

**IBM WebSphere Business Integration
Adapters**



**Adapter for WebSphere Business
Integration Message Broker
ユーザース・ガイド**

V 2.6.x

**IBM WebSphere Business Integration
Adapters**



**Adapter for WebSphere Business
Integration Message Broker
ユーザース・ガイド**

V 2.6.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、111ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM WebSphere Business Integration Adapter for WebSphere Integration Message Broker (5724-H37) バージョン 2.6.x 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 WebSphere Business Integration Message Broker User Guide
V 2.6.x

発 行： 日本アイ・ピー・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 2004.7

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 2000, 2004. All rights reserved.

© Copyright IBM Japan 2004

目次

本書について	v
対象読者	v
本書の前提条件	v
関連文書	v
表記上の規則	vi
本リリースの新機能	ix
リリース 2.6.x の新機能	ix
リリース 2.5.x での新機能	ix
リリース 2.4.x の新機能	ix
リリース 2.3.x の新機能	ix
リリース 2.2.x の新機能	x
リリース 2.1.x の新機能	x
リリース 1.5.x の新機能	xi
リリース 1.4.x の新機能	xi
リリース 1.3.x の新機能	xi
第 1 章 概要	1
作業ロードマップ	1
用語	1
アダプター環境の概要	3
アダプターでの処理の概要	4
第 2 章 コネクターのインストール	13
インストール作業の概要	13
アダプター環境	13
アダプターと関連ファイルのインストール	15
インストールの検証	16
第 3 章 コネクターの構成	19
コネクターの構成の概要	19
Connector Configurator の概要	20
Connector Configurator の始動	21
System Manager からの Configurator の実行	22
コネクター固有のプロパティ・テンプレートの作成	22
新規構成ファイルの作成	25
既存ファイルの使用	26
構成ファイルの完成	28
構成ファイル・プロパティの設定	28
構成ファイルの保管	35
構成ファイルの変更	36
構成の完了	36
グローバル化環境における Connector Configurator の使用	37
複数コネクター・インスタンスの作成	37
構成例の検証	39
キューの Uniform Resource Identifier (URI) の設定	39
イベント・デリバリーの保証	41
第 4 章 コネクターの実行	43
コネクターの実行の概要	43

始動ファイルの構成	43
コネクターの始動	44
コネクターの停止	45
エラー処理の概要	46
トレースの概要	48
第 5 章 オブジェクトの作成	49
オブジェクトの作成の概要	49
ビジネス・オブジェクトの作成	50
ビジネス・オブジェクトの変更	51
メタオブジェクトの作成	53
第 6 章 データ・ハンドラーの構成	67
データ・ハンドラーの構成の概要	67
データ・ハンドラーの指定	67
メッセージ・フローの変更	68
第 7 章 トラブルシューティング	71
始動時の問題のトラブルシューティング	71
イベント処理のトラブルシューティング	72
サポートを受ける方法	72
付録 A. コネクターの標準構成プロパティ	73
新規プロパティと削除されたプロパティ	73
標準コネクター・プロパティの構成	73
標準プロパティの要約	75
標準構成プロパティ	80
付録 B. Adapter for WBI Message Broker のコネクター固有のプロパティ	93
コネクター固有のプロパティの概要	93
付録 C. チュートリアル	99
チュートリアルの概要	99
始める前に	100
環境のセットアップ	101
シナリオの実行	104
索引	107
特記事項	111
プログラミング・インターフェース情報	113
商標	113

本書について

IBM(R) WebSphere(R) Business Integration Adapter ポートフォリオは、優れた e-business テクノロジー、エンタープライズ・アプリケーション、レガシー・システム、およびメインフレーム・システムへの統合コネクティビティを提供します。この製品セットには、ビジネス・プロセス統合の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やテンプレートが含まれています。

本書では、Adapter for WebSphere Integration Message Broker のインストール、構成、およびビジネス・オブジェクト開発について説明します。

対象読者

本書は、お客様のサイトで WebSphere Business Integration 製品のサポートおよび管理を担当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。

本書の前提条件

本書の読者は、WebSphere Business Integration システム、ビジネス・オブジェクトとコラボレーションの開発、および WebSphere Integration Message Broker アプリケーションについて十分な知識と経験を持っている必要があります。

関連文書

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

本書には、以下の 2 つの資料への参照が多数含まれています。「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)」、および「WebSphere InterChange Server システム・インプリメンテーション・ガイド」本書を印刷する場合は、これらの資料も印刷すると便利です。

関連資料は以下のサイトからインストールできます。

- 一般的なアダプター情報が必要な場合、アダプターを WebSphere Message Broker (WebSphere MQ Integrator、WebSphere MQ Integrator Broker、WebSphere Business Integration Message Broker) とともに使用する場合は、およびアダプターを WebSphere Application Server とともに使用する場合は、以下のサイトを参照してください。

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

- アダプターを InterChange Server とともに使用する場合は、以下のサイトを参照してください。

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

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

- Message Brokers (WebSphere MQ Integrator Broker、WebSphere MQ Integrator、および WebSphere Business Integration Message Broker) の詳細については、以下のサイトを参照してください。

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

- WebSphere Application Server の詳細については、以下のサイトを参照してください。 <http://www.ibm.com/software/webservers/appserv/library.html>

これらのサイトでは、資料のダウンロード、インストール、および表示方法を簡単に説明しています。

注: 本書の発行後に公開されたテクニカル・サポートの技術情報や速報に、本書の対象製品に関する重要な情報が記載されている場合があります。これらの情報は、WebSphere Business Integration Support Web サイト (<http://www.ibm.com/software/integration/websphere/support/>) にあります。関心のあるコンポーネント・エリアを選択し、「Technotes」セクションと「Flashes」セクションを参照してください。また、IBM Redbooks (<http://www.redbooks.ibm.com/>) にもその他の有効な情報があることがあります。

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、記述されたとおりの値を示します。
太字	初出語を示します。
イタリック、イタリック青のアウトライン	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ことにより、参照先オブジェクトにジャンプできます。
{ }	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、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。
< >	命名規則では、不等号括弧は名前の個々の要素を囲み、各要素を区別します (例: <server_name><connector_name>tmp.log)。
/, ¥	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。すべての製品のパス名は、ご使用のシステムで製品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。

本リリースの新機能

リリース 2.6.x の新機能

2 つのコネクター固有のプロパティ `EnableMessageProducerCache` および `SessionPoolSizeForRequests` が追加されました。詳細については、93 ページの『付録 B. Adapter for WBI Message Broker のコネクター固有のプロパティ』を参照してください。

アダプターの名称が `Adapter for WebSphere MQ Integrator Broker` から `Adapter for WebSphere Integration Message Broker` へ変更されました。

バージョン 2.6.x から、アダプターは Solaris 7 でサポートされなくなりました。そのため、このプラットフォーム・バージョンに関する記述は本書から削除されました。

リリース 2.5.x での新機能

コネクターは、以下のプラットフォーム上で実行されます。

- Microsoft Windows 2000
- Solaris 7、8 または AIX 5.1、5.2 または HP UX 11.i

バージョン 2.5.0 からは、Adapter for WebSphere MQ は Microsoft Windows NT ではサポートされなくなりました。

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。この情報の新規掲載場所については 2 章を参照してください。

リリース 2.4.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、14 ページの『ブローカーの互換性』を参照してください。

コネクターは、以下のプラットフォーム上で実行されます。

- Microsoft Windows NT 4.0 Service Pack 6A または Windows 2000
- Solaris 7、8 または AIX 5.1、5.2 または HP UX 11.i

リリース 2.3.x の新機能

2003 年 3 月更新。「CrossWorlds」という名前は、現在ではシステム全体を表したり、コンポーネント名やツール名を修飾するためには使用されなくなりました。コンポーネント名およびツール名自体は、以前とほとんど変わりません。例えば、「CrossWorlds System Manager」は現在では「System Manager」となり、「CrossWorlds InterChange Server」は「WebSphere InterChange Server」となっています。

データ・ハンドラーを入力キューに関連付けることができるようになりました。詳細については、60 ページの『データ・ハンドラーの入力キューへのマッピングの概要』を参照してください。

アダプターは、WebSphere MQ Integrator Broker と WebSphere InterChange Server の統合ブローカーのほかに、WebSphere MQ Event Broker をサポートするようになりました。

保証付きイベント・デリバリー機能が拡張されました。詳細については、41 ページの『イベント・デリバリーの保証』を参照してください。

リリース 2.2.x の新機能

InProgress キューは不要になったため、使用不可にすることができます。詳しくは、97 ページの『InProgressQueue』を参照してください。

コネクターは、MQSeries 5.1、5.2、および 5.3 を介したアプリケーションとのインターオペラビリティをサポートします。詳しくは、15 ページの『アダプターの依存関係』を参照してください。

このリリースのコネクターには、ビジネス・オブジェクト処理のため UseDefaults プロパティが用意されています。詳しくは、98 ページの『UseDefaults』を参照してください。

コネクターは、データ・ハンドラーがビジネス・オブジェクトに明示的にデフォルトの動詞を割り当てない場合、デフォルトの動詞を適用できるようになりました。詳しくは、96 ページの『DefaultVerb』を参照してください。

ReplyToQueue は、ReplyToQueue コネクター・プロパティではなく、動的子メタオブジェクトを介して指示されるようになりました。詳しくは、62 ページの『JMS ヘッダーと動的子メタオブジェクトの属性』を参照してください。

メッセージ選択子を使用して識別やフィルター操作を行うほか、アダプターが特定の要求に対して応答メッセージを識別する方法を制御できます。この JMS 機能は同期要求処理にのみ適用されます。詳しくは、6 ページの『同期送達』を参照してください。

リリース 2.1.x の新機能

コネクターは国際化されました。詳しくは、1 ページの『作業ロードマップ』と 73 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。

このガイドでは、このアダプターを ICS とともに使用するための情報を提供します。

注: 保証イベント・デリバリー機能を使用するには、ICS のリリース 4.1.1.2 をインストールする必要があります。

リリース 1.5.x の新機能

IBM WebSphere Business Integration Adapter for MQ Integrator には、MQ Integrator 用のコネクタが含まれます。このアダプターは、InterChange Server (ICS) 統合ブローカーとともに動作します。統合ブローカーとは、異種のアプリケーション・セット間の統合を実行するアプリケーションです。統合ブローカーは、データ・ルーティングなどのサービスを提供します。アダプターには、以下の要素が含まれます。

- MQ Integrator に固有のアプリケーション・コンポーネント
- サンプル・ビジネス・オブジェクト
- IBM WebSphere Adapter Framework。コンポーネントは以下のとおりです。
 - コネクタ・フレームワーク
 - 開発ツール (Business Object Designer と IBM CrossWorlds System Manager を含む)
 - API (CDK を含む)

本書では、このアダプターを ICS とともに使用するための情報を提供します。

重要: コネクタは国際化に対応していないため、ISO Latin-1 データのみが処理されることが確実である場合を除いて、コネクタと ICS バージョン 4.1.1 を併用しないでください。

IBM CrossWorlds 4.1.x システムでコネクタが使用可能になりました。

リリース 1.4.x の新機能

本書の 1.4.x リリースには、欠陥を修正し IBM CrossWorlds インフラストラクチャー・リリース・バージョン 4.0.0 との互換性を保証するために、小さい変更が加えられています。

リリース 1.3.x の新機能

本書の 1.3.x リリースでは、以下の新機能と製品機能強化が追加されています。

- Create、Update、および Delete 操作を確認するための同期要求/応答処理のサポート。
- Retrieve、Retrieve By Content、および Exist 操作のサポート。
- 正常に処理されたメッセージ、アンサブスクライブされたメッセージ、およびエラーが発生したメッセージを含む、メッセージの全アーカイブ。
- 同じメッセージ・フォーマットを複数のビジネス・オブジェクトに割り当てる拡張機能。
- コネクタは、完全修飾 URI なしでローカル・キューを識別することが可能になりました。したがって、InputQueueURI、InProgressQueueURI、UnsubscribedURI、および ErrorQueueURI の各コネクタ・プロパティには、「URI」サフィックスが含まれなくなりました。
- コネクタ・メタオブジェクトにデフォルトの変換プロパティが追加されました。したがって、コネクタ・プロパティ DefaultOutputQueueURI が除去されました。

第 1 章 概要

- 『作業ロードマップ』
- 『用語』
- 3 ページの『アダプター環境の概要』

この章では、知っておく必要がある用語について説明し、アダプターでの処理について解説して、全体像を示します。アダプターをインストール、構成、および使用する前にアダプターについて理解しておくことが重要です。

作業ロードマップ

Adapter for WebSphere Business Integration Message Broker を使用するには、表 1 に記載されている作業を行う必要があります。

表 1. アダプターの使用: 作業ロードマップ

作業	関連手順 (参照先)	詳細情報 (参照先)
コネクターのインストール	13 ページの『第 2 章 コネクターのインストール』	「 <i>WebSphere Business Integration Adapters インストール・ガイド</i> 」
ビジネス・オブジェクトとメタオブジェクトの構成	49 ページの『第 5 章 オブジェクトの作成』	「 <i>ビジネス・オブジェクト開発ガイド</i> 」
データ・ハンドラーの構成	67 ページの『第 6 章 データ・ハンドラーの構成』	「 <i>データ・ハンドラー・ガイド</i> 」
コネクターの構成	19 ページの『第 3 章 コネクターの構成』	73 ページの『付録 A. コネクターの標準構成プロパティ』、93 ページの『付録 B. Adapter for WBI Message Broker のコネクター固有のプロパティ』、および「 <i>コネクター開発ガイド</i> 」
コネクターの実行	43 ページの『第 4 章 コネクターの実行』	
コネクターのトラブルシューティング	71 ページの『第 7 章 トラブルシューティング』	
チュートリアルの実行	99 ページの『付録 C. チュートリアル』	

用語

アダプターについて理解するには、以下の用語を理解する必要があります。

アダプター

統合ブローカーとアプリケーション (またはテクノロジー) との間の通信をサポートするコンポーネントを備えている、WebSphere Business Integration

システムのコンポーネント。アダプターには、コネクター、メッセージ・ファイル、および構成ツールが必ず含まれています。また、Object Discovery Agent (ODA) も含まれていることがあります。また、一部のアダプターではデータ・ハンドラーが必要な場合があります。

