんでした。	ODA を使用して、一致するビジネス・オブジェクト定義を作成します。 VariantMappings コネクター・アプリケーション構成プロパティで、ビジネス・オブジェクトの接頭部を指定します。
Variant prefix not specified for variant: <variant name>	指定されたバリエーションの接頭部が VariantMappings コネクター・アプリケーション構成プロパティで検出されませんでした。	VariantMappings コネクター・アプリケーション構成プロパティで、ビジネス・オブジェクトの接頭部を指定します。

## ODA エラー処理

ODA は ODA ログにエラー・メッセージのログを記録します。ODA は、トレース・レベルにかかわらず、処理中に発生した異常状態をすべてログに記録します。

メッセージは、条件と結果の詳細記述を含み、ビジネス・オブジェクト・ダンプや(例外的) スタック・トレースなど、デバッグに役立つ可能性のある追加の情報を含む場合もあります。

表 20 には、起こりうるエラーと、各エラー処理のために取りうる処置をリストしています。

表 20. ODA エラー

エラー・メッセージ	説明	修正処置
The specified path does not exist: <path name>	コネクター ODA プロパティ BuyerPath が設定されていないか、無効です。	BuyerPath プロパティが正しく設定されていることを検証します。
File not found: <file name>	指定されたメタデータ・ファイルが見つかりません。	BuyerPath プロパティが正しく設定されていることを検証します。 Ariba Buyer からメタデータ・ファイルを再生成します。
Class not found for file: <file name>	ABClassMeta クラスが検出できないか、メタデータ・ファイルがバックレベル ABClassMeta クラスで作成されました。	ABClassMeta クラスがクラスパス内にあることを検証します。メタデータが最新の ABClassMeta クラスで作成されたことを検証します。
IO Error reading file: <file name>	メタデータ・ファイルの読み取り時に、ODA がエラーを受信します。	ユーザーがファイルを開く権限を持っていることを検証します。ファイルが使用中ではないことを検証します。



表 20. ODA エラー (続き)

エラー・メッセージ	説明	修正処置
The ABClassMeta is null for class name: <class name> variant name: <variant name>	指定されたクラスとバリエーションのメタデータ・ファイルが見つかりませんでした。	Ariba Buyer からメタデータ・ファイルを再生成します。
Error creating BusObjAttr for <field name> in class <class name>	指定されたクラス用のビジネス・オブジェクト定義に、指定された属性を追加中にエラーが発生しました。	以下を検証してください。 <ul style="list-style-type: none"> <li>フィールド・タイプがビジネス・オブジェクトによってサポートされている。</li> <li>ビジネス・オブジェクト内で属性名が重複していない。</li> </ul>
Error creating Business Object definition: <BO name>. Error Text: <error text>	ODA がビジネス・オブジェクト定義の作成時にエラーを受信します。	詳しくは、エラー・テキストを参照してください。

## トレース・メッセージ

トレースは、コネクタの振る舞いを詳細にトレースするためにオンにできるオプションのデバッグ機能です。トレース・メッセージは構成可能であり、動的変更が可能です。要求の詳細に応じてさまざまなレベルを設定します。トレース・メッセージは、デフォルトでは「STDOUT」(画面) に書き込まれます。また、トレースをファイルに書き込むように構成することもできます。

**推奨：**トレースは、パフォーマンスを向上させ、ファイル・サイズを小さくするために、実動システム上ではオフにするか、できるだけ低レベルに設定しておいてください。

表 21 では、Ariba Buyer コネクタが各トレース・レベルで出力するトレース・メッセージのタイプを説明しています。すべてのトレース・メッセージがコネクタ・プロパティ `TraceFileName` によって指定されたファイルに表示されます。これらのメッセージは、IBM WebSphere Business Integration Adapter アーキテクチャによって出力されるトレース・メッセージに追加されます。

表 21. トレース・メッセージ

トレース・レベル	トレース・メッセージ
レベル 0	コネクタのバージョンを識別するメッセージ。このレベルでは他のトレースは行われません。このメッセージは、常時表示されます。例えば、次のとおりです。 <code>'2003/03/21 15:01:46.812: This is version 3.0 of the Adapter for Ariba Buyer'</code>
レベル 1	<code>pollForEvents</code> メソッドが実行される度に配信されるメッセージ。
レベル 2	<ul style="list-style-type: none"> <li><code>gotAppEvent</code> から統合ブローカーにビジネス・オブジェクトが通知される度にログに記録されるメッセージ。</li> <li>ビジネス・オブジェクト要求を受信する度に表示されるメッセージ。</li> </ul>
レベル 3	該当なし。

表 21. トレース・メッセージ (続き)

トレース・レベル	トレース・メッセージ
レベル 4	<ul style="list-style-type: none"> <li>アプリケーション固有の通知メッセージ。例えば、ビジネス・オブジェクトのアプリケーション固有の通知フィールドを構文解析する関数によって戻される値を示すメッセージ。</li> <li>コネクターが、コネクターのプロセス・フローのトレースに役立つ関数を開始または終了したかどうかを識別するメッセージ。</li> </ul>
レベル 5	<ul style="list-style-type: none"> <li>コネクターの初期化を示すメッセージ。例えば、統合ブローカーから検索された各構成プロパティの値を示すメッセージ。</li> <li>ビジネス・オブジェクト・ダンプから構成されるメッセージ。このトレース・レベルでは、コネクターは、(コネクターがコラボレーションから受信するオブジェクトを示す) オブジェクトの処理を開始する前、および (コネクターがコラボレーションに戻すオブジェクトを示す) オブジェクトの処理を終了した後に、ビジネス・オブジェクトのテキスト表現を出力します。</li> </ul>

## トラブルシューティングのヒント

トラブルシューティングの問題には次のヒントを利用してください。

- チャンネルがリモートで実行されている場合は、リモート・マシンを PING してみると同時に、リモート・マシンからもこのマシンを PING してみてください。
- コネクターが実行しているかどうかをチェックしてください。
- サーバーが実行しているかどうかをチェックしてください。
- ビジネス・オブジェクトの構造がオペレーションと矛盾していないことを確認してください。

---

## 付録 A. コネクターの標準構成プロパティ

この付録では、WebSphere Business Integration アダプターのコネクタ・コンポーネントの標準構成プロパティについて説明します。この付録の内容は、次の統合ブローカーで実行されるコネクタを対象としています。

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (WebSphere Message Brokers (WMQI) と総称)
- WebSphere Application Server (WAS)

コネクタによっては、一部の標準プロパティが使用されないことがあります。Connector Configurator から統合ブローカーを選択すると、そのブローカーで実行されるアダプターについて構成する必要がある標準プロパティのリストが表示されます。

コネクタ固有のプロパティの詳細については、該当するアダプターのユーザーズ・ガイドを参照してください。

**注:** 本書では、ディレクトリー・パスの規則として円記号 (¥) を使用します。UNIX システムを使用している場合は、円記号をスラッシュ (/) に置き換えてください。また、各オペレーティング・システムの規則に従ってください。

---

### 新規プロパティと削除されたプロパティ

本リリースには、次の標準プロパティが追加されました。

#### 新規プロパティ

- XMLNamespaceFormat

#### 削除されたプロパティ

- RestartCount
- RHF2MessageDomain

---

### 標準コネクタ・プロパティの構成

アダプター・コネクタには 2 つのタイプの構成プロパティがあります。

- 標準構成プロパティ
- コネクタ固有のプロパティ

このセクションでは、標準構成プロパティについて説明します。コネクタ固有の構成プロパティについては、該当するアダプターのユーザーズ・ガイドを参照してください。

## Connector Configurator の使用

Connector Configurator からコネクタ・プロパティを構成します。Connector Configurator には、System Manager からアクセスします。Connector Configurator の使用法の詳細については、付録 C. の『Connector Configurator』を参照してください。

**注:** Connector Configurator と System Manager は、Windows システム上でのみ動作します。コネクタを UNIX システム上で稼動している場合でも、これらのツールがインストールされた Windows マシンが必要です。UNIX 上で動作するコネクタのコネクタ・プロパティを設定する場合は、Windows マシン上で System Manager を起動し、UNIX の統合ブローカーに接続してから、コネクタ用の Connector Configurator を開く必要があります。

## プロパティ値の設定と更新

プロパティ・フィールドのデフォルトの長さは 255 文字です。

コネクタは、以下の順序に従ってプロパティの値を決定します (最も番号の大きい項目が他の項目よりも優先されます)。

1. デフォルト
2. リポジトリ (WebSphere InterChange Server が統合ブローカーである場合のみ)
3. ローカル構成ファイル
4. コマンド行

コネクタは、始動時に構成値を取得します。実行時セッション中に 1 つ以上のコネクタ・プロパティの値を変更する場合は、プロパティの**更新メソッド**によって、変更を有効にする方法が決定されます。標準コネクタ・プロパティには、以下の 4 種類の更新メソッドがあります。

### • 動的

変更を System Manager に保管すると、変更が即時に有効になります。例えば WebSphere Message Brokers で稼動している場合など、コネクタがスタンドアロン・モードで (System Manager から独立して) 稼動している場合は、構成ファイルでのみプロパティを変更できます。この場合、動的更新は実行できません。

### • コンポーネント再始動

System Manager でコネクタを停止してから再始動しなければ、変更が有効になりません。アプリケーション固有コンポーネントまたは統合ブローカーを停止、再始動する必要はありません。

### • サーバー再始動

アプリケーション固有のコンポーネントおよび統合ブローカーを停止して再始動しなければ、変更が有効になりません。

### • エージェント再始動 (ICS のみ)

アプリケーション固有のコンポーネントを停止して再始動しなければ、変更が有効になりません。

特定のプロパティの更新方法を確認するには、「Connector Configurator」ウィンドウ内の「更新メソッド」列を参照するか、次に示すプロパティの要約の表の「更新メソッド」列を参照してください。

## 標準プロパティの要約

表 22 は、標準コネクタ構成プロパティの早見表です。標準プロパティの依存関係は RepositoryDirectory に基づいているため、コネクタによっては使用されないプロパティがあり、使用する統合ブローカーによってプロパティの設定が異なる可能性があります。

コネクタを実行する前に、これらのプロパティの一部の値を設定する必要があります。各プロパティの詳細については、次のセクションを参照してください。

表 22. 標準構成プロパティの要約

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
AdminInQueue	有効な JMS キュー名	CONNECTORNAME /ADMININQUEUE	コンポーネント再始動	Delivery Transport は JMS
AdminOutQueue	有効な JMS キュー名	CONNECTORNAME /ADMINOUTQUEUE	コンポーネント再始動	Delivery Transport は JMS
AgentConnections	1 から 4	1	コンポーネント再始動	Delivery Transport は MQ および IDL: Repository Directory は <REMOTE>
AgentTraceLevel	0 から 5	0	動的	
ApplicationName	アプリケーション名	コネクタ・アプリケーション名として指定された値	コンポーネント再始動	
BrokerType	ICS、WMQI、WAS			
CharacterEncoding	ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437 注: これは、サポートされる値の一部です。	ascii7	コンポーネント再始動	
ConcurrentEventTriggeredFlows	1 から 32、767	1	コンポーネント再始動	Repository Directory は <REMOTE>
ContainerManagedEvents	値なしまたは JMS	値なし	コンポーネント再始動	Delivery Transport は JMS
ControllerStoreAndForwardMode	true または false	True	動的	Repository Directory は <REMOTE>
ControllerTraceLevel	0 から 5	0	動的	Repository Directory は <REMOTE>
DeliveryQueue		CONNECTORNAME /DELIVERYQUEUE	コンポーネント再始動	JMS トランスポートのみ
DeliveryTransport	MQ、IDL、または JMS	JMS	コンポーネント再始動	Repository Directory がローカルの場合は、値は JMS のみ

表 22. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
DuplicateEventElimination	True または False	False	コンポーネント再始動	JMS トランスポートのみ: Container Managed Events は <NONE> でなければならない
FaultQueue		CONNECTORNAME/FAULTQUEUE	コンポーネント再始動	JMS トランスポートのみ
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory または CxCommon.Messaging.jms.SonicMQFactory または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	JMS トランスポートのみ
jms.MessageBrokerName	FactoryClassName が IBM の場合は crossworlds.queue.manager を使用。FactoryClassName が Sonic の場合は localhost:2506 を使用。	crossworlds.queue.manager	コンポーネント再始動	JMS トランスポートのみ
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	JMS トランスポートのみ
jms.Password	任意の有効なパスワード		コンポーネント再始動	JMS トランスポートのみ
jms.UserName	任意の有効な名前		コンポーネント再始動	JMS トランスポートのみ
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	Repository Directory は <REMOTE>
ListenerConcurrency	1 から 100	1	コンポーネント再始動	Delivery Transport は MQ でなければならない
Locale	en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR 注: これは、サポートされるロケールの一部です。	en_US	コンポーネント再始動	
LogAtInterchangeEnd	True または False	False	コンポーネント再始動	Repository Directory は <REMOTE> でなければならない

表 22. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE> でなければならぬ
MessageFileName	パスまたはファイル名	InterchangeSystem.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	CONNECTORNAME/MONITORQUEUE	コンポーネント再始動	JMS トランスポートのみ: DuplicateEvent Elimination は True でなければならぬ
OADAutoRestartAgent	True または False	False	動的	Repository Directory は <REMOTE> でなければならぬ
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE> でなければならぬ
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE> でなければならぬ
PollEndTime	HH:MM	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒) no (ポーリングを使用不可にする) key (コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力された場合にのみポーリングする)	10000	動的	
PollQuantity	1 から 500	1	エージェント再始動	JMS トランスポートのみ: Container Managed Events を指定
PollStartTime	HH:MM(HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	

表 22. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
RepositoryDirectory	メタデータ・リポジトリの場所		エージェント再始動	ICS の場合は <REMOTE> に設定する。 WebSphere MQ Message Brokers および WAS の場合: C:\crossworlds¥repository に設定する
RequestQueue	有効な JMS キュー名	CONNECTORNAME/REQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
ResponseQueue	有効な JMS キュー名	CONNECTORNAME/RESPONSEQUEUE	コンポーネント再始動	Delivery Transport が JMS の場合: Repository Directory が <REMOTE> の場合のみ必要
RestartRetryCount	0 から 99	3	動的	
RestartRetryInterval	適切な正数 (単位: 分): 1 から 2147483547	1	動的	
SourceQueue	有効な WebSphere MQ 名	CONNECTORNAME/SOURCEQUEUE	エージェント再始動	Delivery Transport が JMS であり、かつ Container Managed Events が指定されている場合のみ
SynchronousRequestQueue		CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
SynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	Delivery Transport は JMS
SynchronousResponseQueue		CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
WireFormat	CwXML、CwBO	CwXML	エージェント再始動	Repository Directory が <REMOTE> でない場合は CwXML。 Repository Directory が <REMOTE> であれば CwBO
WsifSynchronousRequest Timeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	WAS のみ
XMLNamespaceFormat	short、long	short	エージェント再始動	WebSphere MQ Message Brokers および WAS のみ



---

## 標準構成プロパティ

このセクションでは、各標準コネクタ構成プロパティの定義を示します。

### AdminInQueue

統合ブローカーからコネクタへ管理メッセージが送信されるときに使用されるキューです。

デフォルト値は `CONNECTORNAME/ADMININQUEUE` です。

### AdminOutQueue

コネクタから統合ブローカーへ管理メッセージが送信されるときに使用されるキューです。

デフォルト値は `CONNECTORNAME/ADMINOUTQUEUE` です。

### AgentConnections

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用可能です。

`AgentConnections` プロパティは、`orb.init[]` により開かれる ORB 接続の数を制御します。

デフォルトでは、このプロパティの値は 1 に設定されます。このデフォルト値を変更する必要はありません。

### AgentTraceLevel

アプリケーション固有のコンポーネントのトレース・メッセージのレベルです。デフォルト値は 0 です。コネクタは、設定されたトレース・レベル以下の該当するトレース・メッセージをすべてデリバリーします。

### ApplicationName

コネクタのアプリケーションを一意的に特定する名前です。この名前は、システム管理者が WebSphere Business Integration システム環境をモニターするために使用されます。コネクタを実行する前に、このプロパティに値を指定する必要があります。

### BrokerType

使用する統合ブローカー・タイプを指定します。オプションは ICS、WebSphere Message Brokers (WMQI、WMQIB または WBIMB) または WAS です。

### CharacterEncoding

文字 (アルファベットの文字、数値表現、句読記号など) から数値へのマッピングに使用する文字コード・セットを指定します。

**注:** Java ベースのコネクターでは、このプロパティは使用しません。C++ ベースのコネクターでは、現在、このプロパティに `ascii7` という値が使用されています。

デフォルトでは、ドロップ・リストには、サポートされる文字エンコードの一部のみが表示されます。ドロップ・リストに、サポートされる他の値を追加するには、製品ディレクトリーにある `¥Data¥Std¥stdConnProps.xml` ファイルを手動で変更する必要があります。詳細については、Connector Configurator に関する付録を参照してください。

## ConcurrentEventTriggeredFlows

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

コネクターがイベントのデリバリー時に並行処理できるビジネス・オブジェクトの数を決定します。この属性の値を、並行してマップおよびデリバリーできるビジネス・オブジェクトの数に設定します。例えば、この属性の値を 5 に設定すると、5 個のビジネス・オブジェクトが並行して処理されます。デフォルト値は 1 です。

このプロパティを 1 よりも大きい値に設定すると、ソース・アプリケーションのコネクターが、複数のイベント・ビジネス・オブジェクトを同時にマップして、複数のコラボレーション・インスタンスにそれらのビジネス・オブジェクトを同時にデリバリーすることができます。これにより、統合ブローカーへのビジネス・オブジェクトのデリバリーにかかる時間、特にビジネス・オブジェクトが複雑なマップを使用している場合のデリバリー時間が短縮されます。ビジネス・オブジェクトのコラボレーションに到達する速度を増大させると、システム全体のパフォーマンスを向上させることができます。

ソース・アプリケーションから宛先アプリケーションまでのフロー全体に並行処理を実装するには、次のようにする必要があります。

- **Maximum number of concurrent events** プロパティの値を増加して、コラボレーションが複数のスレッドを使用できるように構成します。
- 宛先アプリケーションのアプリケーション固有コンポーネントが複数の要求を並行して実行できることを確認します。つまり、このコンポーネントがマルチスレッド化されているか、またはコネクター・エージェント並列処理を使用でき、複数プロセスに対応するよう構成されている必要があります。Parallel Process Degree 構成プロパティに、1 より大きい値を設定します。

ConcurrentEventTriggeredFlows プロパティは、順次に実行される単一スレッド処理であるコネクターのポーリングでは無効です。

## ContainerManagedEvents

このプロパティにより、JMS イベント・ストアを使用する JMS 対応コネクターが、保証付きイベント・デリバリーを提供できるようになります。保証付きイベント・デリバリーでは、イベントはソース・キューから除去され、単一 JMS トランザクションとして宛先キューに配置されます。

デフォルト値は No value です。

ContainerManagedEvents を JMS に設定した場合には、保証付きイベント・デリバリーを使用できるように次のプロパティも構成する必要があります。

- PollQuantity = 1 から 500
- SourceQueue = CONNECTORNAME/SOURCEQUEUE

また、MimeType、DHClass、および DataHandlerConfigMOName (オプション) プロパティを設定したデータ・ハンドラーも構成する必要があります。これらのプロパティの値を設定するには、Connector Configurator の「データ・ハンドラー」タブを使用します。「データ・ハンドラー」タブの値のフィールドは、ContainerManagedEvents を JMS に設定した場合にのみ表示されます。

**注:** ContainerManagedEvents を JMS に設定した場合、コネクタはその pollForEvents() メソッドを呼び出さなくなるため、そのメソッドの機能は使用できなくなります。

このプロパティは、DeliveryTransport プロパティが値 JMS に設定されている場合にのみ表示されます。

## ControllerStoreAndForwardMode

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

宛先側のアプリケーション固有のコンポーネントが使用不可であることをコネクタ・コントローラーが検出した場合に、コネクタ・コントローラーが実行する動作を設定します。

このプロパティを true に設定した場合、イベントが ICS に到達したときに宛先側のアプリケーション固有のコンポーネントが使用不可であれば、コネクタ・コントローラーはそのアプリケーション固有のコンポーネントへの要求をブロックします。アプリケーション固有のコンポーネントが作動可能になると、コネクタ・コントローラーはアプリケーション固有のコンポーネントにその要求を転送します。

ただし、コネクタ・コントローラーが宛先側のアプリケーション固有のコンポーネントにサービス呼び出し要求を転送した後でこのコンポーネントが使用不可になった場合、コネクタ・コントローラーはその要求を失敗させます。

このプロパティを false に設定した場合、コネクタ・コントローラーは、宛先側のアプリケーション固有のコンポーネントが使用不可であることを検出すると、ただちにすべてのサービス呼び出し要求を失敗させます。

デフォルト値は true です。

## ControllerTraceLevel

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

コネクタ・コントローラーのトレース・メッセージのレベルです。デフォルト値は 0 です。

## DeliveryQueue

DeliveryTransport が JMS の場合のみ適用されます。

コネクタから統合ブローカーへビジネス・オブジェクトが送信されるときに使用されるキューです。

デフォルト値は CONNECTORNAME/DELIVERYQUEUE です。

## DeliveryTransport

イベントのデリバリーのためのトランスポート機構を指定します。指定可能な値は、WebSphere MQ の MQ、CORBA IIOP の IDL、Java Messaging Service の JMS です。

- ICS がブローカー・タイプの場合は、DeliveryTransport プロパティの指定可能な値は MQ、IDL、または JMS であり、デフォルトは IDL になります。
- RepositoryDirectory がローカル・ディレクトリーの場合は、指定可能な値は JMS のみです。

DeliveryTransport プロパティに指定されている値が、MQ または IDL である場合、コネクタは、CORBA IIOP を使用してサービス呼び出し要求と管理メッセージを送信します。

### WebSphere MQ および IDL

イベントのデリバリー・トランスポートには、IDL ではなく WebSphere MQ を使用してください (1 種類の製品だけを使用する必要がある場合を除きます)。

WebSphere MQ が IDL よりも優れている点は以下のとおりです。

- 非同期通信:  
WebSphere MQ を使用すると、アプリケーション固有のコンポーネントは、サーバーが利用不能である場合でも、イベントをポーリングして永続的に格納することができます。
- サーバー・サイド・パフォーマンス:  
WebSphere MQ を使用すると、サーバー・サイドのパフォーマンスが向上します。最適化モードでは、WebSphere MQ はイベントへのポインターのみをリポジトリ・データベースに格納するので、実際のイベントは WebSphere MQ キュー内に残ります。これにより、サイズが大きい可能性のあるイベントをリポジトリ・データベースに書き込む必要がありません。
- エージェント・サイド・パフォーマンス:  
WebSphere MQ を使用すると、アプリケーション固有のコンポーネント側のパフォーマンスが向上します。WebSphere MQ を使用すると、コネクタのポーリング・スレッドは、イベントを選出した後、コネクタのキューにそのイベントを入れ、次のイベントを選出します。この方法は IDL よりも高速で、IDL の場合、コネクタのポーリング・スレッドは、イベントを選出した後、ネットワーク経由でサーバー・プロセスにアクセスしてそのイベントをリポジトリ・データベースに永続的に格納してから、次のイベントを選出する必要があります。

### JMS

Java Messaging Service (JMS) を使用しての、コネクタとクライアント・コネクタ・フレームワークとの間の通信を可能にします。

JMS をデリバリー・トランスポートとして選択した場合は、`.jms.MessageBrokerName`、`.jms.FactoryClassName`、`.jms.Password`、`.jms.UserName` などの追加の JMS プロパティが Connector Configurator 内に表示されます。このうち最初の 2 つは、このトランスポートの必須プロパティです。

**重要:** 以下の環境では、コネクタに JMS トランスポート機構を使用すると、メモリ制限が発生することもあります。

- AIX 5.0
- WebSphere MQ 5.3.0.1
- ICS が統合ブローカーの場合

この環境では、WebSphere MQ クライアント内でメモリが使用されるため、(サーバー側の) コネクタ・コントローラと (クライアント側の) コネクタの両方を始動するのは困難な場合があります。ご使用のシステムのプロセス・ヒープ・サイズが 768M 未満である場合には、次のように設定することをお勧めします。