アダプター・フレームワーク

IBM が提供する、アダプターの構成と実行のためのソフトウェア。アダプター・フレームワークには、ランタイム・コンポーネントとして、Java ランタイム環境、コネクター・フレームワーク、および Object Discovery Agent (ODA) ランタイムが組み込まれています。コネクター・フレームワークには、コネクターを新規開発するときに必要となるコネクター・ライブラリー (C++ および Java) が含まれています。ODA ランタイムには、ODA を新規開発するときに必要となる Object Development Kit (ODK) のライブラリーが含まれています。構成用コンポーネントとしては、以下のツールが用意されています。

- Business Object Designer
- Connector Configurator
- Log Viewer
- System Manager
- Adapter Monitor
- Test Connector
- アダプターに関連付けられた Object Discovery Agent (ODA) (用意されていない場合もあります)

Adapter Development Kit (ADK)

アダプター開発用のサンプルをいくつか備えた開発キット。サンプルには、コネクターと Object Discovery Agent (ODA) のサンプルも含まれます。

コネクター

ビジネス・オブジェクトを使用して、統合ブローカーにイベント関連の情報を送信し (イベント通知)、統合ブローカーから要求関連の情報を受信する (要求処理)、アダプターのコンポーネント。コネクターは、コネクター・フレームワークと、コネクターのアプリケーション固有のコンポーネントで構成されています。

コネクター・フレームワーク

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統合ブローカーの間の相互作用を管理する、コネクターのコンポーネント。このコンポーネントは、必要な管理サービスをすべて備えており、コネクターが必要とするメタデータをリポジトリから取得します。コネクター・フレームワークは Java で記述されており、C++ で記述されたアプリケーション固有のコンポーネントをサポートできるように C++ 拡張が組み込まれています。コネクター・フレームワークのコードはすべてのコネクターで共通です。

コネクター・コントローラー

コラボレーションと相互作用する、コネクター・フレームワークのサブコンポーネント。コネクター・コントローラーは、InterChange Server 内で動作するもので、アプリケーション固有のビジネス・オブジェクトと汎用ビジネス・オブジェクトの間のマッピングを開始し、コラボレーションのビジネス・オブジェクト定義に対するサブスクリプションを管理します。

統合ブローカー

異種アプリケーションの間でデータを統合する、WebSphere Business Integration システムのコンポーネント。通常、統合ブローカーはさまざまなサービスを備えています。このサービスには、データをルーティングする機能、統合プロセスを決定する規則のリポジトリ、各種アプリケーションに接続する機能、および統合を容易にする管理機能が含まれます。統合ブローカーには、WebSphere Business Integration Message Broker や WebSphere Business InterChange Server などがあります。

WebSphere Business Integration システム

多様なソースの間で情報を移動してビジネス関連の情報を交換し、エンタープライズ環境内の異種アプリケーションの間で情報の処理とルーティングを行う、エンタープライズ・ソリューション。このビジネス・インテグレーション・システムは、統合ブローカーと 1 つ以上のアダプターで構成されています。

WebSphere Integration Message Broker バージョン 2.2

WebSphere MQ キュー間でメッセージの変換とルーティングを行う、メッセージ・ブローカー製品。このテクノロジーにより、アプリケーションは、リモート・キューとの間でメッセージを送受信して非同期的に通信することができます。WebSphere Integration Message Broker での大きな変更点は、ユーザー定義のロジックに基づいてメッセージをフォーマット、格納、およびルーティングできる機能を追加するメッセージ・フローが追加されたことです。

アダプター環境の概要

図 1 に、WebSphere Business Integration システムでのアダプター、関連コンポーネント、およびこれらの間の関係を示します。この図は標準的な構成を示しています。アダプターはレガシー・アプリケーションとメッセージを交換するように構成されており、レガシー・アプリケーションのメッセージは WebSphere Business Integration Message Broker を通じて送受信されます。また、アダプターは、InterChange Server とビジネス・オブジェクトを交換するようにも構成されています。コネクタはメタデータ主導型です。メッセージのルーティングおよびフォーマット変換は、イベント・ポーリング技法によって開始されます。コネクタは、Java™ Message Service (JMS) の MQ インプリメンテーションを使用します。JMS は、エンタープライズ・メッセージング・システムにアクセスするための API です。

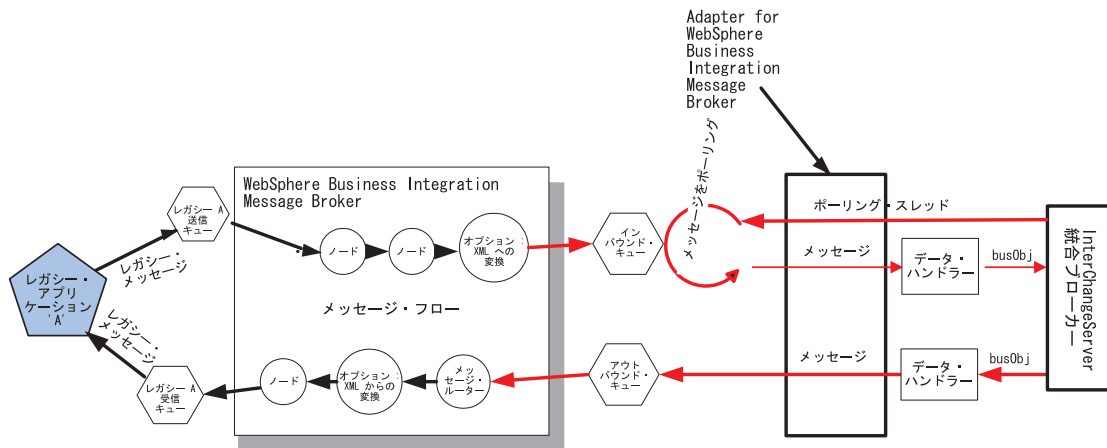


図 1. WBI 環境でのアダプター

コネクターを使用すると、コラボレーションと、データの変更が発生したときに WebSphere MQ メッセージを送受信するアプリケーションとの間で、非同期的にビジネス・オブジェクトを交換できます。

コネクターはキューから WebSphere MQ メッセージを検索し、データ・ハンドラーを呼び出してメッセージを対応するビジネス・オブジェクトに変換し、コラボレーションにデリバリーします。反対方向の場合、コネクターはコラボレーションからビジネス・オブジェクトを受け取り、同じデータ・ハンドラーを使用して WebSphere MQ メッセージに変換し、WebSphere MQ キューにデリバリーします。

推奨: コネクターは、任意のデータ・ハンドラーを使用してメッセージを処理するように構成できます。ただし、WebSphere Integration Message Broker は構文解析可能なすべてのメッセージをオプションで XML 形式に変換できるため、すべてのメッセージを XML 形式でデリバリーするようにコネクターを構成することを強くお勧めします。これは、処理のために XML データ・ハンドラーを実装することを意味します。概要および手順については、67 ページの『データ・ハンドラーの構成の概要』を参照してください。

アダプターでの処理の概要

アダプターは、Java Message Service (JMS) の IBM による WebSphere MQ インプリメンテーションを使用します。JMS は、エンタープライズ・メッセージング・システムにアクセスするためのオープン・スタンダード API です。JMS は、ビジネス・アプリケーションがデータとイベントを非同期的に送受信できるように設計されています。

アダプターは、WBI Message Broker と直接相互作用しません。コネクターを構成する際には、WBI Message Broker メッセージ・フローの入力ノードおよび出力ノードとして WebSphere MQ キューをセットアップします。アダプターは、このメッセージ・フローが配置されているメッセージ・ブローカーをホスティングしている WebSphere MQ キュー・マネージャーと通信します。

メッセージ要求

図2 に、アダプターのランタイム・コンポーネントであるコネクターが行うメッセージ要求の通信を示します。doVerbFor() メソッドがコラボレーションからビジネス・オブジェクトを受け取ると、コネクターはそのビジネス・オブジェクトをデータ・ハンドラーに渡します。データ・ハンドラーはそのビジネス・オブジェクトを適したメッセージに変換し、キューに送ります。このとき、JMS 層は適切な呼び出しを実行してキュー・セッションを開き、メッセージの経路を指定します。コネクターは、要求を非同期的に送付する（応答不要送信を行う）ように構成できます。また、要求を同期的に処理できるようにコネクター・プロパティを構成することもできます。

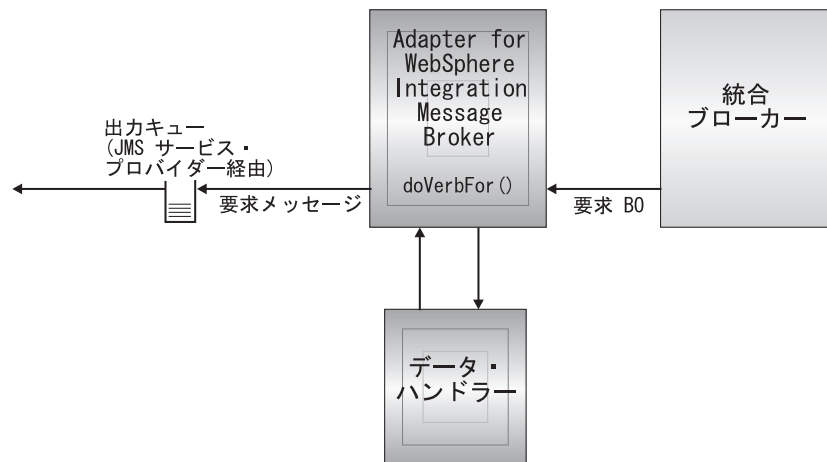


図2. メッセージ要求の処理

コネクターは、コラボレーションから渡されたビジネス・オブジェクトを、各ビジネス・オブジェクトの動詞に基づいて処理します。コネクターはビジネス・オブジェクト・ハンドラーと doVerbFor() メソッドを使用して、コネクターがサポートするビジネス・オブジェクトを処理します。コネクターは、以下のビジネス・オブジェクトの動詞をサポートします。

- Create
- Update
- Delete
- Retrieve
- Exists
- Retrieve by Content

注: Create 動詞、Update 動詞、および Delete 動詞を持つビジネス・オブジェクトは、非同期的にも同期的にも送信できます。デフォルト・モードは非同期送信です。コネクターは、Retrieve 動詞、Exists 動詞、Retrieve by Content 動詞のビジネス・オブジェクトの非同期送信をサポートしません。したがって、Retrieve 動詞、Exists 動詞、または Retrieve by Content 動詞のデフォルト・モードは同期送信です。

Create、Update、および Delete

Create 動詞、Update 動詞、および Delete 動詞を持つビジネス・オブジェクトの処理は、ビジネス・オブジェクトが非同期的に送信されたか同期的に送信されたかによって決まります。

非同期送達: これは、Create 動詞、Update 動詞、および Delete 動詞を持つビジネス・オブジェクトのデフォルト送信モードです。データ・ハンドラーを使用して、ビジネス・オブジェクトからメッセージが作成され、出力キューに書き込まれます。メッセージがデリバリーされると、コネクタは SUCCESS を戻します。それ以外の場合は FAIL を戻します。

注: コネクタには、メッセージが受信されたかどうか、または、処置が行われたかどうかを確認する方法はありません。

同期送達: コネクタ固有のプロパティで ReplyToQueue が定義されており、かつビジネス・オブジェクトのメタオブジェクト変換プロパティに responseTimeout が存在する場合、コネクタは同期モードで要求を送信します。続いて、コネクタは、受信側のアプリケーションで適切な処置が行われたかどうかを確認するために応答を待ちます。

WebSphere Integration Message Broker の場合、コネクタは最初に、表 2 に示すヘッダーを持つメッセージを発行します。

表 2. 要求メッセージ記述子ヘッダー (MQMD)

フィールド	説明	値
Format	フォーマット名	メタオブジェクト変換プロパティに定義されている出力フォーマット。IBM の要件に合わせて、8 文字を超える部分が切り捨てられます (例 : MQSTR)
MsgType	メッセージ・タイプ	MQMT_DATAGRAM*
Report	要求されたレポート・メッセージのオプション	応答メッセージの返送が予測される場合、このフィールドには次の値が取り込まれます。処理が成功したときに肯定処理レポートが必要な場合は、MQRO_PAN*。処理が失敗したときに否定処理レポートが必要な場合は、MQRO_NAN*。生成されるレポートの相関 ID が最初に発行された要求のメッセージ ID と同じになる必要がある場合は、MQRO_COPY_MSG_ID_TO_CORREL_ID*。
ReplyToQ	応答キューの名前	応答メッセージの返送が予測される場合、このフィールドにはコネクタ・プロパティ ReplyToQueue の値が取り込まれます。
Persistence	メッセージのパーシスタンス	MQPER_PERSISTENT*
Expiry	メッセージの存続時間	MQEI_UNLIMITED*

* は、IBM によって定義される定数を示します。

表 2 に示すヘッダーの後に、メッセージ本文が続きます。メッセージの本文は、データ・ハンドラーを使用して直列化されたビジネス・オブジェクトです。

Report フィールドは、受信側アプリケーションから肯定処理レポートと否定処理レポートの両方の返送が予測されることを示すために設定されます。メッセージを発行したスレッドは、受信側アプリケーションが要求を処理できたかどうかを示す応答メッセージを待ちます。

アプリケーションがコネクタから同期要求を受け取ると、アプリケーションはビジネス・オブジェクトを処理し、表 3、4、および 5 に示すレポート・メッセージを発行します。

表 3. 応答メッセージ記述子ヘッダー (MQMD)

フィールド	説明	値
Format	フォーマット名	変換プロパティ内で定義された busObj の入力フォーマット
MsgType	メッセージ・タイプ	MQMT_REPORT*

* は、IBM によって定義される定数を示します。

表 4. 応答メッセージに含まれるデータ

動詞	Feedback フィールド	メッセージの本文
Create、Update、または Delete	SUCCESS VALCHANGE	(オプション) 変更を反映する、直列化されたビジネス・オブジェクト。
	VALDUPES FAIL	(オプション) エラー・メッセージ。

表 5. WebSphere Integration Message Broker フィードバック・コードと WebSphere Business Integration システムの応答値

WebSphere Integration Message Broker フィードバック・コード	対応する WebSphere Business Integration システムの応答*
MQFB_PAN または MQFB_APPL_FIRST	SUCCESS
MQFB_NAN または MQFB_APPL_FIRST + 1	FAIL
MQFB_APPL_FIRST + 2	VALCHANGE
MQFB_APPL_FIRST + 3	VALDUPES
MQFB_APPL_FIRST + 4	MULTIPLE_HITS
MQFB_APPL_FIRST + 5	FAIL_RETRIEVE_BY_CONTENT
MQFB_APPL_FIRST + 6	BO_DOES_NOT_EXIST
MQFB_APPL_FIRST + 7	UNABLE_TO_LOGIN (この応答後、コネクタ・エージェントは即時に終了します)
MQFB_APPL_FIRST + 8	APP_RESPONSE_TIMEOUT

* 詳細については、「コネクタ開発ガイド」を参照してください。

ビジネス・オブジェクトを処理できる場合、アプリケーションは、feedback フィールドが MQFB_PAN (または特定の WebSphere Business Integration システムの値) に設定されたレポート・メッセージを作成します。また、オプションで、すべての変更を含む直列化されたビジネス・オブジェクトをメッセージ本文に取り込みます。ビジネス・オブジェクトを処理できない場合、アプリケーションは、feedback フィールドが MQFB_NAN (または特定の WebSphere Business Integration システムの値) に設定されたレポート・メッセージを作成します。オプションで、このレポート・メッセージの本文にエラー・メッセージを含めることもできます。どちらの場合でも、

アプリケーションはメッセージの correlationID フィールドをコネクター・メッセージの messageID に設定し、ReplyToQueue フィールドで指定されたキューにメッセージを発行します。

コネクターは、応答メッセージを取り出すと、デフォルトにより応答の correlationID を要求メッセージの messageID と突き合わせます。続いて、要求を発行したスレッドに通知を送信します。コネクターは、応答の feedback フィールドの設定によって、メッセージの本文にビジネス・オブジェクトとエラー・メッセージのどちらが含まれているかを予測します。ビジネス・オブジェクトが含まれていると予測したにもかかわらず、メッセージの本文にビジネス・オブジェクトが取り込まれていなかった場合、コネクターは InterChange Server が要求操作のために最初に発行したのと同じビジネス・オブジェクトを単純に返送します。エラー・メッセージが含まれていると予測したにもかかわらず、メッセージの本文にエラー・メッセージが取り込まれていなかった場合、InterChange Server には応答コードと汎用エラー・メッセージが返送されます。ただし、メッセージ選択子を使用して識別やフィルター操作を行うこともできます。あるいは、アダプターが特定の要求に対して応答メッセージを識別する方法を制御できます。このメッセージ選択子機能は JMS 機能です。この JMS 機能は同期要求処理にのみ適用されます。以下に詳細を説明します。

Retrieve, Exists, および Retrieve by content: Retrieve 動詞、Exists 動詞、および Retrieve By Content 動詞を持つビジネス・オブジェクトは、同期送信のみをサポートします。コネクターは、これらの動詞を持つビジネス・オブジェクトを、Create 動詞、Update 動詞、および Delete 動詞に対して定義されている同期送信と同様に処理します。ただし、Retrieve 動詞、Exists 動詞、および Retrieve By Content 動詞を使用する場合には、responseTimeout と replyToQueue が必要です。さらに、Retrieve By Content 動詞と Retrieve 動詞の場合、トランザクションを完了するためにはメッセージの本文に直列化されたビジネス・オブジェクトが取り込まれている必要があります。

表 6 に、これらの動詞に対応する応答メッセージを示します。

表 6. 応答メッセージへの取り込み

動詞	Feedback フィールド	メッセージの本文
Retrieve または RetrieveByContent	FAIL FAIL_RETRIEVE_BY_CONTENT	(オプション) エラー・メッセージ。
	MULTIPLE_HITS SUCCESS	直列化されたビジネス・オブジェクト。
Exist	FAIL	(オプション) エラー・メッセージ。
	SUCCESS	

イベント処理

図 3 に、コネクターでのイベント処理を示します。pollForEvents() メソッドは、次の該当するメッセージを入力キューから検索します。メッセージは実行中のキューに入れられ、処理が完了するまでキュー内に残ります。コネクターは最初に、コネクター・メタオブジェクトを使用して、そのメッセージ・タイプがサポートされているかどうかを調べます。サポートされている場合、コネクターは構成されてい

るデータ・ハンドラーにメッセージを渡し、データ・ハンドラーがそのメッセージをビジネス・オブジェクトに変換します。設定される動詞には、そのメッセージ・タイプに対して定義されている変換プロパティが反映されます。次に、コネクターは、そのビジネス・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かどうかを調べます。サブスクライブされている場合、`gotAppEvents()` メソッドがビジネス・オブジェクトを統合ブローカーにデリバリーし、実行中のキューからメッセージが削除されます。

コネクターは、イベント通知のために、データベース・トリガーではなくアプリケーションによってキューに書き込まれたイベントを検出します。イベントは、アプリケーションまたはその他の MQ 対応ソフトウェアが WebSphere MQ メッセージを生成して MQ メッセージ・キューに格納するときが発生します。

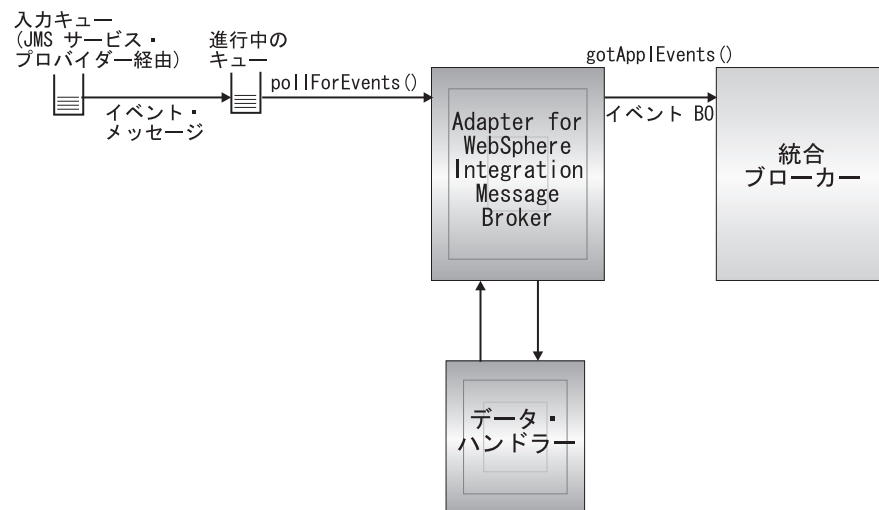


図3. アプリケーションとコネクターの間の通信方法: メッセージ戻り

検索

コネクターは、`pollForEvents()` メソッドを使用して MQ キューからメッセージを定期的にポーリングします。メッセージを検出すると、コネクターはそれを MQ キューから検索して調べ、メッセージのフォーマットを判別します。フォーマットがコネクター・メタオブジェクト内で定義されている場合、コネクターはデータ・ハンドラーを使用して適切な動詞付きのビジネス・オブジェクトを生成します。イベント失敗のシナリオについては、46 ページの『エラー処理の概要』を参照してください。

コネクターは、最初に入力キューとのトランザクション・セッションを開いて、メッセージを処理します。このトランザクション・アプローチを使用すると、コネクターがビジネス・オブジェクトを正常にサブミットしたにもかかわらず、キューでトランザクションをコミットできなかった場合に、コラボレーションにビジネス・オブジェクトが 2 回デリバリーされるという可能性が少し残ります。この問題を回避するために、コネクターは、すべてのメッセージを進行中キューに移動します。

メッセージは、処理が完了するまでキュー内に残ります。処理中にコネクタが予期しないエラーでシャットダウンした場合、メッセージは元の入力キューには戻されず、実行中のキュー内に残ります。

注: JMS サービス・プロバイダーとのトランザクション・セッションでは、キュー上の要求されたすべての処理が、キューからイベントが削除される前に実行され、コミットされる必要があります。したがって、コネクタがキューからメッセージを検索するときには、次の 3 つの処理が実行されるまでは検索がコミットされません。1) メッセージからビジネス・オブジェクトへの変換、2) `getApplEvents()` メソッドによる、InterChange Server へのビジネス・オブジェクトのデリバリー、および 3) 戻り値の受信。

リカバリー

コネクタは初期化の際に実行中のキューを調べ、コネクタのシャットダウンが原因で未処理のまま残っているメッセージがないかどうかを調べます。コネクタの構成プロパティ `InDoubtEvents` を使用すると、そのようなメッセージのリカバリー処理に関する 4 つのオプション (`fail on startup`、`reprocess`、`ignore`、または `log error`) のうち、いずれかを指定できます。

Fail on startup: `fail on startup` オプションを指定した場合、コネクタが初期化の際に実行中のキュー内でメッセージを検出すると、コネクタはエラーを記録し、即時にシャットダウンします。ユーザーまたはシステム管理者は、検出されたメッセージを調べ、これらのメッセージを完全に削除するかまたは別のキューに移動するなどの適切な処置を取る必要があります。

Reprocess: `reprocessing` オプションを指定した場合、コネクタが初期化の際に実行中のキュー内でメッセージを検出すると、コネクタは以降のポーリングでそのメッセージを最初に処理します。実行中のキュー内にあったすべてのメッセージの処理が完了すると、コネクタは入力キューからのメッセージの処理を開始します。

Ignore: `ignore` オプションを指定した場合、初期化の際にコネクタが実行中のキュー内でメッセージを検出すると、コネクタはそれを無視しますが、シャットダウンはしません。

Log error: `log error` オプションを指定した場合、初期化の際にコネクタが実行中のキュー内でメッセージを検出すると、コネクタはエラーを記録しますが、シャットダウンはしません。

アーカイブ

コネクタのプロパティ `ArchiveQueue` が指定されており、かつ有効なキューを示している場合には、コネクタは正常に処理されたすべてのメッセージのコピーをアーカイブ・キューに格納します。`ArchiveQueue` が未定義の場合、メッセージは処理後に破棄されます。アンサブスクライブされたメッセージまたはエラーを含むメッセージのアーカイブの詳細については、46 ページの『エラー処理の概要』を参照してください。

注: JMS 規則により、検索したメッセージを即時に別のキューに送信することはできません。メッセージをアーカイブして再デリバリーできるようにするために、コネクタは、オリジナルのメッセージから本文とヘッダー（該当する場合

のみ) を複製した第 2 のメッセージを最初に生成します。JMS サービス・プロバイダーとの競合を避けるため、JMS に必須フィールドのみが複製されます。したがって、`format` フィールドは、アーカイブまたは再デリバリーされるメッセージにコピーされる唯一の追加メッセージ・プロパティとなります。

第 2 章 コネクタのインストール

- 『アダプター環境』
- 『インストール作業の概要』
- 16 ページの『インストールの検証』

この章では、コネクタのインストール方法および構成方法と、メッセージ・フローをコネクタとともに動作させるための構成方法について説明します。

インストール作業の概要

Adapter for WebSphere Integration Message Broker をインストールするには、以下の作業を行う必要があります。

- **統合ブローカーのインストール:** この作業では、WebSphere Business Integration システムのインストールと統合ブローカーの始動を行います。作業の詳細については、使用するブローカーおよびオペレーティング・システムのインストール文書に説明があります。
- **アダプターおよび関連ファイルのインストール:** この作業では、アダプター用のファイルをソフトウェア・パッケージから使用システムにインストールします。15 ページの『アダプターと関連ファイルのインストール』を参照してください。

アダプターをインストールする前に、アダプター環境について理解しておく必要があります。詳細については、『アダプター環境』を参照してください。

この章で説明する作業

この章で説明する作業は、以下のとおりです。

表 7. アダプターのインストール: 作業ロードマップ

作業	関連手順 (参照先)	詳細情報 (参照先)
アダプターのインストール	15 ページの『アダプターと関連ファイルのインストール』	「 <i>WebSphere Business Integration Adapters</i> インストール・ガイド」
インストールの検証	16 ページの『インストールの検証』	

アダプター環境

アダプターをインストール、構成、および使用する前に、アダプターの環境要件を理解しておく必要があります。

- 14 ページの『ブローカーの互換性』
- 14 ページの『アダプターのプラットフォーム』
- 15 ページの『アダプターの依存関係』
- 15 ページの『ロケール依存データ』

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 WebSphere Integration Message Broker バージョン 2.6 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ます。

- アダプター・フレームワーク: WebSphere Business Integration Adapter Framework の以下のバージョン:
 - 2.1
 - 2.2
 - 2.3.x
 - 2.4
- 統合ブローカー:
 - WebSphere InterChange Server の以下のバージョン:
 - 4.11
 - 4.2
 - 4.2.1
 - 4.2.x
 - WebSphere Studio Application Developer Integration Edition バージョン 5.0.1 が組み込まれた WebSphere Application Server Enterprise バージョン 5.0.2

例外については、「リリース情報」を参照してください。

注: 統合ブローカーのインストールおよびその前提条件に関する説明については、以下の資料を参照してください。

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

Message Broker (WebSphere Integration Message Broker) 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*」、および Message Broker のインストール関連資料を参照してください。これらの資料には、Web サイト <http://www.ibm.com/software/integration/mqfamily/library/manualsa/> から検索できるものもあります。WebSphere Application Server については、「アダプター実装ガイド (*WebSphere Application Server*)」および <http://www.ibm.com/software/webservers/appserv/library.html> にある資料を参照してください。

アダプターのプラットフォーム

このアダプターは以下のプラットフォームでサポートされます。

- Windows 2000
- AIX 5.1、5.2
- Solaris 8
- HP-UX 11i

アダプターの依存関係

このアダプターには、以下のソフトウェア前提条件と、その他の依存関係があります。

- コネクターは、WebSphere MQ または WebSphere MQ 5.1、5.2¹、および 5.3 を介したアプリケーションとのインターオペラビリティをサポートします。したがって、これらのいずれかのソフトウェア・リリースをインストールする必要があります。