- `CWSharedEnv.sh` スクリプト内で `LDR_CNTRL` 環境変数を設定する。

このスクリプトは、製品ディレクトリ配下の `¥bin` ディレクトリにあります。テキスト・エディターを使用して、`CWSharedEnv.sh` スクリプトの最初の行として次の行を追加します。

```
export LDR_CNTRL=MAXDATA=0x30000000
```

この行は、ヒープ・メモリの使用量を最大 768 MB (3 セグメント \* 256 MB) に制限します。プロセス・メモリがこの制限値を超えると、ページ・スワッピングが発生し、システムのパフォーマンスに悪影響を与える場合があります。

- `IPCCBaseAddress` プロパティの値を 11 または 12 に設定する。このプロパティの詳細については、「システム・インストール・ガイド (UNIX 版)」を参照してください。

## DuplicateEventElimination

このプロパティを `true` に設定すると、JMS 対応コネクタによるデリバリー・キューへの重複イベントのデリバリーが防止されます。この機能を使用するには、コネクタに対し、アプリケーション固有のコード内でビジネス・オブジェクトの **ObjectEventId** 属性として一意のイベント ID が設定されている必要があります。これはコネクタ開発時に設定されます。

このプロパティは、`false` に設定することもできます。

**注:** `DuplicateEventElimination` を `true` に設定する際は、`MonitorQueue` プロパティを構成して保証付きイベント・デリバリーを使用可能にする必要があります。

## FaultQueue

コネクタでメッセージを処理中にエラーが発生すると、コネクタは、そのメッセージを状況表示および問題説明とともにこのプロパティに指定されているキューに移動します。

デフォルト値は `CONNECTORNAME/FAULTQUEUE` です。

## JvmMaxHeapSize

エージェントの最大ヒープ・サイズ (メガバイト単位)。このプロパティは、RepositoryDirectory の値が <REMOTE> の場合にのみ適用されます。

デフォルト値は 128M です。

## JvmMaxNativeStackSize

エージェントの最大ネイティブ・スタック・サイズ (キロバイト単位)。このプロパティは、RepositoryDirectory の値が <REMOTE> の場合にのみ適用されます。

デフォルト値は 128K です。

## JvmMinHeapSize

エージェントの最小ヒープ・サイズ (メガバイト単位)。このプロパティは、RepositoryDirectory の値が <REMOTE> の場合にのみ適用されます。

デフォルト値は 1M です。

## jms.FactoryClassName

JMS プロバイダーのためにインスタンスを生成するクラス名を指定します。JMS をデリバリー・トランスポート機構 (DeliveryTransport) として選択する際は、このコネクタ・プロパティを必ず設定してください。

デフォルト値は CxCommon.Messaging.jms.IBMMQSeriesFactory です。

## jms.MessageBrokerName

JMS プロバイダーのために使用するブローカー名を指定します。JMS をデリバリー・トランスポート機構 (DeliveryTransport) として選択する際は、このコネクタ・プロパティを必ず設定してください。

デフォルト値は crossworlds.queue.manager です。

## jms.NumConcurrentRequests

コネクタに対して同時に送信することができる並行サービス呼び出し要求の数 (最大値) を指定します。この最大値に達した場合、新規のサービス呼び出し要求はブロックされ、既存のいずれかの要求が完了した後で処理されます。

デフォルト値は 10 です。

## jms.Password

JMS プロバイダーのためのパスワードを指定します。このプロパティの値はオプションです。

デフォルトはありません。



## jms.UserName

JMS プロバイダーのためのユーザー名を指定します。このプロパティの値はオプションです。

デフォルトはありません。

## ListenerConcurrency

このプロパティは、統合ブローカーとして ICS を使用する場合の MQ Listener でのマルチスレッド化をサポートしています。このプロパティにより、データベースへの複数イベントの書き込み操作をバッチ処理できるので、システム・パフォーマンスが向上します。デフォルト値は 1 です。

このプロパティは、MQ トランSPORTを使用するコネクタにのみ適用されます。DeliveryTransport プロパティには MQ を設定してください。

## Locale

言語コード、国または地域、および、希望する場合には、関連した文字コード・セットを指定します。このプロパティの値は、データの照合やソート順、日付と時刻の形式、通貨記号などの国/地域別情報を決定します。

ロケール名は、次の書式で指定します。

`ll_TT.codeset`

ここで、以下のように説明されます。

<code>ll</code>	2 文字の言語コード (普通は小文字)
<code>TT</code>	2 文字の国または地域コード (普通は大文字)
<code>codeset</code>	関連文字コード・セットの名前。名前のこの部分は、通常、オプションです。

デフォルトでは、ドロップ・リストには、サポートされるロケールの一部のみが表示されます。ドロップ・リストに、サポートされる他の値を追加するには、製品ディレクトリーにある `¥Data¥Std¥stdConnProps.xml` ファイルを手動で変更する必要があります。詳細については、Connector Configurator に関する付録を参照してください。

デフォルト値は `en_US` です。コネクタがグローバル化に対応していない場合、このプロパティの有効な値は `en_US` のみです。特定のコネクタがグローバル化に対応しているかどうかを判別するには、以下の Web サイトにあるコネクタのバージョン・リストを参照してください。

<http://www.ibm.com/software/websphere/wbiadapters/infocenter>、または  
<http://www.ibm.com/websphere/integration/wicsserver/infocenter>

## LogAtInterchangeEnd

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。ブローカーのログ宛先にログを記録すると、電子メール通知もオンになります。これにより、

エラーまたは致命的エラーが発生すると、InterchangeSystem.cfg ファイルに指定された MESSAGE\_RECIPIENT に対する電子メール・メッセージが生成されます。

例えば、LogAtInterChangeEnd を true に設定した場合にコネクタからアプリケーションへの接続が失われると、指定されたメッセージ宛先に、電子メール・メッセージが送信されます。デフォルト値は false です。

## MaxEventCapacity

コントローラー・バッファ内のイベントの最大数。このプロパティはフロー制御が使用し、RepositoryDirectory プロパティの値が <REMOTE> の場合にのみ適用されます。

値は 1 から 2147483647 の間の正整数です。デフォルト値は 2147483647 です。

## MessageFileName

コネクタ・メッセージ・ファイルの名前です。メッセージ・ファイルの標準位置は %connectors%messages です。メッセージ・ファイルが標準位置に格納されていない場合は、メッセージ・ファイル名を絶対パスで指定します。

コネクタ・メッセージ・ファイルが存在しない場合は、コネクタは InterchangeSystem.txt をメッセージ・ファイルとして使用します。このファイルは、製品ディレクトリーに格納されています。

**注:** 特定のコネクタについて、コネクタ独自のメッセージ・ファイルがあるかどうかを判別するには、該当するアダプターのユーザズ・ガイドを参照してください。

## MonitorQueue

コネクタが重複イベントをモニターするために使用する論理キューです。このプロパティは、DeliveryTransport プロパティ値が JMS であり、かつ DuplicateEventElimination が TRUE に設定されている場合にのみ使用されます。

デフォルト値は CONNECTORNAME/MONITORQUEUE です。

## OADAutoRestartAgent

RepositoryDirectory が <REMOTE> の場合のみ有効です。

コネクタが自動再始動およびリモート再始動機能を使用するかどうかを指定します。この機能では、MQ により起動される Object Activation Daemon (OAD) を使用して、異常シャットダウン後にコネクタを再始動したり、System Monitor からリモート・コネクタを始動したりします。

自動再始動機能およびリモート再始動機能を使用可能にするには、このプロパティを true に設定する必要があります。MQ によりトリガーされる OAD 機能の構成方法については、「システム・インストール・ガイド (Windows 版)」または「システム・インストール・ガイド (UNIX 版)」を参照してください。

デフォルト値は false です。



## OADMaxNumRetry

RepositoryDirectory が <REMOTE> の場合のみ有効です。

異常シャットダウンの後で MQ によりトリガーされる OAD がコネクタの再始動を自動的に試行する回数の最大数を指定します。このプロパティを有効にするためには、OADAutoRestartAgent プロパティを true に設定する必要があります。

デフォルト値は 1000 です。

## OADRetryTimeInterval

RepositoryDirectory が <REMOTE> の場合のみ有効です。

MQ によりトリガーされる OAD の再試行時間間隔の分数を指定します。コネクタ・エージェントがこの再試行時間間隔内に再始動しない場合は、コネクタ・コントローラーはコネクタ・エージェントを再び再始動するように OAD に要求します。OAD はこの再試行プロセスを OADMaxNumRetry プロパティで指定された回数だけ繰り返します。このプロパティを有効にするためには、OADAutoRestartAgent プロパティを true に設定する必要があります。

デフォルト値は 10 です。

## PollEndTime

イベント・キューのポーリングを停止する時刻です。形式は HH:MM です。ここで、HH は 0 から 23 時を表し、MM は 0 から 59 分を表します。

このプロパティには必ず有効な値を指定してください。デフォルト値は HH:MM ですが、この値は必ず変更する必要があります。

## PollFrequency

ポーリング・アクション間の時間の長さです。PollFrequency は以下の値のいずれかに設定します。

- ポーリング・アクション間のミリ秒数。
- ワード key。コネクタは、コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力されたときのみポーリングを実行します。このワードは小文字で入力します。
- ワード no。コネクタはポーリングを実行しません。このワードは小文字で入力します。

デフォルト値は 10000 です。

**重要:** 一部のコネクタでは、このプロパティの使用が制限されています。このプロパティが使用されるかどうかを特定のコネクタについて判断するには、該当するアダプター・ガイドのインストールと構成についての章を参照してください。

## PollQuantity

コネクタがアプリケーションからポーリングする項目の数を指定します。アダプターにコネクタ固有のポーリング数設定プロパティがある場合、標準プロパティの値は、このコネクタ固有のプロパティの設定値によりオーバーライドされます。

## PollStartTime

イベント・キューのポーリングを開始する時刻です。形式は *HH:MM* です。ここで、*HH* は 0 から 23 時を表し、*MM* は 0 から 59 分を表します。

このプロパティには必ず有効な値を指定してください。デフォルト値は *HH:MM* ですが、この値は必ず変更する必要があります。

## RequestQueue

統合ブローカーが、ビジネス・オブジェクトをコネクタに送信するときに使用されるキューです。

デフォルト値は `CONNECTOR/REQUESTQUEUE` です。

## RepositoryDirectory

コネクタが XML スキーマ文書を読み取るリポジトリの場所です。この XML スキーマ文書には、ビジネス・オブジェクト定義のメタデータが含まれています。

統合ブローカーが ICS の場合はこの値を `<REMOTE>` に設定する必要があります。これは、コネクタが InterChange Server リポジトリからこの情報を取得するためです。

統合ブローカーが WebSphere Message Broker または WAS の場合は、この値を `<local directory>` に設定する必要があります。

## ResponseQueue

`DeliveryTransport` が JMS の場合のみ適用可能で、`RepositoryDirectory` が `<REMOTE>` の場合のみ必須です。

JMS 応答キューを指定します。JMS 応答キューは、応答メッセージをコネクタ・フレームワークから統合ブローカーへデリバリーします。統合ブローカーが ICS の場合、サーバーは要求を送信し、JMS 応答キューの応答メッセージを待ちます。

## RestartRetryCount

コネクタによるコネクタ自体の再始動の試行回数を指定します。このプロパティを並列コネクタに対して使用する場合、コネクタのマスター側のアプリケーション固有のコンポーネントがスレーブ側のアプリケーション固有のコンポーネントの再始動を試行する回数が指定されます。

デフォルト値は 3 です。

## RestartRetryInterval

コネクタによるコネクタ自体の再始動の試行間隔を分単位で指定します。このプロパティを並列コネクタに対して使用する場合、コネクタのマスター側のアプリケーション固有のコンポーネントがスレーブ側のアプリケーション固有のコンポーネントの再始動を試行する間隔が指定されます。指定可能な値の範囲は 1 から 2147483647 です。

デフォルト値は 1 です。

## SourceQueue

DeliveryTransport が JMS で、ContainerManagedEvents が指定されている場合のみ適用されます。

JMS イベント・ストアを使用する JMS 対応コネクタでの保証付きイベント・デリバリーをサポートするコネクタ・フレームワークに、JMS ソース・キューを指定します。詳細については、58 ページの『ContainerManagedEvents』を参照してください。