注: このアダプターは、WebSphere MQ 5.3 環境で SSL (Secure Socket Layer) をサポートしていません。アダプター・フレームワークと統合ブローカーの通信にとって適切な WebSphere MQ ソフトウェア・バージョンについては、使用プラットフォーム (Windows または UNIX) のインストール・ガイドを参照してください。

- さらに、IBM WebSphere MQ Java クライアント・ライブラリーも必要です。

ロケール依存データ

コネクターは、2 バイト文字セットをサポートし、指定された言語でメッセージ・テキストを送達できるように国際化されています。コネクターは、1 つの文字コードを使用する場所から別のコード・セットを使用する場所にデータを転送するとき、データの意味を保存するように文字変換を実行します。

Java 仮想マシン (JVM) 内での Java ランタイム環境は、Unicode 文字コード・セットでデータを表します。Unicode には、ほとんどの既知の文字コード・セット (1 バイト系とマルチバイト系をいずれも含む) に対応できるエンコード方式が組み込まれています。WebSphere Business Integration システムのほとんどのコンポーネントは Java で記述されています。したがって、ほとんどのインテグレーション・コンポーネントの間でデータが転送されても、文字変換の必要はありません。

エラー・メッセージと通知メッセージを適切な言語で記録するには、該当する環境の Locale 標準構成プロパティを設定します。構成プロパティの詳細については、73 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。

アダプターと関連ファイルのインストール

始める前に: インストールするアダプターに関して、要件、依存関係、およびブローカーの互換性を検討します。詳細については、13 ページの『アダプター環境』を参照してください。

WebSphere Business Integration Adapter 製品のインストールについては、「*WebSphere Business Integration Adapters* インストール・ガイド」を参照してください。この資料は、次の Web サイトの WebSphere Business Integration Adapters Infocenter にあります。

1. ご使用の環境に「get 時の変換」方式の文字セット変換が実装されている場合、IBM から最新の MA88 (JMS クラス) をダウンロードする必要があります。パッチ・レベルは最低でも 5.2.2 である必要があります (WebSphere MQ バージョン 5.2 の場合)。これにより、サポートされないエンコード・エラーを避けることができます。

インストールの検証

以下のセクションでは、インストール後の製品のパスとファイル名およびアダプターのインストールの検証方法について説明します。

Windows システムでのインストールの検証の概要

始める前に: アダプターをインストールします。インストーラーは、アダプターに関連付けられた標準ファイルをご使用のシステムにコピーします。このユーティリティーは、コネクターを `ProductDir%\connectors\WBIMB` ディレクトリーにインストールし、そのコネクターへのショートカットを「スタート」メニューに追加します。

Windows システムでのインストールの検証のステップ

Windows システムでアダプターのインストールを検証するときは、次のステップを実行します。

- アダプターをインストールしたディレクトリー (`ProductDir\`) に移動し、そのディレクトリーの内容を、表 8 に記載されている内容と比較します。

表 8 で、アダプターにより使用される Windows ファイル構造について説明し、インストーラーによるアダプターのインストールを選択したときに自動的にインストールされるファイルを示します。

表 8. アダプター用にインストールされた Windows ファイル構造

<code>ProductDir</code> のサブディレクトリー	説明
<code>connectors\WebSphereBIMessageBroker\CWWebSphereBIMessageBroker.jar</code>	WebSphere Business Integration Message Broker コネクター・エージェントによってのみ使用されるクラスを含みません。
<code>connectors\WebSphereBIMessageBroker\start_WebSphereBIMessageBroker.bat</code>	コネクターの始動スクリプト
<code>connectors\messages\WebSphereBIMessageBrokerConnector.txt</code>	コネクターのメッセージ・ファイル
<code>bin\Data\App\WebSphereBIMessageBrokerConnectorTemplate</code>	アダプター定義のテンプレート・ファイル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\WBIMBConnector.cfg</code>	WBIMB アダプター構成ファイルのサンプル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\PortConnector.cfg</code>	ポート・コネクター構成ファイルのサンプル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\Sample_WBIMB_LegacyItem.xsd</code>	サンプル・ビジネス・オブジェクト・リポジトリー・ファイル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\Sample_WBIMB_LegacyItem_XMLDoc.xsd</code>	サンプル・ビジネス・オブジェクト・リポジトリー・ファイル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\Sample_WBIMB_MO_Config.xsd</code>	コネクター・エージェント・メタオブジェクトのサンプル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\Sample_WBIMB_MO_DataHandler.xsd</code>	データ・ハンドラー用のトップレベル・メタオブジェクトのサンプル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\Sample_WBIMB_MO_DataHandler_XMLConfig.xsd</code>	XML データ・ハンドラーの構成に使用されるメタオブジェクトのサンプル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\LegacyItem.txt</code>	サンプルを使用するための入力ファイルのサンプル
<code>connectors\WebSphereBIMessageBroker\samples\LegacyItem\mqinput.exe</code>	サンプルとともに使用する WebSphere MQ 用のユーティリティー

表 8. アダプター用にインストールされた Windows ファイル構造 (続き)

<i>ProductDir</i> のサブディレクトリー	説明
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥Sample_WBIMB_Project¥.project	WBIMB プロジェクトのサンプル
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥Sample_WBIMB_Project¥LocalDomain.configmgr	
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥Sample_WBIMB_Project¥Sample_WBIMB_bar.bar	メッセージ・フロー・プロジェクトのサンプル
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥MSG_FLOW_RPROJECT¥.project	
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥MSG_FLOW_RPROJECT¥AdjustFormat.esql	メッセージ・フロー・プロジェクトのサンプル
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥MSG_FLOW_RPROJECT¥Check_Color.esql	
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥MSG_FLOW_RPROJECT¥From_LegacyApplication_to_WBI.msgflow	メッセージ・フロー・プロジェクトのサンプル
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥MSG_FLOW_RPROJECT¥From_WBI_to_LegacyApplication.msgflow	
connectors¥WebSphereBIMessageBroker¥samples¥LegacyItem¥MSG_FLOW_RPROJECT¥Loopback.msgflow	

注: すべての製品のパス名は、ご使用のシステムで製品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。

UNIX システムでのインストールの検証の概要

始める前に: アダプターをインストールします。インストーラーは、アダプターに関連付けられた標準ファイルをご使用のシステムにコピーします。このユーティリティーは、コネクタ・エージェントを *ProductDir/connectors/WBIMB* ディレクトリーにインストールします。

UNIX システムでのインストールの検証のステップ

UNIX システムでアダプターのインストールを検証するときは、次のステップを実行します。

- アダプターをインストールしたディレクトリー (*ProductDir*) に移動し、そのディレクトリーの内容を、表 9 に記載されている内容と比較します。

表 9 で、アダプターにより使用される UNIX ファイル構造について説明し、インストーラーによるアダプターのインストールを選択したときに自動的にインストールされるファイルを示します。

表 9. コネクタ用にインストールされた UNIX ファイル構造

<i>ProductDir</i> のサブディレクトリー	説明
connectors/WebSphereBIMessageBroker/CWWebSphereBIMessageBroker.jar	WebSphere Business Integration Message Broker コネクタ・エージェントによってのみ使用されるクラスを含みません。
connectors/WebSphereBIMessageBroker/start_WebSphereBIMessageBroker.sh	コネクタのシステム始動スクリプト
connectors/messages/WebSphereBIMessageBrokerConnector.txt	コネクタのメッセージ・ファイル
bin/Data/App/WebSphereBIMessageBrokerConnectorTemplate	アダプター定義のテンプレート・ファイル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/WBIMBConnector.cfg	WBIMB アダプター構成ファイルのサンプル

表9. コネクター用にインストールされた UNIX ファイル構造 (続き)

ProductDir のサブディレクトリー	説明
connectors/WebSphereBIMessageBroker/samples/LegacyItem/PortConnector.cfg	ポート・コネクター構成ファイルのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_LegacyItem.xsd	サンプル・ビジネス・オブジェクト・リポジトリ・ファイル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_LegacyItem_XMLDoc.xsd	サンプル・ビジネス・オブジェクト・リポジトリ・ファイル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_MO_Config.xsd	コネクター・エージェント・メタオブジェクトのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_MO_DataHandler.xsd	データ・ハンドラー用のトップレベル・メタオブジェクトのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_MO_DataHandler_XMLConfig.xsd	XML データ・ハンドラーの構成に使用されるメタオブジェクトのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/LegacyItem.txt	サンプルを使用するための入力ファイルのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/mqsiput.exe	サンプルとともに使用する WebSphere MQ 用のユーティリティー
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_Project/.project	WBIMB プロジェクトのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_Project/LocalDomain.configmgr	
connectors/WebSphereBIMessageBroker/samples/LegacyItem/Sample_WBIMB_Project/Sample_WBIMB_bar.bar	
connectors/WebSphereBIMessageBroker/samples/LegacyItem/MSG_FLOW_RPROJECT /.project	メッセージ・フロー・プロジェクトのサンプル
connectors/WebSphereBIMessageBroker/samples/LegacyItem/MSG_FLOW_RPROJECT/AdjustFormat.esql	
connectors/WebSphereBIMessageBroker/samples/LegacyItem/MSG_FLOW_RPROJECT/Check_Color.esql	
connectors/WebSphereBIMessageBroker/samples/LegacyItem/MSG_FLOW_RPROJECT/From_LegacyApplication_to_WBI.msgflow	
connectors/WebSphereBIMessageBroker/samples/LegacyItem/MSG_FLOW_RPROJECT/From_WBI_to_LegacyApplication.msgflow	
connectors/WebSphereBIMessageBroker/samples/LegacyItem/MSG_FLOW_RPROJECT/Loopback.msgflow	

注: すべての製品のパス名は、ご使用のシステムで製品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。

第 3 章 コネクタの構成

- 20 ページの『Connector Configurator の概要』
- 21 ページの『Connector Configurator の始動』
- 22 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』
- 25 ページの『新規構成ファイルの作成』
- 26 ページの『既存ファイルの使用』
- 28 ページの『構成ファイル・プロパティの設定』
- 35 ページの『構成ファイルの保管』
- 36 ページの『構成ファイルの変更』
- 36 ページの『構成の完了』
- 37 ページの『グローバル化環境における Connector Configurator の使用』
- 39 ページの『構成例の検証』
- 37 ページの『複数コネクタ・インスタンスの作成』
- 39 ページの『キューの Uniform Resource Identifier (URI) の設定』

この章では、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。

コネクタの構成の概要

コネクタの構成は、Connector Configurator とその他のツールを使用して行います。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。

コネクタのインスタンスを複数作成する場合も、Connector Configurator を使用します。

この章で説明するその他の構成手順については、作業ロードマップを参照してください。

この章で説明する作業

この章で説明する作業は、以下のとおりです。

表 10. コネクターの構成: 作業ロードマップ

作業	関連手順 (参照先)	詳細情報 (参照先)
Connector Configurator の始動	21 ページの『Connector Configurator の始動』	
コネクター固有のプロパティ・テンプレートの作成	22 ページの『コネクター固有のプロパティ・テンプレートの作成』	
構成ファイルの作成	25 ページの『新規構成ファイルの作成』	
構成プロパティの設定	28 ページの『構成ファイル・プロパティの設定』	
構成ファイルの保管	35 ページの『構成ファイルの保管』	
構成ファイルの変更	36 ページの『構成ファイルの変更』	
構成の検証	36 ページの『構成の完了』	
構成のローカライズ	37 ページの『グローバル化環境における Connector Configurator の使用』	
複数コネクター・インスタンスの作成	37 ページの『複数コネクター・インスタンスの作成』	
構成例の検証	39 ページの『構成例の検証』	
キュー URI の設定	39 ページの『キューの Uniform Resource Identifier (URI) の設定』	
保証付きイベント・デリバリーの使用	41 ページの『イベント・デリバリーの保証』	

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ー・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。

- コネクターを構成するためのコネクター固有のプロパティ・テンプレートを作成する。
- コネクター構成ファイルを作成する。インストールするコネクター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する。
場合によっては、コネクター・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る場合があります。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 (『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 (すべてのコネクタがもつプロパティ) 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、23 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、以下の手順を実行します。

- 「スタート」>「プログラム」から、「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Tools」>「Connector Configurator」をクリックします。

- 「ファイル」>「新規」>「コネクタ構成」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 (28 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、以下の手順を実行します。

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、以下の手順を実行します。

- 「System Manager」ウィンドウの「コネクタ」フォルダーで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。
- Connector Configurator で「ファイル」>「開く」を選択します。プロジェクトまたはプロジェクトが保管されているディレクトリーからコネクタ構成ファイルを選択します。
- 「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、または既存のコネクタ定義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、23 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。既存のテンプレートは `¥WebSphereAdapters¥bin¥Data¥App` ディレクトリーにあります。

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。

Connector Configurator でテンプレートを作成するには、以下のステップを実行します。

1. 「ファイル」>「新規」>「コネクタ固有プロパティ・テンプレート (Connector-Specific Property Template)」とクリックします。
2. 「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。
 - 「新規テンプレート名を入力してください」の下の「名前」フィールドに、新規テンプレート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。
 - 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。
3. テンプレートを作成するときには、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。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。
 - 既存のテンプレートを変更する場合には、「変更する既存のテンプレートを選択してください: 検索テンプレート」の下の「テンプレート名」テーブルのリストから、テンプレート名を選択します。
 - このテーブルには、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せん。テンプレートを検索することもできます。

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、最大複数値、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、以下の手順を実行します。

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」のフィールドに値を入力します。

新規プロパティ値を作成するには、以下のステップを実行します。

1. 「プロパティを編集」リストでプロパティを選択し、右マウス・ボタンでクリックします。
2. ダイアログ・ボックスから「追加」を選択します。
3. 新規プロパティ値の名前を入力し、「OK」をクリックします。右側の「値」パネルに値が表示されます。

「値」パネルには、3つの列からな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、作成した以前の値が表示されます。

「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、以下の手順を実行します。

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」表示で強調表示された依存プロパティで、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 (Finish)」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、構成ファイルの名前を指定し、統合ブローカーを選択する必要があります。