デフォルト値は CONNECTOR/SOURCEQUEUE です。

## SynchronousRequestQueue

DeliveryTransport が JMS の場合のみ適用されます。

同期応答を要求する要求メッセージを、コネクタ・フレームワークからブローカーに配信します。このキューは、コネクタが同期実行を使用する場合にのみ必要です。同期実行の場合、コネクタ・フレームワークは、SynchronousRequestQueue にメッセージを送信し、SynchronousResponseQueue でブローカーから戻される応答を待機します。コネクタに送信される応答メッセージには、元のメッセージの ID を指定する 相関 ID が含まれています。

デフォルトは CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE です。

## SynchronousResponseQueue

DeliveryTransport が JMS の場合のみ適用されます。

同期要求に対する応答として送信される応答メッセージを、ブローカーからコネクタ・フレームワークに配信します。このキューは、コネクタが同期実行を使用する場合にのみ必要です。

デフォルトは CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE です。

## SynchronousRequestTimeout

DeliveryTransport が JMS の場合のみ適用されます。

コネクタが同期要求への応答を待機する時間を分単位で指定します。コネクタは、指定された時間内に応答を受信できなかった場合、元の同期要求メッセージをエラー・メッセージとともに障害キューに移動します。

デフォルト値は 0 です。

## WireFormat

トランスポートのメッセージ・フォーマットです。

- RepositoryDirectory がローカル・ディレクトリーの場合は、設定は CwXML になります。
- RepositoryDirectory の値が <REMOTE> の場合には、設定値は CwBO です。

## WsifSynchronousRequest Timeout

WAS 統合ブローカーでのみ使用されます。

コネクタが同期要求への応答を待機する時間を分単位で指定します。コネクタは、指定された時間内に応答を受信できなかった場合、元の同期要求メッセージをエラー・メッセージとともに障害キューに移動します。

デフォルト値は 0 です。

## XMLNamespaceFormat

WebSphere Message Brokers および WAS 統合ブローカーでのみ使用されます。

ビジネス・オブジェクト定義の XML 形式でネーム・スペースを short と long のどちらにするかをユーザーが指定できるようにするための、強力なプロパティです。

デフォルト値は short です。

---

## 付録 B. Connector Configurator

この付録では、Connector Configurator を使用してアダプターの構成プロパティ値を設定する方法について説明します。

Connector Configurator を使用して次の作業を行います。

- コネクタを構成するためのコネクタ固有のプロパティ・テンプレートを作成する
- 構成ファイルを作成する
- 構成ファイル内のプロパティを設定する

注:

本書では、ディレクトリー・パスの規則として円記号 (¥) を使用します。UNIX システムを使用している場合は、円記号をスラッシュ (/) に置き換えてください。また、各オペレーティング・システムの規則に従ってください。

この付録では、次のトピックについて説明します。

- 69 ページの『Connector Configurator の概要』
- 70 ページの『Connector Configurator の始動』
- 71 ページの『コネクタ固有のプロパティ・テンプレートの作成』
- 74 ページの『新規構成ファイルの作成』
- 77 ページの『構成ファイル・プロパティの設定』
- 85 ページの『グローバル化環境における Connector Configurator の使用』

---

### Connector Configurator の概要

Connector Configurator では、次の統合ブローカーで使用するアダプターのコネクタ・コンポーネントを構成できます。

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (WebSphere Message Brokers (WMQI) と総称)
- WebSphere Application Server (WAS)

Connector Configurator を使用して次の作業を行います。

- コネクタを構成するためのコネクタ固有のプロパティ・テンプレートを作成します。
- **コネクタ構成ファイル**を作成します。インストールするコネクタごとに構成ファイルを 1 つ作成する必要があります。
- 構成ファイル内のプロパティを設定します。  
場合によっては、コネクタ・テンプレートでプロパティに対して設定されているデフォルト値を変更する必要があります。また、サポートされるビジネス・オブジェクト定義と、ICS の場合はコラボレーションとともに使用するマップを

指定し、必要に応じてメッセージング、ロギング、トレース、およびデータ・ハンドラー・パラメーターを指定する必要があります。

Connector Configurator の実行モードと使用する構成ファイルのタイプは、実行する統合ブローカーによって異なります。例えば、使用している統合ブローカーが WMQI の場合、Connector Configurator を System Manager から実行するのではなく、直接実行します (70 ページの『スタンドアロン・モードでの Configurator の実行』を参照)。

コネクタ構成プロパティには、標準の構成プロパティ (すべてのコネクタにもつプロパティ) と、コネクタ固有のプロパティ (特定のアプリケーションまたはテクノロジーのためにコネクタで必要なプロパティ) とが含まれます。

**標準プロパティ**はすべてのコネクタにより使用されるので、標準プロパティを新規に定義する必要はありません。ファイルを作成すると、Connector Configurator により標準プロパティがこの構成ファイルに挿入されます。ただし、Connector Configurator で各標準プロパティの値を設定する必要があります。

標準プロパティの範囲は、ブローカーと構成によって異なる可能性があります。特定のプロパティに特定の値が設定されている場合にのみ使用できるプロパティがあります。Connector Configurator の「標準のプロパティ」ウィンドウには、特定の構成で設定可能なプロパティが表示されます。

ただし**コネクタ固有プロパティ**の場合は、最初にプロパティを定義し、その値を設定する必要があります。このため、特定のアダプターのコネクタ固有プロパティのテンプレートを作成します。システム内で既にテンプレートが作成されている場合には、作成されているテンプレートを使用します。システム内でまだテンプレートが作成されていない場合には、71 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

**注:** Connector Configurator は、Windows 環境内でのみ実行されます。UNIX 環境でコネクタを実行する場合には、Windows で Connector Configurator を使用して構成ファイルを変更し、このファイルを UNIX 環境へコピーします。

---

## Connector Configurator の始動

以下の 2 種類のモードで Connector Configurator を開始および実行できます。

- スタンドアロン・モードで個別に実行
- System Manager から

### スタンドアロン・モードでの Configurator の実行

どのブローカーを実行している場合にも、Connector Configurator を個別に実行し、コネクタ構成ファイルを編集できます。

これを行うには、以下のステップを実行します。

- 「スタート」>「プログラム」から、「**IBM WebSphere InterChange Server**」>「**IBM WebSphere Business Integration Toolset**」>「**開発**」>「**Connector Configurator**」をクリックします。
- 「ファイル」>「新規」>「構成ファイル」を選択します。

- 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、ICS、WebSphere Message Brokers、または WAS を選択します。

Connector Configurator を個別に実行して構成ファイルを生成してから、System Manager に接続してこの構成ファイルを System Manager プロジェクトに保存することもできます (76 ページの『構成ファイルの完成』を参照)。

---

## System Manager からの Configurator の実行

System Manager から Connector Configurator を実行できます。

Connector Configurator を実行するには、以下のステップを実行します。

1. System Manager を開きます。
2. 「System Manager」ウィンドウで、「統合コンポーネント・ライブラリー」アイコンを展開し、「コネクタ」を強調表示します。
3. System Manager メニュー・バーから、「ツール」>「Connector Configurator」をクリックします。「Connector Configurator」ウィンドウが開き、「新規コネクタ」ダイアログ・ボックスが表示されます。
4. 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、ICS、WebSphere Message Brokers、または WAS を選択します。

既存の構成ファイルを編集するには、以下のステップを実行します。

1. 「System Manager」ウィンドウの「コネクタ」フォルダーでいずれかの構成ファイルを選択し、右クリックします。Connector Configurator が開き、この構成ファイルの統合ブローカー・タイプおよびファイル名が上部に表示されます。
2. 「標準のプロパティ」タブをクリックし、この構成ファイルに含まれているプロパティを確認します。

---

## コネクタ固有のプロパティ・テンプレートの作成

コネクタの構成ファイルを作成するには、コネクタ固有プロパティのテンプレートとシステム提供の標準プロパティが必要です。

コネクタ固有プロパティのテンプレートを新規に作成するか、または既存のファイルをテンプレートとして使用します。

- テンプレートの新規作成については、71 ページの『新規テンプレートの作成』を参照してください。
- 既存のファイルを使用する場合には、既存のテンプレートを変更し、新しい名前でのこのテンプレートを保管します。

### 新規テンプレートの作成

このセクションでは、テンプレートでプロパティを作成し、プロパティの一般特性および値を定義し、プロパティ間の依存関係を指定する方法について説明します。次にそのテンプレートを保管し、新規コネクタ構成ファイルを作成するためのベースとして使用します。



テンプレートは以下のように作成します。

1. 「ファイル」>「新規」>「コネクタ固有プロパティ・テンプレート」をクリックします。
2. 以下のフィールドを含む「コネクタ固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。
  - 「テンプレート」、「名前」  
このテンプレートが使用されるコネクタ（またはコネクタのタイプ）を表す固有の名前を入力します。テンプレートから新規構成ファイルを作成するためのダイアログ・ボックスを開くと、この名前が再度表示されます。
  - 「旧テンプレート」、「変更する既存のテンプレートを選択してください」  
「テンプレート名」表示に、現在使用可能なすべてのテンプレートの名前が表示されます。
  - テンプレートに含まれているコネクタ固有のプロパティ定義を調べるには、「テンプレート名」表示でそのテンプレートの名前を選択します。そのテンプレートに含まれているプロパティ定義のリストが「テンプレートのプレビュー」表示に表示されます。テンプレートを作成するときには、ご使用のコネクタに必要なプロパティ定義に類似したプロパティ定義が含まれている既存のテンプレートを使用できます。
3. 「テンプレート名」表示からテンプレートを選択し、その名前を「名前の検索」フィールドに入力し（または「テンプレート名」で自分の選択項目を強調表示し）、「次へ」をクリックします。

ご使用のコネクタで使用するコネクタ固有のプロパティが表示されるテンプレートが見つからない場合は、自分で作成する必要があります。

### 一般特性の指定

「次へ」をクリックしてテンプレートを選択すると、「プロパティ: コネクタ固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。このダイアログ・ボックスには、定義済みプロパティの「一般」特性のタブと「値」の制限のタブがあります。「一般」表示には以下のフィールドがあります。

- **一般:**
  - プロパティ・タイプ
  - 更新されたメソッド
  - 説明
- **フラグ**
  - 標準フラグ
- **カスタム・フラグ**
  - フラグ

プロパティの一般特性の選択を終えたら、「値」タブをクリックします。

### 値の指定

「値」タブを使用すると、プロパティの最大長、最大複数值、デフォルト値、または値の範囲を設定できます。編集可能な値も許可されます。これを行うには、以下のステップを実行します。



1. 「値」タブをクリックします。「一般」のパネルに代わって「値」の表示パネルが表示されます。
2. 「プロパティを編集」表示でプロパティの名前を選択します。
3. 「最大長」および「最大複数値」のフィールドで、変更を行います。次のステップで説明するように、プロパティの「プロパティ値」ダイアログ・ボックスを開かない限り、そのプロパティの変更内容は受け入れられませんので、注意してください。
4. 値テーブルの左上の隅にあるボックスを右マウス・ボタンでクリックしてから、「追加」をクリックします。「プロパティ値」ダイアログ・ボックスが表示されます。このダイアログ・ボックスではプロパティのタイプに応じて、値だけを入力できる場合と、値と範囲の両方を入力できる場合があります。適切な値または範囲を入力し、「OK」をクリックします。
5. 「値」パネルが最新表示され、「最大長」および「最大複数値」で行った変更が表示されます。以下のような 3 つの列があるテーブルが表示されます。
 

「値」の列には、「プロパティ値」ダイアログ・ボックスで入力した値と、以前に作成した値が表示されます。

「デフォルト値」の列では、値のいずれかをデフォルトとして指定することができます。

「値の範囲」の列には、「プロパティ値」ダイアログ・ボックスで入力した範囲が表示されます。

値が作成されて、グリッドに表示されると、そのテーブルの表示内から編集できるようになります。テーブルにある既存の値の変更を行うには、その行の行番号をクリックして行全体を選択します。次に「値」フィールドを右マウス・ボタンでクリックし、「値の編集 (Edit Value)」をクリックします。

## 依存関係の設定

「一般」タブと「値」タブで変更を行ったら、「次へ」をクリックします。「依存関係: コネクター固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。