- 「System Manager」ウィンドウで「コネクタ」フォルダーを右クリックし、「新規コネクタの作成」を選択します。Connector Configurator 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。
- スタンドアロン・モードの場合は、Connector Configurator で「ファイル」>「新規」>「コネクタ構成」を選択します。「新規コネクタ」ウィンドウで、新規コネクタの名前を入力します。

また、統合ブローカーも選択する必要があります。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。

- 「Integration Broker」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」ウィンドウの残りのフィールドに入力します。

コネクタ固有のテンプレートからの構成ファイルの作成

コネクタ固有のテンプレートを作成すると、その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」ダイアログ・ボックスが表示されます。

- **名前**

コネクタの名前を入力します。名前では大文字と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 (Connector-Specific Property Template)」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ーにナビゲートし、「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ー・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 つ以上の形式になります。

- 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

ジの `¥repository` ディレクトリー内には、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 リポジトリー・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リポジトリー・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。

以下の手順を実行して、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」とクリックします。
2. 「ファイル・コネクタを開く」ダイアログ内で、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 リポジトリー (`*.in`、`*.out`)

ICS 環境でのコネクタの構成にリポジトリー・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。リポジトリー・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せん。

- 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、以下の手順を実行します。

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」とクリックします。

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ーを開くと、「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ー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、コネクター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、以下の手順を実行します。

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 31 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。

複数のコネクター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、コネクター構成をすべて 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

新規のコネクター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ー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ーで、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
- コネクター固有のプロパティ
- 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ーの場合に該当する)

注: JMS メッセージングを使用するコネクタの場合、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。

- コネクタの標準プロパティは、コネクタ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プロパティは、コネクタ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。各コネクタには、そのコネクタ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。

「標準のプロパティ」と「コネクタ固有プロパティ (Connector-Specific Properties)」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。
- プロパティごとに「更新メソッド」フィールドが表示されます。これは、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。この設定を構成することはできません。

標準コネクタ・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る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すると、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 内の他のカテゴリの値を入力するには、そのカテゴリ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) で入力した値は、次のカテゴリに移動しても保持されます。ウィンドウを閉じるときに、すべてのカテゴリ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. 29 ページの『標準コネクタ・プロパティの設定』で説明したように、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

「コネクタ固有プロパティ」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。値の暗号化を解除するには、「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、暗号化が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、各コネクターのアダプター・ユーザー・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 A『コネクターの標準構成プロパティ』の 74 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

コネクター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: コネクターによっては、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。

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ー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ー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」リストの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ー定義が、System Manager の ICL (Integration Component Library) 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、「プロジェクトに保管」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・オブジェクトにエージェント・サポートがある場合、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・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: コネクタの最大トランザクション・レベルは、そのコネクタ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

独立型方式で作業している (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。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 を選択する場合、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。

独立型方式で作業している (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できるビジネス・オブジェクトのリストが示されます。このリストから目的のビジネス・オブジェクトを選択します。

関連付けられたマップ (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リプト・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、サポートされるビジネス・オブジェクトを追加指定した場合、それらの内容は、「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、コネクターの、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表示されます。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、コネクターごとに、サポートされる各ビジネス・オブジェクトにマップを自動的にバインドしようとしています。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ています。

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 (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することができます。

すべてのコネクターでこの機能がサポートされ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の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

コネクタ構成ファイルまたはコネクタ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このデータ・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に使用する値については、『付録 A. コネクタの標準構成プロパティ』にある 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。

構成ファイルの保管

コネクタの構成が完了したら、コネクタ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するか、あるいは
- 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。デフォルトでは、このファイルは %WebSphereAdapters%bin\Data%App に保管されます。
- WebSphere Application Server プロジェクトをセットアップしている場合には、このファイルを WebSphere Application Server プロジェクトに保管することもできます。

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* システム・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。
- 「標準のプロパティー」タブを選択します。
- 「標準のプロパティー」タブの「ブローカー・タイプ」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、プロパティー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變更され、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

コネクターの構成ファイルを作成し、そのファイルを変更した後で、コネクターの始動時にコネクター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、以下に太字で示される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>
```

複数コネクタ・インスタンスの作成

このトピックでは、複数コネクタ・インスタンスの作成の概要と手順を示します。

複数コネクタ・インスタンスの作成の概要

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、

コネクター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。

- コネクター・インスタンス用に新規ディレクトリーを作成します。
- 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。
- 新規コネクター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。

新規ディレクトリーの作成のステップ

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

```
ProductDir¥connectors¥connectorInstance
```

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

コネクターに、コネクター固有のメタオブジェクトがある場合、コネクター・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、次のディレクトリーを作成して、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のステップ

各コネクター・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. 初期コネクター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。初期コネクター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. 初期コネクターのファイルは、次のディレクトリーに入っていないと必要ありません。

```
ProductDir¥repository¥initialConnectorInstance
```

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

コネクター定義の作成のステップ

Connector Configurator 内で、コネクター・インスタンスの構成ファイル (コネクター定義) を作成します。これを行うには、以下の手順を実行します。

1. 初期コネクターの構成ファイル (コネクター定義) をコピーし、名前変更をします。
2. 各コネクター・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、コネクター・プロパティーをカスタマイズします。

始動スクリプトの作成のステップ

始動スクリプトを作成するには、以下のようになります。

1. 初期コネクタの始動スクリプトをコピーし、以下のコネクタ・ディレクトリ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、38 ページの『新規ディレクトリの作成のステップ』で作成したコネクタ・ディレクトリに格納します。
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. 初期コネクタのショートカット・テキストをコピーし、新規コネクタ・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・コネクタ作成の詳細については、「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。

構成例の検証

構成例を検証するには、次のようにします。

- 99 ページの『付録 C. チュートリアル』を参照してください。このチュートリアルに従うと、アダプターに付属しているサンプルを使用してサンプル・シナリオを簡単に構成し実行することができます。

キューの Uniform Resource Identifier (URI) の設定

Uniform Resource Identifier を使用して、キューの指定や、指定したキューの値の設定または変更を行うことができます。これらは、キューを定義するコネクタ固有のプロパティの値を指定するときに行います。

キューの URI を設定するには、次のようにします。

- Connector Configurator を使用して、コネクタ固有のキューを次の構文で指定します。

キューの URI の構文: キューの URI は、シーケンス `queue://` を先頭とし、その後以下に以下の要素が続きます。

- キューが存在しているキュー・マネージャーの名前
- 別の /
- キューの名前
- (オプション) 残りのキュー・プロパティの、名前と値のペアのリスト

例えば、次の URI を指定した場合、キュー・マネージャー `crossworlds.queue.manager` に存在するキュー `IN` に接続し、すべてのメッセージが優先順位 5 の WebSphere MQ メッセージとして送信されます。

`queue://crossworlds.queue.manager/IN?targetClient=1&persistence=5`

40 ページの表 11 に、キュー URI に対応するプロパティ名を示します。

表 11. WebSphere Business Integration Message Broker キューの URI プロパティー

プロパティー名	説明	値
expiry	メッセージの存続時間 (ミリ秒単位)	0 = 無制限
priority	メッセージの優先順位	正の整数 = タイムアウト (ミリ秒単位) 0 から 9 で、1 が最高の優先順位。値 -1 は、このプロパティーがキューの構成によって決定されることを意味します。値 -2 は、コネクタ自身のデフォルト値を使用できるよう指定します。
persistence	メッセージをディスクに「ハード化」するかどうか	1 = 非永続 2 = 永続
CCSID	宛先の文字セット	値 -1 は、このプロパティーがキューの構成によって決定されることを意味します。値 -2 は、コネクタ自身のデフォルト値を使用できるよう指定します。 整数: WebSphere Business Integration Message Broker の資料にリストされている有効な値。この値は、CCSID のコネクタ固有構成プロパティーの値と一致していることが必要です。95 ページの『CCSID』を参照してください。
targetClient	受信側アプリケーションが JMS 準拠であるかどうか	0 = JMS (MQRFH2 ヘッダー)
encoding	数値フィールドの表示方法	1 = MQ (MQMD ヘッダーのみ) WebSphere Business Integration Message Broker の資料に記載されている整数値。

注: コネクタは、MQ メッセージ内のデータの文字セット (CCSID) またはエンコード属性を制御できません。データ変換はデータがメッセージ・バッファから検索されるかメッセージ・バッファにデリバリーされる時に行われるため、コネクタはデータ変換を JMS の IBM WebSphere MQ インプリメンテーションに依存します (IBM MQSeries Java クライアント・ライブラリーの資料を参照してください)。したがって、これらの変換は、ネイティブ MQSeries または WebSphere MQ の API がオプション MQGMO_CONVERT を使用して実行する変換と双方向で等しくなければなりません。コネクタは、変換プロセスにおける差異または失敗を制御できません。コネクタは、特別な変更を必要とせず、WebSphere MQ によってサポートされるすべての CCSID またはエンコードのメッセージ・データを検索できます。特定の CCSID またはエンコードのメッセージを送信するには、出力キューが完全修飾 URI であり、CCSID および encoding の値を指定している必要があります。コネクタはこの情報を WebSphere MQ に渡し、WebSphere MQ は MQMessage のデリバリーのためにデータをエンコードするときに (JMS API を介して) この情報を使用します。多くの場合、CCSID およびエンコードのサポートの欠如は、IBM の Web サイトから最新バージョンの IBM MQSeries Java クライアント・ライブラリーをダウンロードすることによって解決できます。それでも CCSID およびエンコードに関する問題が解消されない場合は、IBM ソフトウェア・サポートに連絡し、代替の Java 仮想マシンを使用してコネクタを実行することを検討してください。

イベント・デリバリーの保証

保証付きイベント・デリバリー機能により、コネクタ・フレームワークは、コネクタのイベント・ストアから、JMS イベント・ストア、そして宛先の JMS キューまでの間で、イベントを失うことなく、また 2 度送りすることなく、確実に送達することができます。コネクタを JMS 対応にするには、コネクタの `DeliveryTransport` 標準プロパティを JMS に設定する必要があります。このように構成されたコネクタは、JMS トランスポートを使用し、コネクタと統合ブローカー間の以降の通信は、このトランスポートを介して行われます。JMS トランスポートにより、メッセージは最終的には確実に宛先に配達されます。このトランスポートの役割は、一度トランザクション・キュー・セッションが開始すると、メッセージはコミットが発行されるまで、確実にそのトランスポートのキャッシュに保持されます。障害が発生するかロールバックが発行されると、メッセージは破棄されます。

注: 保証イベント・デリバリー機能が使用されない場合、コネクタがイベントをパブリッシュしてから (コネクタがその `pollForEvents()` メソッドの内部から `gotAppEvent()` メソッドを呼び出してから) コネクタがイベント・レコードを削除することによりイベント・ストアを更新するまで (またはイベント・ストアを「イベント通知済み」状況に更新するまで)、障害を想定した小さなウィンドウが表示されます。このウィンドウで障害が発生すると、イベントは送信されますが、そのイベント・レコードは「進行中」の状況でイベント・ストアに残ります。コネクタが再始動すると、イベント・ストアにイベント・レコードが残っているため、これを送信する結果、イベントが 2 回送信されることになります。

JMS イベント・ストアを使用する JMS 対応コネクタ用、または JMS イベント・ストアを使用しない JMS 対応コネクタ用の保証付きイベント・デリバリー機能を構成することができます。保証付きイベント・デリバリーを行うようにコネクタを構成するには、「コネクタ開発ガイド (Java 用)」の説明を参照してください。

コネクタ・フレームワークがビジネス・オブジェクトを ICS 統合ブローカーに配達できない場合、オブジェクトは `FaultQueue` (`UnsubscribedQueue` や `ErrorQueue` でなく) に入り、状況表示と問題の記述を生成します。`FaultQueue` メッセージは、MQRFH2 フォーマットで書き込まれます。

第 4 章 コネクタの実行

- 『始動ファイルの構成』
- 44 ページの『コネクタの始動』
- 45 ページの『コネクタの停止』
- 46 ページの『エラー処理の概要』
- 48 ページの『トレースの概要』

この章では、コネクタ始動ファイルの構成方法と、コネクタの始動と停止の方法について説明します。

コネクタの実行の概要

コネクタは、以下の作業が完了している場合に実行できます。

- コネクタと関連ファイルのインストール
- ビジネス・オブジェクトとメタオブジェクトの構成
- データ・ハンドラーの構成
- コネクタの構成

この章で説明する作業

この章では、以下の作業について説明します。

表 12. アダプターの実行: 作業ロードマップ

作業	関連手順 (参照先)	詳細情報 (参照先)
始動ファイルの構成	『始動ファイルの構成』	「コネクタ開発ガイド」
コネクタの始動	44 ページの『コネクタの始動』	
コネクタの停止	45 ページの『コネクタの停止』	60 ページの『データ・ハンドラーの入力キューへのマッピングの概要』

始動ファイルの構成

WebSphere Business Integration Message Broker 対応コネクタを開始する前に、始動ファイルを構成しておく必要があります。

Windows プラットフォーム用の始動ファイルの構成のステップ

Windows プラットフォーム用のコネクタの構成を完了するためには、次のようにして start_WBIMB.bat ファイルを変更する必要があります。

1. start_WBIMB.bat ファイルを開きます。
2. スクロールして「Set the directory containing your WebSphereMQ Java client libraries」で始まるセクションに移動し、WebSphere MQ Java クライアント・ライブラリーの場所を指定します。

UNIX プラットフォーム用の始動ファイルの構成のステップ

UNIX プラットフォーム用のコネクタの構成を完成させるには、`start_WBIMB.sh` ファイルを変更する必要があります。

1. `start_WBIMB.sh` ファイルを開きます。
2. スクロールして「Set the directory containing your WebSphere MQ Java client libraries」で始まるセクションに移動し、WebSphere MQ Java クライアント・ライブラリーの場所を指定します。

コネクタの始動

コネクタは、**コネクタ始動スクリプト**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のランタイム・ディレクトリに存在していなければなりません。

```
ProductDir%connectors%connName
```

ここで、`connName` はコネクタを示します。始動スクリプトの名前は、表 13 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 13. コネクタの始動スクリプト

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	<code>connector_manager_connName</code>
Windows	<code>start_connName.bat</code>

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。

コネクタの停止

コネクタを停止する方法は、以下に示すように、コネクタ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を始動した場合は、コネクタ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、コネクタが停止します。
 - UNIX ベースのシステムでは、コネクタ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を停止します。