依存プロパティは、別のプロパティの値が特定の条件に合致する場合にのみ、テンプレートに組み込まれて、構成ファイルで使用されるプロパティです。例えば、テンプレートに PollQuantity が表示されるのは、トランスポート機構が JMS であり、DuplicateEventElimination が True に設定されている場合のみです。プロパティを依存プロパティとして指定し、依存する条件を設定するには、以下のステップを実行します。

1. 「使用可能なプロパティ」表示で、依存プロパティとして指定するプロパティを選択します。
2. 「プロパティを選択」フィールドで、ドロップダウン・メニューを使用して、条件値を持たせるプロパティを選択します。
3. 「条件演算子」フィールドで以下のいずれかを選択します。
  - == (等しい)
  - != (等しくない)
  - > (より大)
  - < (より小)

>= (より大か等しい)

<= (より小か等しい)

4. 「条件値」フィールドで、依存プロパティをテンプレートに組み込むために必要な値を入力します。
5. 「使用可能なプロパティ」表示で依存プロパティを強調表示させて矢印をクリックし、「依存プロパティ」表示に移動させます。
6. 「完了」をクリックします。Connector Configurator により、XML 文書として入力した情報が、Connector Configurator がインストールされている %bin ディレクトリーの %data¥app の下に保管されます。

---

## 新規構成ファイルの作成

構成ファイルを新規に作成するには、最初に統合ブローカーを選択します。選択したブローカーによって、構成ファイルに記述されるプロパティが決まります。

ブローカーを選択するには、以下のステップを実行します。

- Connector Configurator のホーム・メニューで、「ファイル」>「新規」>「コネクター構成」をクリックします。「新規コネクター」ダイアログ・ボックスが表示されます。
- 「統合ブローカー」フィールドで、ICS 接続、WebSphere Message Brokers 接続、WAS 接続のいずれかを選択します。
- この章で後述する説明に従って「新規コネクター」ウィンドウの残りのフィールドに入力します。

また、以下の作業も実行できます。

- 「System Manager」ウィンドウで「コネクター」フォルダーを右クリックし、「新規コネクターの作成」を選択します。Connector Configurator が開き、「新規コネクター」ダイアログ・ボックスが表示されます。

## コネクター固有のテンプレートからの構成ファイルの作成

コネクター固有のテンプレートを作成すると、テンプレートを使用して構成ファイルを作成できます。

1. 「ファイル」>「新規」>「コネクター構成」をクリックします。
2. 以下のフィールドを含む「新規コネクター」ダイアログ・ボックス表示されます。

- **名前**

コネクターの名前を入力します。名前では大文字小文字が区別されます。入力する名前は、システムにインストールされているコネクターのファイル名に対応した一意の名前でなければなりません。

**重要:** Connector Configurator では、入力された名前のスペルはチェックされません。名前が正しいことを確認してください。

- **システム接続**

ICS 接続、WebSphere Message Brokers 接続、WAS のいずれかをクリックします。

- 「コネクタ固有プロパティ・テンプレート」を選択します。

ご使用のコネクタ用に設計したテンプレートの名前を入力します。「テンプレート名」表示に、使用可能なテンプレートが表示されます。「テンプレート名」表示で名前を選択すると、「プロパティ・テンプレートのプレビュー」表示に、そのテンプレートで定義されているコネクタ固有のプロパティが表示されます。

使用するテンプレートを選択し、「OK」をクリックします。

3. 構成しているコネクタの構成画面が表示されます。タイトル・バーに統合ブローカーとコネクタの名前が表示されます。ここですべてのフィールドに値を入力して定義を完了するか、ファイルを保管して後でフィールドに値を入力するかを選択できます。
4. ファイルを保管するには、「ファイル」>「保管」>「ファイルに」をクリックするか、「ファイル」>「保管」>「プロジェクトに」をクリックします。プロジェクトに保管するには、System Manager が実行中でなければなりません。ファイルとして保管する場合は、「ファイル・コネクタを保管」ダイアログ・ボックスが表示されます。\*.cfg をファイル・タイプとして選択し、「ファイル名」フィールド内に名前が正しいスペル (大文字と小文字の区別を含む) で表示されていることを確認してから、ファイルを保管するディレクトリにナビゲートし、「保管」をクリックします。Connector Configurator のメッセージ・パネルの状況表示に、構成ファイルが正常に作成されたことが示されます。

**重要:** ここで設定するディレクトリ・パスおよび名前は、コネクタの始動ファイルで指定するコネクタ構成ファイルのパスおよび名前に一致している必要があります。

5. この章で後述する手順に従って、「Connector Configurator」ウィンドウの各タブにあるフィールドに値を入力し、コネクタ定義を完了します。

---

## 既存ファイルの使用

使用可能な既存ファイルは、以下の 1 つまたは複数の形式になります。

- コネクタ定義ファイル。  
コネクタ定義ファイルは、特定のコネクタのプロパティと、適用可能なデフォルト値がリストされたテキスト・ファイルです。コネクタの配布パッケージの `¥repository` ディレクトリ内には、このようなファイルが格納されていることがあります (通常、このファイルの拡張子は `.txt` です。例えば、XML コネクタの場合は `CN_XML.txt` です)。
- ICS リポジトリ・ファイル。  
コネクタの以前の ICS インプリメンテーションで使用した定義は、そのコネクタの構成で使用されたりポジトリ・ファイルで使用可能になります。そのようなファイルの拡張子は、通常 `.in` または `.out` です。
- コネクタの以前の構成ファイル。  
これらのファイルの拡張子は、通常 `*.cfg` です。

これらのいずれのファイル・ソースにも、コネクタのコネクタ固有プロパティのほとんど、あるいはすべてが含まれますが、この章内の後で説明するように、コネクタ構成ファイルは、ファイルを開いて、プロパティを設定しない限り完成しません。

既存ファイルを使用してコネクタを構成するには、Connector Configurator でそのファイルを開き、構成を修正し、そのファイルを再度保管する必要があります。

以下のステップを実行して、ディレクトリーから \*.txt、\*.cfg、または \*.in ファイルを開きます。

1. Connector Configurator 内で、「ファイル」>「開く」>「ファイルから」をクリックします。
2. 「ファイル・コネクタを開く」ダイアログ・ボックス内で、以下のいずれかのファイル・タイプを選択して、使用可能なファイルを調べます。
  - 構成 (\*.cfg)
  - ICS リポジトリ (\*.in、\*.out)  
ICS 環境でのコネクタの構成にリポジトリ・ファイルが使用された場合には、このオプションを選択します。リポジトリ・ファイルに複数のコネクタ定義が含まれている場合は、ファイルを開くとすべての定義が表示されます。
  - すべてのファイル (\*.\*)  
コネクタのアダプター・パッケージに \*.txt ファイルが付属していた場合、または別の拡張子で定義ファイルが使用可能である場合は、このオプションを選択します。
3. ディレクトリー表示内で、適切なコネクタ定義ファイルへ移動し、ファイルを選択し、「開く」をクリックします。

System Manager プロジェクトからコネクタ構成を開くには、以下のステップを実行します。

1. System Manager を始動します。System Manager が開始されている場合にのみ、構成を System Manager から開いたり、System Manager に保管したりできます。
2. Connector Configurator を始動します。
3. 「ファイル」>「開く」>「プロジェクトから」をクリックします。

---

## 構成ファイルの完成

構成ファイルを開くか、プロジェクトからコネクタを開くと、「Connector Configurator」ウィンドウに構成画面が表示されます。この画面には、現在の属性と値が表示されます。

構成画面のタイトルには、ファイル内で指定された統合ブローカーとコネクタの名前が表示されます。正しいブローカーが設定されていることを確認してください。正しいブローカーが設定されていない場合、コネクタを構成する前にブローカー値を変更してください。これを行うには、以下のステップを実行します。

1. 「標準のプロパティ」タブで、BrokerType プロパティの値フィールドを選択します。ドロップダウン・メニューで、値 ICS、WMQI、または WAS を選択します。
2. 選択したブローカーに関連付けられているプロパティが「標準のプロパティ」タブに表示されます。ここでファイルを保管するか、または 79 ページの

『サポートされるビジネス・オブジェクト定義の指定』の説明に従い残りの構成フィールドに値を入力することができます。

3. 構成が完了したら、「ファイル」>「保管」>「プロジェクトに」を選択するか、または「ファイル」>「保管」>「ファイルに」を選択します。

ファイルに保管する場合は、\*.cfg を拡張子として選択し、ファイルの正しい格納場所を選択して、「保管」をクリックします。

複数のコネクタ構成を開いている場合、構成をすべてファイルに保管するには「すべてファイルに保管」を選択し、コネクタ構成をすべて System Manager プロジェクトに保管するには「すべてプロジェクトに保管」をクリックします。

Connector Configurator では、ファイルを保管する前に、必須の標準プロパティすべてに値が設定されているかどうかを確認されます。必須の標準プロパティに値が設定されていない場合、Connector Configurator は、検証が失敗したというメッセージを表示します。構成ファイルを保管するには、そのプロパティの値を指定する必要があります。

---

## 構成ファイル・プロパティの設定

新規のコネクタ構成ファイルを作成して名前を付けるとき、または既存のコネクタ構成ファイルを開くときには、Connector Configurator によって構成画面が表示されます。構成画面には、必要な構成値のカテゴリーに対応する複数のタブがあります。

Connector Configurator では、すべてのブローカーで実行されているコネクタで、以下のカテゴリーのプロパティに値が設定されている必要があります。

- 標準のプロパティ
- コネクタ固有のプロパティ
- サポートされるビジネス・オブジェクト
- トレース/ログ・ファイルの値
- データ・ハンドラー (保証付きイベント・デリバリーで JMS メッセージングを使用するコネクタの場合に該当する)

**注:** JMS メッセージングを使用するコネクタの場合は、データをビジネス・オブジェクトに変換するデータ・ハンドラーの構成に関して追加のカテゴリーが表示される場合があります。

**ICS** で実行されているコネクタの場合、以下のプロパティの値も設定されている必要があります。

- 関連付けられたマップ
- リソース
- メッセージング (該当する場合)

**重要:** Connector Configurator では、英語文字セットまたは英語以外の文字セットのいずれのプロパティ値も設定可能です。ただし、標準のプロパティおよびコネクタ固有プロパティ、およびサポートされるビジネス・オブジェクトの名前では、英語文字セットのみを使用する必要があります。

標準プロパティとコネクタ固有プロパティの違いは、以下のとおりです。



- コネクターの標準プロパティは、コネクターのアプリケーション固有のコンポーネントとブローカー・コンポーネントの両方によって共有されます。すべてのコネクターが同じ標準プロパティのセットを使用します。これらのプロパティの説明は、各アダプター・ガイドの付録 A にあります。変更できるのはこれらの値の一部のみです。
- アプリケーション固有のプロパティは、コネクターのアプリケーション固有コンポーネント (アプリケーションと直接対話するコンポーネント) のみに適用されます。各コネクターには、そのコネクターのアプリケーションだけで使用されるアプリケーション固有のプロパティがあります。これらのプロパティには、デフォルト値が用意されているものもあれば、そうでないものもあります。また、一部のデフォルト値は変更することができます。各アダプター・ガイドのインストールおよび構成の章に、アプリケーション固有のプロパティおよび推奨値が記述されています。

「標準プロパティ」と「コネクター固有プロパティ」のフィールドは、どのフィールドが構成可能であるかを示すために色分けされています。

- 背景がグレーのフィールドは、標準のプロパティを表します。値を変更することはできますが、名前の変更およびプロパティの除去はできません。
- 背景が白のフィールドは、アプリケーション固有のプロパティを表します。これらのプロパティは、アプリケーションまたはコネクターの特定のニーズによって異なります。値の変更も、これらのプロパティの除去も可能です。
- 「値」フィールドは構成できます。
- 「更新メソッド」フィールドは通知用であり、構成できません。このフィールドは、値が変更されたプロパティをアクティブにするために必要なアクションを示します。

## 標準コネクター・プロパティの設定

標準のプロパティの値を変更するには、以下の手順を実行します。

1. 値を設定するフィールド内でクリックします。
2. 値を入力するか、ドロップダウン・メニューが表示された場合にはメニューから値を選択します。
3. 標準のプロパティの値をすべて入力後、以下のいずれかを実行することができます。
  - 変更内容を破棄し、元の値を保持したままで Connector Configurator を終了するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「いいえ」をクリックします。
  - Connector Configurator 内の他のカテゴリーの値を入力するには、そのカテゴリーのタブを選択します。「標準のプロパティ」(またはその他のカテゴリー) で入力した値は、次のカテゴリーに移動しても保持されます。ウィンドウを閉じると、すべてのカテゴリーで入力した値を一括して保管するかまたは破棄するかを確認するプロンプトが出されます。
  - 修正した値を保管するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「はい」をクリックします。「ファイル」メニューまたはツールバーから「保管」>「ファイルに」をクリックする方法もあります。

## アプリケーション固有の構成プロパティの設定

アプリケーション固有の構成プロパティの場合、プロパティ名の追加または変更、値の構成、プロパティの削除、およびプロパティの暗号化が可能です。プロパティのデフォルトの長さは 255 文字です。

1. グリッドの左上端の部分で右マウス・ボタンをクリックします。ポップアップ・メニュー・バーが表示されます。プロパティを追加するときは「追加」をクリックします。子プロパティを追加するには、親の行番号で右マウス・ボタンをクリックし、「子を追加」をクリックします。
2. プロパティまたは子プロパティの値を入力します。
3. プロパティを暗号化するには、「暗号化」ボックスを選択します。
4. 78 ページの『標準コネクタ・プロパティの設定』の説明に従い、変更内容を保管するかまたは破棄するかを選択します。

各プロパティごとに表示される「更新メソッド」は、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。

**重要:** 事前設定のアプリケーション固有のコネクタ・プロパティ名を変更すると、コネクタに障害が発生する可能性があります。コネクタをアプリケーションに接続したり正常に実行したりするために、特定のプロパティ名が必要である場合があります。

### コネクタ・プロパティの暗号化

「プロパティを編集」ウィンドウの「暗号化」チェック・ボックスにチェックマークを付けると、アプリケーション固有のプロパティを暗号化することができます。値を暗号化解除するには、「暗号化」チェック・ボックスをクリックしてチェックマークを外し、「検証」ダイアログ・ボックスに正しい値を入力し、「OK」をクリックします。入力された値が正しい場合は、暗号化解除された値が表示されます。

各プロパティとそのデフォルト値のリストおよび説明は、各コネクタのアダプター・ユーザーズ・ガイドにあります。

プロパティに複数の値がある場合には、プロパティの最初の値に「暗号化」チェック・ボックスが表示されます。「暗号化」を選択すると、そのプロパティのすべての値が暗号化されます。プロパティの複数の値を暗号化解除するには、そのプロパティの最初の値の「暗号化」チェック・ボックスをクリックしてチェックマークを外してから、「検証」ダイアログ・ボックスで新規の値を入力します。入力値が一致すれば、すべての複数值が暗号化解除されます。

### 更新メソッド

付録『コネクタの標準構成プロパティ』の 52 ページの『プロパティ値の設定と更新』にある更新メソッドの説明を参照してください。

## サポートされるビジネス・オブジェクト定義の指定

コネクタで使用するビジネス・オブジェクトを指定するには、Connector Configurator の「サポートされているビジネス・オブジェクト」タブを使用します。

汎用ビジネス・オブジェクトと、アプリケーション固有のビジネス・オブジェクトの両方を指定する必要があり、またそれらのビジネス・オブジェクト間のマップの関連を指定することが必要です。

**注:** コネクタによっては、アプリケーションでイベント通知や (メタオブジェクトを使用した) 追加の構成を実行するために、特定のビジネス・オブジェクトをサポートされているものとして指定することが必要な場合もあります。詳細は、「コネクタ開発ガイド (C++ 用)」または「コネクタ開発ガイド (Java 用)」を参照してください。

## ご使用のブローカーが ICS の場合

ビジネス・オブジェクト定義がコネクタでサポートされることを指定する場合や、既存のビジネス・オブジェクト定義のサポート設定を変更する場合は、「サポートされているビジネス・オブジェクト」タブをクリックし、以下のフィールドを使用してください。

**ビジネス・オブジェクト名:** ビジネス・オブジェクト定義がコネクタによってサポートされることを指定するには、System Manager を実行し、以下の手順を実行します。

1. 「ビジネス・オブジェクト名」リストで空のフィールドをクリックします。  
System Manager プロジェクトに存在するすべてのビジネス・オブジェクト定義を示すドロップダウン・リストが表示されます。
2. 追加するビジネス・オブジェクトをクリックします。
3. ビジネス・オブジェクトの「エージェント・サポート」(以下で説明) を設定します。
4. 「Connector Configurator」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。追加したビジネス・オブジェクト定義に指定されたサポートを含む、変更されたコネクタ定義が、System Manager のプロジェクトに保管されます。

サポートされるリストからビジネス・オブジェクトを削除する場合は、以下の手順を実行します。

1. ビジネス・オブジェクト・フィールドを選択するため、そのビジネス・オブジェクトの左側の番号をクリックします。
2. 「Connector Configurator」ウィンドウの「編集」メニューから、「行を削除」をクリックします。リスト表示からビジネス・オブジェクトが除去されます。
3. 「ファイル」メニューから、「プロジェクトの保管」をクリックします。

サポートされるリストからビジネス・オブジェクトを削除すると、コネクタ定義が変更され、削除されたビジネス・オブジェクトはコネクタのこのインプリメンテーションで使用不可になります。コネクタのコードに影響したり、そのビジネス・オブジェクト定義そのものが System Manager から削除されることはありません。

**エージェント・サポート:** ビジネス・オブジェクトがエージェント・サポートを備えている場合、システムは、コネクタ・エージェントを介してアプリケーションにデータを配布する際にそのビジネス・オブジェクトの使用を試みます。



一般に、コネクターのアプリケーション固有ビジネス・オブジェクトは、そのコネクターのエージェントによってサポートされますが、汎用ビジネス・オブジェクトはサポートされません。

ビジネス・オブジェクトがコネクター・エージェントによってサポートされるよう指定するには、「エージェント・サポート」ボックスにチェックマークを付けます。「Connector Configurator」ウィンドウでは「エージェント・サポート」の選択の妥当性は検査されません。

**最大トランザクション・レベル:** コネクターの最大トランザクション・レベルは、そのコネクターがサポートする最大のトランザクション・レベルです。

ほとんどのコネクターの場合、選択可能な項目は「最大限の努力」のみです。

トランザクション・レベルの変更を有効にするには、サーバーを再始動する必要があります。

### ご使用のブローカーが WebSphere Message Broker の場合

スタンドアロン・モードで作業している (System Manager に接続していない) 場合、手動でビジネス名を入力する必要があります。

System Manager を実行している場合、「サポートされているビジネス・オブジェクト」タブの「ビジネス・オブジェクト名」列の下にある空のボックスを選択できます。コンボ・ボックスが表示され、コネクターが属する統合コンポーネント・ライブラリー・プロジェクトから選択可能なビジネス・オブジェクトのリストが示されます。このリストから必要なビジネス・オブジェクトを選択します。

「メッセージ・セット ID」は、WebSphere Business Integration Message Broker 5.0 のオプションのフィールドです。この ID が提供される場合、一意である必要はありません。ただし、WebSphere MQ Integrator および Integrator Broker 2.1 の場合は、一意の ID を提供する必要があります。

### ご使用のブローカーが WAS の場合

使用するブローカー・タイプとして WebSphere Application Server を選択した場合、Connector Configurator にメッセージ・セット ID は必要ありません。「サポートされているビジネス・オブジェクト」タブには、サポートされるビジネス・オブジェクトの「ビジネス・オブジェクト名」列のみが表示されます。

スタンドアロン・モードで作業している (System Manager に接続していない) 場合、手動でビジネス・オブジェクト名を入力する必要があります。

System Manager を実行している場合、「サポートされているビジネス・オブジェクト」タブの「ビジネス・オブジェクト名」列の下にある空のボックスを選択できます。コンボ・ボックスが表示され、コネクターが属する統合コンポーネント・ライブラリー・プロジェクトから選択可能なビジネス・オブジェクトのリストが示されます。このリストから必要なビジネス・オブジェクトを選択します。

## 関係付けられたマップ (ICS のみ)

各コネクタは、現在 WebSphere InterChange Server でアクティブなビジネス・オブジェクト定義、およびそれらの関連付けられたマップのリストをサポートします。このリストは、「**関連付けられたマップ**」タブを選択すると表示されます。

ビジネス・オブジェクトのリストには、エージェントでサポートされるアプリケーション固有のビジネス・オブジェクトと、コントローラーがサブスクライブ・コラボレーションに送信する、対応する汎用オブジェクトが含まれます。マップの関連によって、アプリケーション固有のビジネス・オブジェクトを汎用ビジネス・オブジェクトに変換したり、汎用ビジネス・オブジェクトをアプリケーション固有のビジネス・オブジェクトに変換したりするときに、どのマップを使用するかが決定されます。

特定のソースおよび宛先ビジネス・オブジェクトについて一意的に定義されたマップを使用する場合、表示を開くと、マップは常にそれらの該当するビジネス・オブジェクトに関連付けられます。ユーザーがそれらを変更する必要はありません (変更できません)。

サポートされるビジネス・オブジェクトで使用可能なマップが複数ある場合は、そのビジネス・オブジェクトを、使用する必要のあるマップに明示的にバインドすることが必要になります。

「**関連付けられたマップ**」タブには以下のフィールドが表示されます。

- **ビジネス・オブジェクト名**

これらは、「**サポートされているビジネス・オブジェクト**」タブで指定した、このコネクタでサポートされるビジネス・オブジェクトです。「**サポートされているビジネス・オブジェクト**」タブでビジネス・オブジェクトを追加指定した場合、その内容は、「Connector Configurator」ウィンドウの「**ファイル**」メニューから「**プロジェクトに保管**」を選択して、変更を保管した後に、このリストに反映されます。

- **関連付けられたマップ**

この表示には、コネクタの、サポートされるビジネス・オブジェクトでの使用のためにシステムにインストールされたすべてのマップが示されます。各マップのソース・ビジネス・オブジェクトは、「**ビジネス・オブジェクト名**」表示でマップ名の左側に表示されます。

- **明示的**

場合によっては、関連付けられたマップを明示的にバインドすることが必要になります。

明示的バインディングが必要なのは、特定のサポートされるビジネス・オブジェクトに複数のマップが存在する場合のみです。ICS は、ブート時、各コネクタでサポートされるそれぞれのビジネス・オブジェクトにマップを自動的にバインドしようとします。複数のマップでその入力データとして同一のビジネス・オブジェクトが使用されている場合、サーバーは、他のマップのスーパーセットである 1 つのマップを見つけて、バインドしようとします。

他のマップのスーパーセットであるマップがないと、サーバーは、ビジネス・オブジェクトを単一のマップにバインドすることができないため、バインディングを明示的に設定することが必要になります。

以下の手順を実行して、マップを明示的にバインドします。

1. 「明示的 (Explicit)」列で、バインドするマップのチェック・ボックスにチェックマークを付けます。
2. ビジネス・オブジェクトに関連付けるマップを選択します。
3. 「Connector Configurator」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。
4. プロジェクトを ICS に配置します。
5. 変更を有効にするため、サーバーをリブートします。

## リソース (ICS)

「リソース」タブでは、コネクター・エージェントが、コネクター・エージェント並列処理を使用して同時に複数のプロセスを処理するかどうか、またどの程度処理するかを決定する値を設定できます。

すべてのコネクターがこの機能をサポートしているわけではありません。複数のプロセスを使用するよりも複数のスレッドを使用する方が通常は効率的であるため、Java でマルチスレッドとして設計されたコネクター・エージェントを実行している場合、この機能を使用することはお勧めできません。

## メッセージング (ICS)

メッセージング・プロパティは、DeliveryTransport 標準プロパティの値として MQ を設定し、ブローカー・タイプとして ICS を設定した場合にのみ、使用可能です。これらのプロパティは、コネクターによるキューの使用方法に影響します。

## トレース/ログ・ファイル値の設定

コネクター構成ファイルまたはコネクター定義ファイルを開くと、Connector Configurator は、そのファイルのログおよびトレースの値をデフォルト値として使用します。Connector Configurator 内でこれらの値を変更できます。

ログとトレースの値を変更するには、以下の手順を実行します。

1. 「トレース/ログ・ファイル」タブをクリックします。
2. ログとトレースのどちらでも、以下のいずれかまたは両方へのメッセージの書き込みを選択できます。
  - コンソールに (STDOUT):  
ログ・メッセージまたはトレース・メッセージを STDOUT ディスプレイに書き込みます。