```
connector_manager_connName -stop
```

ここで、`connName` はコネクタの名前です。

- 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、コネクターのロード、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ー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、コネクターは停止します。

エラー処理の概要

コネクターによって生成されたすべてのエラー・メッセージは、MQSIV2Connector.txt という名前のメッセージ・ファイルに格納されます。(このファイルの名前は、コネクター構成標準プロパティ `LogFileNames` によって決定されます。) 各エラーはエラー番号が付けられ、その後にエラー・メッセージが表示されます。

Message number
Message text

コネクターは、以降のセクションで説明する方法で特定のエラーを処理します。

アプリケーション・タイムアウト

エラー・メッセージ `ABON_APPRESPONSETIMEOUT` は、以下の場合に戻されません。

- コネクターは、メッセージの検索中に JMS サービス・プロバイダーとの接続を確立できませんでした。
- コネクターはビジネス・オブジェクトをメッセージに正常に変換しましたが、接続切断が原因でメッセージを出力キューにデリバリーできませんでした。
- コネクターはメッセージを発行しましたが、変換プロパティ `TimeoutFatal` が `True` であるビジネス・オブジェクトに対する応答の待機中にタイムアウトが発生しました。
- コネクターは、戻りコードが `APP_RESPONSE_TIMEOUT` または `UNABLE_TO_LOGIN` の応答メッセージを受信しました。

アンサブスクライブされたビジネス・オブジェクト

コネクターは、次の場合に、`UnsubscribedQueue` プロパティで指定されたキューにメッセージをデリバリーします。

- アンサブスクライブされたビジネス・オブジェクトに関連付けられているメッセージをコネクターが検索した場合、または `getAppEvent()` メソッドによって `NO_SUBSCRIPTION_FOUND` コードが戻された場合。
- コネクターがメッセージを検索したが、`FORMAT` フィールドのテキストをビジネス・オブジェクト名に関連付けることができない場合。

注: `UnsubscribedQueue` が定義されていない場合、アンサブスクライブされたメッセージは破棄されます。

コネクタがアクティブでない

gotApplEvent() メソッドが CONNECTOR_NOT_ACTIVE コードを戻すと、pollForEvents() メソッドは APP_RESPONSE_TIMEOUT コードを戻し、イベントは InProgress キューに置かれたままになります。

データ・ハンドラーによる変換

データ・ハンドラーがメッセージからビジネス・オブジェクトへの変換に失敗した場合、または、(JMS プロバイダーではなく) ビジネス・オブジェクトに固有の処理エラーが発生した場合、メッセージは ErrorQueue で指定されたキューにデリバリーされます。ErrorQueue が定義されていない場合、エラーが原因で処理できなかったメッセージは破棄されます。

データ・ハンドラーがビジネス・オブジェクトからメッセージへの変換に失敗した場合、BON_FAIL が戻されます。

JMS プロパティ

メッセージに対して JMS プロパティの読み取りまたは書き込みを実行できない場合、コネクタはエラーをログに記録し、要求またはイベントは失敗します。ReplyToQueue が未指定またはアクセス不可の場合、コネクタはエラーをログに記録し、要求は失敗します。CorrelationID が無効であるか設定できない場合、コネクタはエラーをログに記録し、要求は失敗します。

いずれの場合も、ログに記録されるメッセージはコネクタ・メッセージ・ファイルから読み込まれます。

入力フォーマットの多重定義

コネクタは通常、メッセージを検索するときに、ビジネス・オブジェクトと動詞の特定の 1 つの組み合わせと入力フォーマットを突き合わせます。そして、コネクタはビジネス・オブジェクト名とメッセージの内容をデータ・ハンドラーに渡します。これにより、データ・ハンドラーは、メッセージの内容がユーザーの予期するビジネス・オブジェクトに対応していることを確認できます。

しかし、複数のビジネス・オブジェクトに対して同じ入力フォーマットが定義されている場合、コネクタは、データ・ハンドラーにデータを渡す前に、データがどのビジネス・オブジェクトを表すのかを判別できません。そのような場合、コネクタはデータ・ハンドラーにメッセージの内容のみを渡し、生成されるビジネス・オブジェクトに基づいて変換プロパティを調べます。したがって、データ・ハンドラーは、メッセージの内容のみに基づいてビジネス・オブジェクトを決定する必要があります。

生成されたビジネス・オブジェクトに動詞が設定されていない場合、コネクタはビジネス・オブジェクトの動詞の種類を限定せずに、このビジネス・オブジェクトに定義されている変換プロパティを検索します。変換プロパティが 1 セットのみ検出された場合は、コネクタは特定の動詞を割り当てます。複数のプロパティが検出された場合は、コネクタは動詞を判別できないため、メッセージの処理が失敗します。

トレースの概要

トレースはオプションのデバッグ機能で、オンにするとコネクターの動作を詳細にトレースできます。デフォルトでは、トレース・メッセージは `STDOUT` に書き込まれます。トレース・メッセージの構成の詳細については、第 2 章のコネクター構成プロパティについての説明を参照してください。トレースの使用可能化や設定方法などの詳細については、「コネクター開発ガイド (Java 用)」を参照してください。

コネクターのトレース・メッセージに推奨される内容を以下に示します。

- レベル 0 コネクターのバージョンを確認するトレース・メッセージに使用します。
- レベル 1 処理される各ビジネス・オブジェクトについての重要な情報を提供するトレース・メッセージや、ポーリング・スレッドが入力キュー内で新しいメッセージを検出するたびに記録されるトレース・メッセージに使用します。
- レベル 2 ビジネス・オブジェクトが `gotAppEvent()` または `executeCollaboration()` から `InterChange Server` に通知されるたびに記録されるトレース・メッセージに使用します。
- レベル 3 データ・フォーマット (例えば XML) からビジネス・オブジェクトへの変換およびビジネス・オブジェクトからデータ・フォーマットへの変換に関する情報を提供するトレース・メッセージや、出力キューへのメッセージのデリバリーに関する情報を提供するトレース・メッセージに使用します。
- レベル 4 コネクターが動作を開始または終了した時間を記録するトレース・メッセージに使用します。
- レベル 5 コネクターの初期化を示すトレース・メッセージ、アプリケーション内で実行されるステートメントを表すトレース・メッセージ、メッセージが除去されるかキューに送出されるたびに記録されるトレース・メッセージ、または、ビジネス・オブジェクトのダンプを記録するトレース・メッセージに使用します。

第 5 章 オブジェクトの作成

- 『オブジェクトの作成の概要』
- 50 ページの『ビジネス・オブジェクトの作成』
- 51 ページの『ビジネス・オブジェクトの変更』
- 53 ページの『メタオブジェクトの作成』

この章では、ビジネス・オブジェクトとメタオブジェクトの作成および変更に必要な作業について説明します。コネクターを正常に機能させるには、これらの 2 種類のオブジェクトの両方が必要です。

オブジェクトの作成の概要

以下の 2 種類のオブジェクトの定義を作成する必要があります。

- **ビジネス・オブジェクト** ビジネス・エンティティ（従業員やカタログ項目など）またはデータ・トランザクション（作成、更新、削除など）を表現します。データ処理に関するアプリケーション固有の指示も含まれています。コネクターにはビジネス・オブジェクトのサンプルのみが付属しています。これらのサンプルはチュートリアルで使用します。ご使用のコネクターに適したビジネス・オブジェクト定義をビルドする必要があります。
- **メタオブジェクト** 構成メタオブジェクトは、データ・フォーマット、キュー、マッピング、メッセージ・ヘッダー情報、および応答時間を定義します。これらの情報は、コネクターがビジネス・オブジェクトとメッセージを交換するときの動作を左右します。コネクターには、メタオブジェクトのサンプルのみが付属しています。このサンプルを変更してアダプターで使用できるようにしても、独自のメタオブジェクトを作成してもかまいません。メタオブジェクトは、プロパティを使用して構成する必要があります。この構成を行わないと、コネクターが正常に機能しません。

この章で説明する作業

この章で説明する作業は、以下のとおりです。

表 14. オブジェクトの作成: 作業ロードマップ

作業	関連手順 (参照先)	詳細情報 (参照先)
ビジネス・オブジェクトの作成	50 ページの『ビジネス・オブジェクトの作成』	「ビジネス・オブジェクト開発ガイド」
ビジネス・オブジェクトの変更	51 ページの『ビジネス・オブジェクトの変更』	「ビジネス・オブジェクト開発ガイド」
メタオブジェクトの作成	53 ページの『メタオブジェクトの作成』	58 ページの『静的メタオブジェクトの作成の概要』、60 ページの『動的子メタオブジェクトの作成の概要』
データ・ハンドラーの入力キューへのマッピング	60 ページの『データ・ハンドラーの入力キューへのマッピングのステップ』	60 ページの『データ・ハンドラーの入力キューへのマッピングの概要』

ビジネス・オブジェクトの作成

このトピックでは、ビジネス・オブジェクト定義の作成の概要と手順を示します。

ビジネス・オブジェクトの作成の概要

コネクタをインストールした後は、ビジネス・オブジェクト定義を作成する必要があります。構成するデータ・ハンドラーやメタオブジェクトの属性に応じて求められる要件以外に、ビジネス・オブジェクトの構造に関する要件はありません。コネクタで使用するデータ・フォーマットが決定したら、適切なデータ・ハンドラーを構成し、そのフォーマットに準拠したビジネス・オブジェクトを生成します。

WBI Message Broker は XML メッセージをネイティブにサポートしています。そのため、Adapter for WBI Message Broker は、デフォルトでは XML データ・ハンドラーに対応するように構成されます。つまり、このアダプターは、ビジネス・オブジェクトからメッセージへの変換およびメッセージからビジネス・オブジェクトへの変換に XML データ・ハンドラーを使用します。

注: メッセージやビジネス・オブジェクトの要件をサポートするために必要な場合には、XML データ・ハンドラー以外の使用可能なデータ・ハンドラーを使用したり、カスタム・データ・ハンドラーを作成したりすることができます。この場合、そのデータ・ハンドラーをコネクタで使えるように構成し、メッセージ・フローをそれに応じて変更する必要があります。使用可能なデータ・ハンドラーとカスタム・データ・ハンドラーの詳細については、「データ・ハンドラー・ガイド」を参照してください。データ・ハンドラーをアダプターで使えるように構成する方法と、メッセージ・フローを変更する方法の詳細については、67 ページの『第 6 章 データ・ハンドラーの構成』を参照してください。

XML データ・ハンドラーとともに使用するビジネス・オブジェクトを構成するには、XML Object Discovery Agent (ODA) を使用します。XML ODA は、XML 文書の DTD またはスキーマ文書に基づいて、その XML 文書に対応するビジネス・オブジェクト定義を生成します。構成されたビジネス・オブジェクトは、アダプターで使用できます。Adapter for WBI Message Broker に付属しているサンプルには、ビジネス・オブジェクトの変換に XML データ・ハンドラーを使用するように構成されているファイルが含まれています。

注: コネクタが処理するビジネス・オブジェクトは、統合ブローカーによって許可されている任意の名前を持つことができます。InterChange Server で許可されている命名規則の詳細については、「*Naming IBM WebSphere InterChange Server Components*」を参照してください。

ビジネス・オブジェクトの作成のステップ

始める前に: コネクタが交換するメッセージとビジネス・オブジェクトのデータ・フォーマット (XML など) を確認する必要があります。

コネクタに適したビジネス・オブジェクトを作成するには、次のステップを実行します。

1. **Object Discovery Agent (ODA)** を使用してビジネス・オブジェクト定義を生成します (ODA が使用可能な場合)。例えば、XML メッセージを交換するように構成されているコネクターに適したビジネス・オブジェクトを作成するには、XML ODA を使用します。詳細については、「データ・ハンドラー・ガイド」を参照してください。
2. 必要に応じて、**Business Object Designer** を使用して、ODA が生成した情報に変更や追加情報を加えます。詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

注: 使用するデータ・フォーマットに対応する ODA がない場合には、**Business Object Designer** を使用してビジネス・オブジェクト定義を作成することができません。詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

ビジネス・オブジェクト定義を作成した後は、**Connector Configurator** を使用して、それらのビジネス・オブジェクト定義を、コネクターがサポートするビジネス・オブジェクト定義のリストに追加します。Connector Configurator の使用についての詳細は、67 ページの『第 6 章 データ・ハンドラーの構成』を参照してください。

ビジネス・オブジェクトの変更

ビジネス・オブジェクトを変更すると、各種処理機能を活用することができます。ビジネス・オブジェクトを変更するときは、**Business Object Designer** を使用します。詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

このトピックでは、ビジネス・オブジェクト定義を変更して同期要求処理機能を使用可能にする作業の概要と手順を示します。

メッセージ選択子による応答メッセージのフィルター操作の概要

同期要求処理の場合にビジネス・オブジェクトを受信すると、コネクターは動詞のアプリケーション固有情報に `response_selector` スtringが含まれているかどうかを検査します。`response_selector` が未定義の場合、コネクターは相関 ID を使用して応答メッセージを識別します (8 ページの『Retrieve、Exists、および Retrieve by content』を参照)。

`response_selector` が定義されている場合、コネクターは以下の構文を持つ名前と値のペアを期待します。

```
response_selector=JMSCorrelationID LIKE'selectorstring'
```

メッセージ選択子 `selectorstring` は応答を一意的に識別する必要があります。また、値は以下に示すように単一引用符で囲む必要があります。