**注:** STDOUT オプションは、Windows プラットフォームで実行しているコネクターの「トレース/ログ・ファイル」タブでのみ使用できます。

- ファイルに:  
ログ・メッセージまたはトレース・メッセージを指定されたファイルに書き込みます。ファイルを指定するには、ディレクトリー・ボタン (省略符号) をクリックし、指定する格納場所へ移動し、ファイル名を指定し、「保管」をクリックします。ログ・メッセージまたはトレース・メッセージは、指定した場所の指定したファイルに書き込まれます。

**注:** ログ・ファイルとトレース・ファイルはどちらも単純なテキスト・ファイルです。任意のファイル拡張子を使用してこれらのファイル名を設定できます。ただし、トレース・ファイルの場合、拡張子として `.trc` ではなく `.trace` を使用することをお勧めします。これは、システム内に存在する可能性がある他のファイルとの混同を避けるためです。ログ・ファイルの場合、通常使用されるファイル拡張子は `.log` および `.txt` です。

## データ・ハンドラー

データ・ハンドラー・セクションの構成が使用可能となるのは、`DeliveryTransport` の値に `JMS` を、また `ContainerManagedEvents` の値に `JMS` を指定した場合のみです。すべてのアダプターでデータ・ハンドラーを使用できるわけではありません。

これらのプロパティーに使用する値については、付録 A の『コネクターの標準構成プロパティー』の `ContainerManagedEvents` の下の説明を参照してください。その他の詳細は、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

---

## 構成ファイルの保管

コネクターの構成が完了したら、コネクター構成ファイルを保管します。Connector Configurator では、構成中に選択したブローカー・モードでファイルを保管します。Connector Configurator のタイトル・バーには現在のブローカー・モード (ICS、WMQI、または WAS) が常に表示されます。

ファイルは XML 文書として保管されます。XML 文書は次の 3 通りの方法で保管できます。

- System Manager から、統合コンポーネント・ライブラリーに `*.con` 拡張子付きファイルとして保管します。
- 指定したディレクトリーに保管します。
- スタンドアロン・モードで、ディレクトリー・フォルダーに `*.cfg` 拡張子付きファイルとして保管します。

System Manager でのプロジェクトの使用法、および配置の詳細については、以下のインプリメンテーション・ガイドを参照してください。

- ICS: 「*WebSphere InterChange Server* インプリメンテーション・ガイド」
- WebSphere Message Brokers: 「*WebSphere Message Brokers* 使用アダプター・インプリメンテーション・ガイド」
- WAS: 「アダプター実装ガイド (*WebSphere Application Server*)」

---

## 構成ファイルの変更

既存の構成ファイルの統合ブローカー設定を変更できます。これにより、他のブローカーで使用する構成ファイルを新規に作成するときに、このファイルをテンプレートとして使用できます。

**注:** 統合ブローカーを切り替える場合には、ブローカー・モード・プロパティーと同様に他の構成プロパティーも変更する必要があります。

既存の構成ファイルでのブローカーの選択を変更するには、以下の手順を実行します (オプション)。

- Connector Configurator で既存の構成ファイルを開きます。
- 「標準のプロパティ」タブを選択します。
- 「標準のプロパティ」タブの「**BrokerType**」フィールドで、ご使用のブローカーに合った値を選択します。  
現行値を変更すると、プロパティ画面の利用可能なタブおよびフィールド選択がただちに更新され、選択した新規ブローカーに適したタブとフィールドのみが表示されます。

---

## 構成の完了

コネクタの構成ファイルを作成し、そのファイルを変更した後で、コネクタの始動時にコネクタが構成ファイルの位置を特定できるかどうかを確認してください。

これを行うには、コネクタが使用する始動ファイルを開き、コネクタ構成ファイルに使用されている格納場所とファイル名が、ファイルに対して指定した名前およびファイルを格納したディレクトリまたはパスと正確に一致しているかどうかを検証します。

---

## グローバル化環境における Connector Configurator の使用

Connector Configurator はグローバル化され、構成ファイルと統合ブローカー間の文字変換を処理できます。Connector Configurator では、ネイティブなエンコード方式を使用しています。構成ファイルに書き込む場合は UTF-8 エンコード方式を使用します。

Connector Configurator は、以下の場所で英語以外の文字をサポートします。

- すべての値のフィールド
- ログ・ファイルおよびトレース・ファイル・パス (「**トレース/ログ・ファイル**」タブで指定)

CharacterEncoding および Locale 標準構成プロパティのドロップ・リストに表示されるのは、サポートされる値の一部のみです。ドロップ・リストに、サポートされる他の値を追加するには、製品ディレクトリの `¥Data¥Std¥stdConnProps.xml` ファイルを手動で変更する必要があります。

例えば、Locale プロパティの値のリストにロケール `en_GB` を追加するには、`stdConnProps.xml` ファイルを開き、以下に太文字で示した行を追加してください。

```
<Property name="Locale"
isRequired="true"
updateMethod="component restart">
  <ValidType>String</ValidType>
  <ValidValues>
    <Value>ja_JP</Value>
    <Value>ko_KR</Value>
    <Value>zh_CN</Value>
    <Value>zh_TW</Value>
    <Value>fr_FR</Value>
    <Value>de_DE</Value>
    <Value>it_IT</Value>
```

```
<Value>es_ES</Value>
<Value>pt_BR</Value>
<Value>en_US</Value>
<Value>en_GB</Value>
    <DefaultValue>en_US</DefaultValue>
</ValidValues>
</Property>
```

---

## 付録 C. 統合チャネルのマイグレーション

この付録では、Ariba Buyer 7.1a から Ariba Buyer 8.1 へ統合チャネルをマイグレーションする方法について説明します。

Ariba Buyer 8.0 から Ariba Buyer 8.1 へとアップグレードする場合は、Ariba Buyer の資料に従って、単に Buyer アプリケーションをマイグレーションしてください。統合チャネルのマイグレーションは必要ありません。

---

### 概要

Ariba Buyer 7.1a から Ariba Buyer 8.1 へのマイグレーションは、2 つのプロセスから成ります。以下の作業を実行する必要があります。

1. Ariba Buyer アプリケーションをバージョン 7.1a からバージョン 8.1 にマイグレーションします。マイグレーションの実行方法に関する説明は、「*Ariba Buyer Migration Guide Version 8.1*」を参照してください。
2. 統合チャネルを前のバージョンのアダプターから現行バージョンにマイグレーションします。

---

### マイグレーション・プロセス

この付録では、統合チャネルを WebSphere Business Integration Adapter for Ariba Buyer にマイグレーションするためのステップについて説明します。ステップは以下の段階に分かれています。

- アダプターのインストール
- Ariba Buyer のマイグレーション
- 統合チャネルのマイグレーション

---

### WebSphere Business Integration Adapter for Ariba Buyer のインストール

IBM では、このバージョンのアダプターをインストールする前に、前のバージョンをバックアップおよびアンインストールすることをお勧めします。このセクションでは、以下の作業の実行方法について説明します。

- 前のバージョン (WAAB) をアンインストール。
- WebSphere Business Integration Adapter for Ariba Buyer をインストール。
- WebSphere Business Integration Adapter マイグレーション・ユーティリティーをインストール。

### WAAB のアンインストール

IBM では、WAAB インスタンスをバックアップすることをお勧めします。バックアップするには、アダプターをアンインストールする前に、アダプターのインストール・ディレクトリー全体をアーカイブ・ロケーションにコピーします。



アダプターのインストール・ディレクトリーをバックアップしたら、「*WebSphere Adapter for Ariba Buyer User Guide*」のアダプターのアンインストールに関する説明に従ってください。

**注:** 後のマイグレーション・プロセスで、カスタマイズした CSV マッピングを MQCSVMapping.xml ファイルから新規の Ariba IntegrationMappings.xml ファイルへ手動でマイグレーションする必要があるため、このファイルのロケーションを覚えておくことが重要です。

## WebSphere Business Integration Adapter for Ariba Buyer のインストール

新規アダプターをインストールするには、本書の 9 ページの『第 2 章 アダプターのインストール』を参照してください。

**注:** 9 ページの『ブローカーの互換性』のセクションに来たら作業を停止してください。このマイグレーションでは Ariba Buyer Configure ツールを実行する必要はありません。

## アダプター・マイグレーション・ユーティリティーの構成

IBM では、WAAB から WebSphere Business Integration Adapter for Ariba Buyer へのマイグレーションを支援するためのユーティリティーを提供しています。マイグレーション・ユーティリティーは、アダプターのインストールの一部として含まれています。

マイグレーション・ユーティリティーを構成するには、以下のステップを実行します。

1. <Ariba Buyer 8 Server dir>/classes/classpath.txt ファイルを変更します。変更するには以下の行を追加します。

```
classes/BIA_IBMMigration.jar
```

2. ご使用の Buyer 8.1 インストール・ディレクトリー中に以下のディレクトリーを作成します。

```
¥Server¥channels¥MQSeries
```

マイグレーション作業一式を正常に実行した後、このディレクトリーを削除して構いません。

ユーティリティーのインストールが完了しました。以下で説明する統合チャネル・マイグレーションのタスクの一部としてユーティリティーを実行します。

表 23 は、マイグレーション・ユーティリティー・パッケージに含まれているファイルのリストと、それぞれのファイルに関する簡単な説明を示しています。

表 23. マイグレーション・ユーティリティー・パッケージの内容

サブディレクトリー	説明
¥channels¥IBM¥migration¥WBIAABMigrationTaskList.xml	アダプター・マイグレーションの一部として実行する必要のある作業のリストを示したファイル。このファイルはマイグレーション・ハーネスへの入力データとして使用されます。



表 23. マイグレーション・ユーティリティー・パッケージの内容 (続き)

サブディレクトリー	説明
¥channels¥IBM¥migration¥ UpdateIBMParameters.txt	Parameters.table ファイルに対して実行する必要がある変更について説明したファイル。
¥channels¥IBM¥migration¥ loaddb_removalist.txt	config¥LoadDB.txt ファイルから除去する必要がある統合イベントのリスト。これらのイベントは新規アダプターでは必要ありません。
¥classes¥BIA_IBMMigration.jar	アダプターのマイグレーションに必要なすべてのコードが含まれたファイル。

## Ariba Buyer のマイグレーション

「*Ariba Buyer Migration Guide Version 8.1*」の指示に従って、Ariba Buyer 8.1 へのマイグレーションを実行してください。

migrationHarness ユーティリティーを起動した後、0 から 26 のタスクのみを実行してください。実行するには、次のコマンドを発行します。

```
rr 0-26
```

ほかの Ariba Buyer 統合タスクは、統合チャネル・マイグレーション・ユーティリティーの実行後に行います (92 ページの『Buyer マイグレーション作業の完了』を参照)。

## 統合チャネルのマイグレーション

アダプターをインストールし、Ariba Buyer を正常にマイグレーションした後、一連の統合チャネル・マイグレーション作業を実行する必要があります。以下の作業を実行します。

1. アダプター・マイグレーション・ユーティリティーの実行
2. 統合イベントのマイグレーション
3. アダプターと統合チャネルの構成
4. メタデータの生成
5. オブジェクト・マッピングの作成

### マイグレーション・ユーティリティーの実行

アダプター・マイグレーション・ユーティリティーは、以下のファイル中の統合チャネル・マイグレーションに対して、いくつか修正を行います。

- <Ariba Buyer 8 Server>¥config¥Parameters.table ファイル
- <Ariba Buyer 8 Server>¥config¥LoadDB.txt ファイル
- <Ariba Buyer 8 Server>¥..¥<variantName>¥..  
¥<partitionName>¥IBM¥MessageConfiguration.table ファイル

#### ユーティリティーの実行

ユーティリティーを実行するには、以下のコマンドを <Ariba Buyer 8 Server>¥bin ディレクトリーから実行します。

```
migrationHarness -tasklist <Ariba Buyer 8 Server dir>
¥channels¥IBM¥migration¥WBIAABMigrationTaskList.xml
```

例えば、次のようになります。

```
migrationHarness -tasklist
C:¥Ariba¥app¥Buyer8¥Server¥channels¥IBM¥migration¥WBIAABMigrationTaskList.xml
```