```
response_selector=JMSCorrelationID LIKE 'Oshkosh'
```

上記の例では、要求メッセージの発行後に、アダプターは「Oshkosh」と等しい `correlationID` を持つ応答メッセージの `ReplyToQueue` をモニターします。アダプターはこのメッセージ選択子に一致する最初のメッセージを検索し、これを応答としてディスパッチします。

メッセージ選択子による応答メッセージのフィルター操作のステップ

メッセージ選択子を使用して応答メッセージをフィルター操作するには、以下の手順を実行します。

1. **Business Object Designer** を起動します。
2. 同期メッセージ・デリバリーの対象として変換されるようにするビジネス・オブジェクトを開きます。
3. 動詞のアプリケーション固有情報に、「`response_selector=JMSCorrelationID LIKE 'selectorstring'`」と指定します。
selectorstring は、応答を一意的に識別するためのストリングです。*selectorstring* に指定されたストリングを相関 ID とする応答がアプリケーションから戻されると、コネクタはその応答を同期要求の応答として識別します。コネクタは、その後、データ・ハンドラーを呼び出してその応答を応答ビジネス・オブジェクトに変換し、要求元のコラボレーションに戻します。

注: *selectorstring* には複数のストリングを指定することができます。また、特殊文字を指定したり、静的メタオブジェクトや動的メタオブジェクトの属性を参照したり、*selectorstring* に指定したストリングを置換できるようにしたりすることもできます。詳細については、以下の例を参照してください。

例: 必要に応じて、アダプターによる実行時置換を実行して、それぞれの要求ごとに固有のメッセージ選択子を生成することもできます。メッセージ選択子の代わりに、整数を中括弧で囲んだ形式でプレースホルダーを指定することもできます。例えば、'`{1}`' のようになります。次に、後ろにコロンを付け、置換に使用する属性をコンマで区切ってリストします。プレースホルダー内の整数は、置換に使用する属性の索引の役割を持ちます。例えば、以下のメッセージ選択子の場合:

```
response_selector=JMSCorrelationID LIKE '{1}': MyDynamicMO.CorrelationID
```

このメッセージ選択子は、アダプターに `{1}` を選択子の後ろの最初の属性の値 (この場合は、子オブジェクト `MyDynamicMO` の `CorrelationId` という属性) に置換するよう通知します。属性 `CorrelationID` の値が `123ABC` であった場合、アダプターは以下の基準によって作成されたメッセージ選択子を生成および使用します。

```
JMSCorrelation LIKE '123ABC'
```

これによって応答メッセージを識別します。

また、以下のように複数の置換対象を指定することもできます。

```
response_selector=PrimaryId LIKE '{1}' AND AddressId LIKE '{2}' :  
PrimaryId, Address[4].AddressId
```

この例では、アダプターは {1} をトップレベル・ビジネス・オブジェクトの PrimaryId 属性の値に置換し、{2} を子コンテナ・オブジェクト Address の 5 番目の位置の AddressId の値に置換します。この方法を使用すると、応答メッセージ選択子内にあるビジネス・オブジェクトおよびメタオブジェクトのどの属性でも参照できます。Address[4].AddressId を使用した詳細検索の実行方法に関する詳細は、JCDK API のマニュアル (getAttribute メソッド) を参照してください。

以下のいずれかの状況が発生した場合、実行時にエラーが報告されます。

- 「{ }」シンボルの間に非整数を指定した場合
- 属性が定義されていない索引を指定した場合
- 指定された属性がビジネス・オブジェクトまたはメタオブジェクトに存在しない場合
- 属性パスの構文が誤っている場合

例えば、メッセージ選択子にリテラル値「{ }」または「:」を組み込む場合は、それぞれ「{{ }」または「{: }」を使用できます。また、これらの文字を属性値に組み込むこともできます。その場合、最初の「{」は不要です。エスケープ文字を使用した例を以下に示します。

```
response_selector=JMSCorrelation LIKE '{1}' and CompanyName='A{P':  
MyDynamicMO.CorrelationID
```

コネクターはこのメッセージ選択子を以下のように解決します。

```
JMSCorrelationID LIKE '123ABC' and CompanyName='A{P'
```

コネクターが属性値内で特殊文字 (「{」、「}」、「:」、または「;」など) を検出すると、これらの文字は照会ストリングに直接挿入されます。これにより、アプリケーション固有情報の区切り文字の役割も持つ特殊文字を照会ストリングに組み込むことができます。

次の例は、リテラル・ストリングの置換値が属性値からどのように抽出されるかを示します。

```
response_selector=JMSCorrelation LIKE '{1}' and CompanyName='A{P':  
MyDynamicMO.CorrelationID
```

MyDynamicMO.CorrelationID に値 {A:B}C;D が含まれていた場合、コネクターはメッセージ選択子を以下のように解決します。

```
JMSCorrelationID LIKE '{A:B}C;D' and CompanyName='A{P'
```

応答選択子コードに関する詳細は、JMS 1.0.1 仕様を参照してください。

メタオブジェクトの作成

このトピックでは、メタオブジェクトの作成の概要と手順を示します。

コネクターは、メタオブジェクト・エントリーを使用して、メッセージに関連付けるビジネス・オブジェクトを決定します。メッセージの処理に使用されるビジネス・オブジェクトのタイプと動詞は、WebSphere MQ メッセージのヘッダーに含まれる FORMAT フィールドによって決定されます。ビジネス・オブジェクト名と動詞

を格納するメタオブジェクト属性を構成し、WebSphere MQ メッセージ・ヘッダーの `FORMAT` フィールドのテキストに関連付けます。メタオブジェクト属性には、メッセージ処理のガイドラインも含まれます。

入力キューからメッセージが検索されると、コネクタは、`FORMAT` テキスト・フィールドに関連付けられているビジネス・オブジェクト名を調べます。次に、ビジネス・オブジェクト名とともに、メッセージがデータ・ハンドラーに渡されます。ビジネス・オブジェクトにメッセージの内容が正常に取り込まれると、コネクタはそのビジネス・オブジェクトがサブスクライブされているかどうかをチェックしてから、`gotApplEvents()` メソッドを使用して統合ブローカーにデリバリーします。

WebSphere Business Integration Message Broker 対応コネクタは、次の 2 種類のメタオブジェクトを認識および読み取りできます。

- 静的コネクタ・メタオブジェクト
- 動的子メタオブジェクト

動的子メタオブジェクトの属性値は、静的メタオブジェクトの属性値を複製およびオーバーライドします。

ご自分の実装にはどちらのメタオブジェクトが最適であるかを判断する際には、以下のことを考慮してください。

- **静的メタオブジェクト**
 - 各種メッセージのメタデータがすべて固定されており、かつ構成時に指定可能である場合に役立ちます。
 - ビジネス・オブジェクト・タイプごとにしか値を指定できません。例えば、`Customer` タイプのオブジェクトのすべてが同一の宛先に送られます。
- **動的メタオブジェクト**
 - ビジネス・プロセスからメッセージ・ヘッダー内の情報にアクセスできます。
 - ビジネス・オブジェクト・タイプに関係なく、実行時にビジネス・プロセスでメッセージの処理を変更できます。例えば、動的メタオブジェクトを使用すると、アダプターに送信された `Customer` タイプのオブジェクトのそれぞれに、別々の宛先を指定することができます。
 - サポートするビジネス・オブジェクトの構造を変更する必要があります。この変更によって、マップとビジネス・プロセスの変更が必要になる場合があります。
 - カスタム・データ・ハンドラーに変更を加える必要があります。

メタオブジェクト・プロパティ

表 15 に、メタオブジェクトでサポートされるプロパティをすべて含むリストを示します。メタオブジェクトを実装するときには、これらのプロパティの説明を参照してください。メタオブジェクトには、表 15 に示すプロパティが 1 つ以上含まれていなければなりません。

一部のプロパティは、静的メタオブジェクトと動的メタオブジェクトのいずれかでしか使用できません。また、メッセージ・ヘッダーからの読み取りやメッセージ・ヘッダーへの書き込みが不可能なプロパティもあります。特定のプロパティ

ーについて、コネクタでどのように解釈および使用されるかを判断するには、1 ページの『第 1 章 概要』のイベント処理と要求処理に関する適切なセクションを参照してください。

表 15. Adapter for WBI Message Broker のメタオブジェクト・プロパティ

プロパティ名	静的メタオブジェクトで定義可能か	動的メタオブジェクトで定義可能か	説明
DataHandlerConfigMO	はい	はい	構成情報を提供するためにデータ・ハンドラーに渡されるメタオブジェクトです。静的メタオブジェクトに指定されている場合、この値は DataHandlerConfigMO コネクタ・プロパティに指定されている値をオーバーライドします。このメタオブジェクト・プロパティは、さまざまなタイプのビジネス・オブジェクトを処理するために複数の異なるデータ・ハンドラーが必要な場合に使用します。データ・フォーマットが実際のビジネス・データに依存する可能性がある場合は、要求処理に動的子メタオブジェクトを使用してください。指定するビジネス・オブジェクトは、コネクタ・エージェントでサポートされるものでなければなりません。93 ページの『付録 B. Adapter for WBI Message Broker のコネクタ固有のプロパティ』の説明を参照してください。
DataHandlerMimeType	はい	はい	使用すると、特定の MIME タイプに基づいたデータ・ハンドラーを要求できます。メタオブジェクトに指定されている場合、この値は DataHandlerMimeType コネクタ・プロパティに指定されている値をオーバーライドします。このメタオブジェクト・プロパティは、さまざまなタイプのビジネス・オブジェクトを処理するために複数の異なるデータ・ハンドラーが必要な場合に使用します。データ・フォーマットが実際のビジネス・データに依存する可能性がある場合は、要求処理に動的子メタオブジェクトを使用してください。DataHandlerConfigMO に指定されているビジネス・オブジェクトに、このプロパティの値に対応する属性が含まれていることが必要です。93 ページの『付録 B. Adapter for WBI Message Broker のコネクタ固有のプロパティ』の説明を参照してください。
DataHandlerClassName	はい	はい	93 ページの『付録 B. Adapter for WBI Message Broker のコネクタ固有のプロパティ』の説明を参照してください。

表 15. Adapter for WBI Message Broker のメタオブジェクト・プロパティ (続き)

プロパティ名	静的メタオブジェクトで定義可能か	動的メタオブジェクトで定義可能か	説明
InputFormat	はい	はい	特定のビジネス・オブジェクトに関連付けるインバウンド (イベント) メッセージのフォーマットまたはタイプです。この値は、メッセージ内容の識別に役立つものであり、メッセージを生成するアプリケーションによって決まります。検索されたメッセージがこのフォーマットである場合、そのメッセージは (可能であれば) 特定のビジネス・オブジェクトに変換されます。ビジネス・オブジェクトにこのフォーマットが指定されていない場合、コネクタは特定のビジネス・オブジェクトのサブスクリプション・デリバリーを処理しません。このプロパティを設定するときは、デフォルトのメタオブジェクト変換プロパティを使用しないでください。デフォルトのメタオブジェクト変換プロパティの値は、着信メッセージをビジネス・オブジェクトに一致させるために使用されます。コネクタがメッセージのフォーマットを定義していると見なすフィールドは、コネクタ固有のプロパティ <code>MessageFormatProperty</code> を使用してユーザーが定義できます。
OutputFormat	はい	はい	アウトバウンド・メッセージに取り込まれるフォーマットです。OutputFormat が指定されていない場合は、入力フォーマットが (利用可能であれば) 使用されます。
InputQueue	はい	はい	新規メッセージを検出するためにコネクタがポーリングする入力キューです。このプロパティは、着信メッセージとビジネス・オブジェクトを一致させる目的でのみ使用されます。このプロパティを設定するときは、デフォルトの変換プロパティを使用しないでください。デフォルトの変換プロパティの値は、着信メッセージをビジネス・オブジェクトに一致させるために使用されます。 注: コネクタ固有のプロパティにある InputQueue プロパティは、アダプターがポーリングするキューを定義します。これは、アダプターがポーリングするキューを決定するのに使用する唯一のプロパティです。静的 MO では、InputQueue プロパティおよび InputFormat プロパティは、アダプターが指定されたメッセージを特定のビジネス・オブジェクトにマップする条件として使用できます。この機能を実装するには、コネクタ固有のプロパティを使用して複数の入力宛先を構成し、さらに、必要に応じて、着信メッセージの入力フォーマットを基に各入力宛先に別個のデータ・ハンドラーをマップします。詳細については、60 ページの『データ・ハンドラーの入力キューへのマッピングの概要』を参照してください。
OutputQueue	はい	はい	特定のビジネス・オブジェクトから派生したメッセージが送信されるキューです。

表 15. Adapter for WBI Message Broker のメタオブジェクト・プロパティ (続き)

プロパティ名	静的メタオブジェクトで定義可能か	動的メタオブジェクトで定義可能か	説明
ResponseTimeout	はい	はい	同期要求処理で応答を待機するときにタイムアウトとするまでの待機時間を、ミリ秒単位で示します。このプロパティが未定義のままか、またはゼロよりも小さい値に設定されている場合、コネクタは応答を待機せず、SUCCESS を即時に戻します。
TimeoutFatal	はい	はい	同期要求処理で、応答の受信がないためコネクタからエラー・メッセージを戻す動作が開始されるときに使用されます。このプロパティの値が True の場合、コネクタは、ResponseTimeout に指定されている時間内に応答の受信がなければ、APPRESPONSETIMEOUT をブローカーに戻します。このプロパティが未定義の場合、または False に設定されている場合、コネクタは応答タイムアウトが発生すると要求を失敗させます。ただし、終了させることはありません。デフォルト値は False です。
以下に示すのは、JMS メッセージ・ヘッダーのみにマップされるフィールドです。具体的な説明や値の解釈などの詳細情報については、JMS API の仕様を参照してください。フィールドによっては、JMS プロバイダー間で解釈が異なることがあります。ご使用の JMS プロバイダーの資料を参照して、解釈の違いがないかどうかを確認してください。			
ReplyToDestination		はい	要求に対する応答メッセージの送信先となる宛先です。
Type		はい	メッセージのタイプです。通常はユーザーが定義できます (JMS プロバイダーによって異なります)。
MessageID		はい	メッセージの固有 ID です (特定の JMS プロバイダーでのみ有効)。
CorrelationID	はい	はい	応答メッセージで、その応答の開始理由にあたる要求メッセージの ID を示すために使用されます。
Delivery Mode	はい	はい	メッセージを MOM システム内で永続化するかどうかを指定します。許容値は次のとおりです。 1 = 非永続 2 = 永続 その他の値も使用できる場合があります (JMS プロバイダーによって異なります)。
Priority		はい	メッセージの優先順位を数値で表現したものです。許容値は 0 から 9 です (数値が大きいほど優先順位が高まります)。
Destination		はい	MOM システム内でのメッセージの現在の位置 (除去済みの場合は除去直前の位置) です。
Expiration		はい	メッセージの存続時間です。
Redelivered		はい	以前に JMS プロバイダーからクライアントに対してメッセージのデリバリーが試行されたが、受取の確認がなかった可能性が高いことを示すものです。
Timestamp		はい	メッセージが JMS プロバイダーに渡された時刻です。
UserID		はい	メッセージを送信したユーザーの ID です。
AppID		はい	メッセージを送信したアプリケーションの ID です。
DeliveryCount		はい	デリバリーを試行した回数です。
GroupID		はい	メッセージ・グループの ID です。
GroupSeq		はい	GroupID に指定されたメッセージ・グループにおける順位です。

表 15. Adapter for WBI Message Broker のメタオブジェクト・プロパティ (続き)

プロパティ名	静的メタオブジェクトで定義可能か	動的メタオブジェクトで定義可能か	説明
JMSProperties	はい		62 ページの『JMS プロパティ』を参照してください。

静的メタオブジェクトの作成の概要

WebSphere Business Integration Message Broker の構成メタオブジェクトは、さまざまなビジネス・オブジェクトに定義される変換プロパティのリストで構成されます。静的メタオブジェクトのサンプルを参照するには、Business Object Designer を起動し、アダプターに付属しているサンプル `connectors\WBIMB\samples\LegacyItem\Sample_WBIMB_MO_Config.xsd` を開きます。

コネクターがサポートする静的メタオブジェクトは、常に 1 つだけです。静的メタオブジェクトを実装するには、そのメタオブジェクトの名前を、コネクター・プロパティ `ConfigurationMetaObject` に指定します。

静的メタオブジェクトは、各属性が、それぞれ 1 つのビジネス・オブジェクトと動詞の組み合わせを、そのオブジェクトの処理に関連するメタデータのすべてとともに示す構造になっています。各属性の名前は、`Customer_Create` のように、ビジネス・オブジェクト・タイプと動詞の間を下線で区切った名前にする必要があります。属性のアプリケーション固有情報には、このオブジェクトと動詞の固有の組み合わせに対して指定するメタデータ・プロパティを表す名前と値のペアを、セミコロンで区切って 1 つ以上含める必要があります。

表 16. 静的メタオブジェクト構造

属性名	アプリケーション固有のテキスト
<code><business object type>_<verb></code>	<code>property=value;property=value;...</code>
<code><business object type>_<verb></code>	<code>property=value;property=value;...</code>

例えば、次のようなメタオブジェクトがあるとします。

表 17. 静的メタオブジェクト構造のサンプル

属性名	アプリケーション固有の情報
<code>Customer_Create</code>	<code>OutputFormat=CUST;OutputDestination=QueueA</code>
<code>Customer_Update</code>	<code>OutputFormat=CUST;OutputDestination=QueueB</code>
<code>Order_Create</code>	<code>OutputFormat=ORDER;OutputDestination=QueueC</code>

このサンプルのメタオブジェクトは、タイプが `Customer` で動詞が `Create` の要求ビジネス・オブジェクトを受信したときは、そのオブジェクトをフォーマットが `CUST` のメッセージに変換して宛先 `QueueA` に入れる必要があることを、コネクターに対して通知します。Cutomer タイプのオブジェクトの動詞が `Update` である場合、変換後のメッセージは `QueueB` に入れられます。オブジェクトのタイプが `Order` で動詞が `Create` である場合には、コネクターはそのオブジェクトをフォー

マップが ORDER のメッセージに変換し、QueueC にデリバリーします。それ以外のビジネス・オブジェクトがコネクタに渡された場合には、サブスクライブされていないものとして処理されます。

また、Default 属性を 1 つ指定して、アプリケーション固有情報 (ASI) のプロパティを 1 つ以上割り当てることもできます。このデフォルト属性のプロパティと、オブジェクトと動詞の組み合わせごとの属性のプロパティが結合されて、メタオブジェクトの属性の最終的なプロパティになります。オブジェクトと動詞の組み合わせに関係なく汎用的に適用するプロパティが 1 つ以上ある場合には、このデフォルト属性を使用すると便利です。次の例の場合、コネクタは、オブジェクトと動詞の組み合わせ Customer_Create と Order_Create については、それぞれに個別に指定されているメタオブジェクト・プロパティに加えて OutputDestination=QueueA が指定されていると見なします。

表 18. 静的メタオブジェクト構造のサンプル

属性名	アプリケーション固有の情報
Default	OutputDestination=QueueA
Customer_Update	OutputFormat=CUST
Order_Create	OutputFormat=ORDER

静的メタオブジェクトにアプリケーション固有情報として指定できるプロパティは、55 ページの表 15 に記載されています。

注: 静的メタオブジェクトが指定されていない場合、コネクタはポーリング時にメッセージ・フォーマットを特定のビジネス・オブジェクト・タイプにマップできません。そのような場合、コネクタは、ビジネス・オブジェクトを指定せずに、構成されているデータ・ハンドラーにメッセージ・テキストを渡します。データ・ハンドラーがテキストのみに基づいてビジネス・オブジェクトを作成することができない場合、コネクタはこのメッセージ・フォーマットを認識できないことを示すエラーを報告します。

静的メタオブジェクトの作成のステップ

静的メタオブジェクトを実装するには、以下の手順を実行します。

1. Business Object Designer を起動します。詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。
2. メタオブジェクトのサンプル connectors\WBIMB\samples\LegacyItem\Sample_WBIMB_MO_Config.xsd を開きます。
3. 必要な要件が反映されるように属性と ASI を編集して (55 ページの表 15 を参照)、メタオブジェクト・ファイルを保管します。
4. 保管したメタオブジェクト・ファイルの名前を、コネクタ・プロパティ ConfigurationMetaObject の値に指定します。

データ・ハンドラーの入力キューへのマッピングの概要

静的メタオブジェクトのアプリケーション固有情報内の `InputQueue` プロパティを使用して、データ・ハンドラーを入力キューに関連付けることができます。この機能は、フォーマットや型変換の要件が異なる複数の取引先との取引を行なう際に役立ちます。

データ・ハンドラーの入力キューへのマッピングのステップ

データ・ハンドラーを `InputQueue` にマップするには、以下の手順を実行します。

1. コネクター固有のプロパティ（96 ページの『`InputQueue`』を参照）を使用して、1 つ以上の入力キューを構成します。
2. **Business Object Designer** で、適切な静的メタオブジェクトを開きます。
3. 静的メタオブジェクトのそれぞれの入力キューについて、アプリケーション固有情報に、キュー・マネージャー、入力キュー名、データ・ハンドラーのクラス名、および MIME タイプを指定します。

例えば、次に示す静的メタオブジェクト内の属性は、データ・ハンドラーを `CompReceipts` という名前の `InputQueue` に関連付けます。