マイグレーション・ユーティリティーが実行する操作は次のとおりです。

### バックアップの作成

ユーティリティーは、<Ariba Buyer 8 Server>¥config ディレクトリーで削除または変更するすべてのファイルのバックアップを作成します。バックアップは <Ariba Buyer 8 Server>¥channels¥IBM¥migration¥backups¥backup<timestamp> にあります。

### Parameters.table ファイルの更新

ユーティリティーは Parameters.table ファイルに対して以下のタスクを実行します。

- System.Messaging.Channels.MQSeries セクションを System.Messaging.Channels.IBM セクションに置換
- それぞれの非 CSV 区画ごとの記入項目を Partitions.<Partition Name>.Application.Messaging.MessageConfigurationFiles.IBM に追加
- System.Logging.Console.Categories および System.Logging.MainLogFile.Categories によって指定されたロギング・カテゴリー名を、“integrationVendor\_IBM” から “integrationChannel\_IBM” に変更

### LoadDB.txt ファイルの更新

ユーティリティーは LoadDB.txt ファイルに対して以下のタスクを実行します。

- None.IntegrationEvent.MQMetaDataExport への参照をすべて除去
- None.IntegrationEvent.MQSchemaDataExport への参照をすべて除去
- None.IntegrationEvent.MQXMLDataExport への参照をすべて除去

### MessageConfiguration.table ファイルの更新

ユーティリティーは MessageConfiguration.table ファイルに対して以下のタスクを実行します。

- ¥config¥variants¥<バリエーション名>¥partitions¥<区画名>¥file 中に IBMMessageConfiguration.table という名前のファイルを作成します。このファイルにはファイル・チャンネルにマイグレーションされる必要のある CSV イベントが含まれています。このプロセスについては、92 ページの『CSV イベントの File Channel への転送』を参照してください。
- <event name>.Channel.Name パラメーターを “MQSeries” から “IBM” に変更
- <event name>.Channel.Action パラメーターを <event name>.Channel.Operation に変更

**注:** LoadAndDelete は “Load And Delete” になり、UpdateAndDelete は “Update And Delete” になります。

- イベントがデータ・プッシュまたはデータ・プルの場合は、ユーティリティーは `<event name>.Channel.Verb` に対して新規のパラメーターを作成し、以下のように割り当てます。
  - データ・プッシュの場合、`<event name>.Channel.Verb = Create` を割り当てます。
  - データ・プルの場合、`<event name>.Channel.Verb = Retrieve` を割り当てます。

表 24 は、マイグレーション・ユーティリティーが `MessageConfiguration` ファイルに対して行う変更の要約を示しています。

表 24. ERP 統合イベントのチャンネル・セクションの値

パラメーター・タグ	古い値	新規の値
Name	MQSeries	IBM
Operation	パラメーター・タグは存在しません。	Action パラメーターの値が割り当てられます。
Verb	パラメーター・タグは存在しません。	イベントがデータ・プッシュの場合は、新規の値は <code>Create</code> になります。 イベントがデータ・プルの場合は、新規の値は <code>Retrieve</code> になります。 サブスクライブの場合は、パラメーターは作成されません。
Timeout	パラメーター・タグは存在しません。	イベントが非同期のデータ・プッシュの場合は、タイムアウトを戻すまでの待機時間 (ミリ秒) に値を設定します。

### サブスクライブ・イベントの場合の `MessageConfiguration.table` ファイルに対する変更

すべてのサブスクライブ・イベントは `MessageConfiguration.table` イベントにバリエーション・レベルで定義する必要があります。例えば、次のようになります。

```
<Ariba Buyer 8 Server>%config%
variant%<Variantname>%IBM%MessageConfiguration.table
```

それぞれのサブスクライブ・イベントごとに、`EventSource` の新規のチャンネル・パラメーターを手動で作成して、パラメーターの値を指定する必要があります (以下の例を参照)。

```
CostCenterSubscribe = {
  Channel = {
    EventSource = "myPart1_CostCenterSubscribe"
  }
  LoggingName = CostCenter;
  TopicName = CostCenterSubscribe;
};
```

`EventSource` パラメーターとその使用法の詳細記述については、「*Ariba Configuration Reference Guide*」を参照してください。

## Buyer マイグレーション作業の完了

統合チャンネル・マイグレーション・ユーティリティーが正常に作業を終了した後、migrationHarness コマンドを使用してマイグレーション一式を実行します。次に、以下のコマンドを発行して残りの Ariba Buyer マイグレーション作業を完了します。

rr 27-29

## 統合イベントのマイグレーション

統合イベントを定義するファイルは、Ariba Buyer 8.x と Ariba Buyer 7.x で異なります (表 25 を参照)。

表 25. 統合イベント定義ファイル

Ariba Buyer バージョン	統合イベントを定義するファイル
Ariba Buyer 7.x	<ul style="list-style-type: none"><li>IntegrationEvents.table</li><li>IntegrationSchema.table</li></ul>
Ariba Buyer 8.x	<ul style="list-style-type: none"><li>MessageConfiguration.table 区画ごとに複数のテーブルが存在します。 (定義された統合チャンネルごとに 1 つずつ)。</li><li>MessageDefinition.table バリエーションごとに 1 つのテーブルが存在します。</li></ul>

MessageDefinition.table ファイルと MessageConfiguration.table ファイルは、Ariba Buyer をバージョン 7.x からバージョン 8.x にマイグレーションする時に作成されます (89 ページの『Ariba Buyer のマイグレーション』を参照)。

MessageDefinition.table ファイルは完全なファイルです。一方、MessageConfiguration.table ファイルにはいくつかの修正が必要です。以下の修正を行います。

- CSV イベント定義を File Channel に転送
- CSV マッピングを File Channel 形式に変換

上記のそれぞれの作業については、以下で説明します。

### CSV イベントの File Channel への転送

Ariba Buyer 7.x では、WAAB バージョンのアダプターは CSVAdapter というコンポーネントを使用して CSV ファイル・データをロードしていました。Ariba Buyer 8.x では、CSV ファイル・データは Ariba Buyer File Channel アダプターによって排他的にロードされます。

したがって、古い IntegrationEvents.table ファイルに対して行われたカスタマイズ (表 25 を参照) は、新規の MessageConfiguration.table ファイルに自動的にマイグレーションされません。以下の作業を実行する必要があります。

1. カスタマイズを以下の場所にある MessageConfiguration.table ファイルに再適用します。

```
<Ariba Buyer 8 dir>%config%variants%<variant name>%partitions%<partition name>%file%MessageConfiguration.table
```

2. 以下の場所にある CSV イベント定義すべてを除去します。

<Ariba Buyer 8 dir>¥config¥variants¥<variant name>¥partitions¥<partition name>¥IBM¥MessageConfiguration.table

すべてのカスタマイズが正しく適用されたことを確認するため、以下のステップを実行します。

1. それぞれの CSV イベント定義ごとに、以下にある記入項目を比較します。  
<Ariba Buyer 8 dir>¥..¥<partition name>¥file¥MessageConfiguration.table および  
<Ariba Buyer 8 dir>¥..¥<partition name>¥file¥IBMMessageConfiguration.table
2. チャンネル・セクション内の相違点以外に相違点がない場合は、<partition name>¥file¥IBMMessageConfiguration.table から記入項目を除去します。
3. チャンネル・セクション内の相違点以外に相違点がある場合 (ファイル名やアクションなど) は、それに応じた変更を <partition name>¥file¥MessageConfiguration.table で行い、<partition name>¥file¥IBMMessageConfiguration.table から記入項目を除去します。
4. <partition name>¥file¥IBMMessageConfiguration.table に加えた変更がすべて <partition name>¥file¥MessageConfiguration.table に転送されたことを確認したら、<partition name>¥file¥IBMMessageConfiguration.table ファイルは不要になったためバックアップ・ロケーションに戻します。

**注:** LoadAndDelete という WAAB アクションは、WebSphere Business Integration Adapter for Ariba Buyer の “Load And Delete” オペレーションと等価です。また、UpdateAndDelete というアクションは “Update And Delete” オペレーションと等価です。

## CSV マッピングの File Channel 形式への変換

Ariba Buyer 8.x では、すべての創造的 CSV 統合イベントのためのマッピングを提供します。

WAAB CSVAdapter で使用されるマッピング・ファイルがカスタマイズされていない場合は、次のセクションに進んでください。

マッピングが変更されたり、WAAB CSVAdapter で使用されるマッピング・ファイルに追加されている場合は、同じ変更と追加を Ariba Buyer File Channel で使用されるマッピング・ファイルに対しても行う必要があります。

マッピング・ファイル IntegrationMappings.aml の形式の詳細については、「Ariba Buyer 8.x Customization Guide」の第 4 章を参照してください。

## アダプターと統合チャネルの構成

WebSphere Business Integration Adapter for Ariba Buyer は 2 つの主要コンポーネントである統合チャネルとコネクタで構成されます。各コンポーネントは独自のプロセスで実行されます。2 つのプロセスは RMI (リモート・メソッド呼び出し) を使用して相互通信します。

2 つのコンポーネント間の通信を構成するには、それぞれのコンポーネントにパラメーターを設定する必要があります。パラメーターの正しい設定方法については、本書の 13 ページの『第 3 章 アダプターの構成』および 16 ページの『統合チャネルの構成』を参照してください。

## メタデータの生成

WebSphere Business Integration Adapter for Ariba Buyer はメタデータ主導型です。つまり、再コード化や再コンパイルを行わなくても、アダプターを通過するビジネス・オブジェクトの構造に対する変更をアダプター自体が処理することができます。ビジネス・オブジェクトの構造はメタデータとして保管されます。

メタデータは常時ビジネス・オブジェクト構造と正確に一致している必要があります。ビジネス・オブジェクトの構造が変更された場合は、必ずそのビジネス・オブジェクトについて新規のメタデータを生成する必要があります。

必要なすべてのメタデータを生成するには、以下のステップに従ってください。

1. <Ariba Buyer 8 Server>¥bin ディレクトリーから、以下のコマンドを実行します。

```
initdb -reshapedb
```

2. Ariba Buyer 8.x を始動します。
3. Ariba Buyer. 8.x を停止します。

この手順によって、Ariba Buyer がメタデータをファイル・システム・リポジトリーに出力します。このリポジトリーのロケーションは `Parameters.table` の `MetadataRepos` パラメーターで指定されます。

メタデータが生成されたことを検証するには、`MetadataRepos` ディレクトリーを検査します。このディレクトリーにはそれぞれのバリエーションごとにフォルダーが含まれており、フォルダー内にはバリエーションで定義されたそれぞれの Ariba Buyer オブジェクトごとにファイルが含まれています。また、バリエーション内のそれぞれの区画ごとのフォルダーも含まれています。このフォルダーには区画のすべての `MessageConfiguration.table` 記入項目が含まれています。

次のステップでは、ファイル・システム・リポジトリーに保管されたメタデータからビジネス・オブジェクト定義を作成します。詳細については、本書の 33 ページの『第 5 章 ビジネス・オブジェクト定義の生成』を参照してください。

## オブジェクト・マッピングの作成

ビジネス・オブジェクト定義を作成したら、それを使用してブローカーのオブジェクト・マッピングを構成できます。オブジェクト・マッピングは、現在カスタマー・アダプターまたは既存のブローカー・フローによって行われているマッピングに置き換わるものです。

WebSphere InterChange Server に対するコラボレーションの作成方法の詳細については、「*WebSphere Business Integration Server* コラボレーション開発ガイド」を参照してください。

WebSphere Message Brokers の MQ フローの作成方法については、「*WebSphere Message Brokers* 使用アダプター・インプリメンテーション・ガイド」を参照してください。

WebSphere Application Server の MQ フローの作成方法については、「アダプター実装ガイド (*WebSphere Application Server*)」を参照してください。



---

## 付録 D. 特記事項

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。

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
Licensing

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せんが、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認できません。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。

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。

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。

---

## 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。

汎用プログラミング・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。

**警告:** 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、IBM Corporation の商標です。

IBM  
IBM ロゴ  
AIX  
CrossWorlds  
DB2  
DB2 Universal Database  
Domino  
Lotus  
Lotus Notes  
MQIntegrator  
MQSeries  
Tivoli  
WebSphere

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。

MMX、Pentium および ProShare は、Intel Corporation の米国およびその他の国における商標です。

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または登録商標です。

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。



IBM WebSphere Business Integration Adapter Framework V2.4.0