```
[Attribute]
Name = Cust_Create
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = InputQueue=//queue.manager/CompReceipts;DataHandlerClassName=
com.crossworlds.DataHandlers.WBIMB.disposition_notification;DataHandlerMimeType=
message/
disposition_notification
IsRequiredServerBound = false
[End]
```

動的子メタオブジェクトの作成の概要

静的メタオブジェクトを使用して必要なメタデータを指定することが困難または実行不可能な場合には、実行時に、ビジネス・オブジェクト・インスタンスごとにデリバリーされるメタデータをコネクターで受信することもできます。

動的メタオブジェクトを使用すると、要求処理では、コネクターがビジネス・オブジェクトの処理に使用するメタデータを要求ごとに変更できるようになります。また、イベント処理では、イベント・メッセージに関する情報の検索が可能になります。

コネクターは、コネクターに渡されるトップレベルのビジネス・オブジェクトに子として追加された動的メタオブジェクトから、変換プロパティを認識して読み取ります。動的子メタオブジェクトの属性値は、コネクターの構成に使用される静的メタオブジェクトによって指定できる変換プロパティを複製します。

動的子メタオブジェクト・プロパティは静的メタオブジェクト・プロパティの値をオーバーライドするため、動的子メタオブジェクトを指定する場合は、静的メタオブジェクトを指定するコネクター・プロパティを含める必要はありません。

したがって、動的子メタオブジェクトは静的メタオブジェクトから独立して使用でき、また、静的メタオブジェクトは動的子メタオブジェクトから独立して使用できません。

動的メタオブジェクトにアプリケーション固有情報として指定できるプロパティは、55 ページの表 15 に記載されています。

以下の属性は JMS および WebSphere MQ ヘッダー・プロパティを反映しており、動的メタオブジェクトで認識されます。

表 19. 動的メタオブジェクト・ヘッダー属性

ヘッダー属性名	モード	対応する JMS ヘッダー
CorrelationID	読み取り/書き込み	JMSCorrelationID
ReplyToQueue	読み取り/書き込み	JMSReplyTo
DeliveryMode	読み取り/書き込み	JMSDeliveryMode
Priority	読み取り/書き込み	JMSPriority
Destination	読み取り	JMSDestination
Expiration	読み取り	JMSExpiration
MessageID	読み取り	JMSMessageID
Redelivered	読み取り	JMSRedelivered
TimeStamp	読み取り	JMSTimeStamp
Type	読み取り	JMSType
UserID	読み取り	JMSXUserID
AppID	読み取り	JMSXAppID
DeliveryCount	読み取り	JMSXDeliveryCount
GroupID	読み取り	JMSXGroupID
GroupSeq	読み取り	JMSXGroupSeq
JMSProperties	読み取り/書き込み	

読み取り専用属性は、イベント通知中にメッセージ・ヘッダーから読み取られ、動的メタオブジェクトに書き込まれます。これらのプロパティは、要求処理中に応答メッセージが発行されたときに動的メタオブジェクトも設定します。読み取り/書き込み属性は、要求処理中に作成されるメッセージ・ヘッダーで設定されます。イベント通知中は、読み取り/書き込み属性はメッセージ・ヘッダーから読み取られ、動的メタオブジェクトを設定します。

動的メタオブジェクトは、各属性がそれぞれ 1 つのメタデータ・プロパティと値を meta-object property name =meta-object property value の形式で表す構造になっています。

注: IBM WebSphere の標準のデータ・ハンドラーは、いずれも、cw_mo_ タグを認識すると、その後に指定されている動的メタオブジェクトを表す属性をビジネス・データ用の属性として処理しないように設計されています。アダプターで使用するカスタム・データ・ハンドラーを開発するときは、同様に設計する必要があります。

ポーリング時の動的子メタオブジェクトの取り込み

ポーリング時に検索されたメッセージに関してより多くの情報を持つコラボレーションを提供するために、コネクタは動的メタオブジェクトの特定の属性を（作成されるビジネス・オブジェクトにすでに定義されていれば）取り込みます。

表 20 に、動的子メタオブジェクトがポーリングのためにどのように構成されるかを示します。

表 20. ポーリング用の動的子メタオブジェクトの構造

プロパティ名	サンプル値
InputFormat	CUST_IN
InputQueue	MYInputQueue
OutputFormat	CxIgnore
OutputQueue	CxIgnore
ResponseTimeout	CxIgnore
TimeoutFatal	CxIgnore

表 20 に示すように、動的子メタオブジェクトに、追加属性 `Input_Format` および `InputQueue` を定義することができます。`Input_Format` には検索されたメッセージのフォーマットが取り込まれ、`InputQueue` 属性には特定のメッセージが検索されたキューの名前が含まれます。これらのプロパティが子メタオブジェクトで定義されていない場合は、これらのプロパティは取り込まれません。

シナリオの例:

- コネクタは、フォーマット `CUST_IN` のメッセージをキュー `MyInputQueue` から検索します。
- コネクタはこのメッセージを `Customer` ビジネス・オブジェクトに変換し、アプリケーション固有のテキストをチェックして、メタオブジェクトが定義されているかどうかを判別します。
- メタオブジェクトが定義されている場合、コネクタはそのメタオブジェクトのインスタンスを作成し、それによって `InputQueue` 属性と `InputFormat` 属性を取り込み、利用可能なコラボレーションにビジネス・オブジェクトを発行します。

JMS ヘッダーと動的子メタオブジェクトの属性

動的メタオブジェクトに属性を追加すると、メッセージ・トランスポートに関する詳細情報を取得したり、メッセージ・トランスポートの管理を強化したりできます。このセクションでは、これらの属性、および属性がイベント通知と要求処理に及ぼす影響について説明します。

JMS プロパティ: 動的メタオブジェクトの他の属性とは異なり、`JMSProperties` には単一カーディナリティーの子オブジェクトを定義しなければなりません。この子オブジェクトの各属性には、JMS メッセージ・ヘッダーの可変部分を読み取り元/書き込み先とするプロパティを、以下の規則に従って 1 つずつ定義する必要があります。

- 属性名にはセマンティック値を含めないこと。
- 属性の型は、JMS プロパティの型に関係なく、必ず `String` にすること。

3. 属性のアプリケーション固有情報には、その属性をマップする JMS メッセージ・プロパティの名前とフォーマットを定義した 2 組の名前と値のペアを含めること。名前はユーザーが定義できます。値の型は次のいずれかでなければなりません。

- Boolean
- String
- Int
- Float
- Double
- Long
- Short
- Byte

次の表に、JMSProperties オブジェクトの各属性に定義する必要があるアプリケーション固有情報のプロパティを示します。

表 21. JMS プロパティ属性のアプリケーション固有情報

属性	指定可能な値	ASI	コメント
Name	任意の有効な JMS プロパティ名 (有効とは、プロパティの型と ASI に定義した型が矛盾しないこと)	name=<JMS プロパティ名>;type=<JMS プロパティの型>	一部のベンダーは、特定のプロパティを機能拡張のために予約しています。一般に、ユーザーは、そのようなベンダー固有の機能を使用する場合を除いて、JMS で始まる名前のカスタム・プロパティを定義してはいけません。

表 21. JMS プロパティ属性のアプリケーション固有情報 (続き)

属性	指定可能な値	ASI	コメント
Type	String	type=<コメントを参照>	JMS プロパティの型です。JMS API には、JMS メッセージ内の値を設定するためのメソッドがいくつか用意されています (setIntProperty、setLongProperty、setStringProperty など)。ここで指定する JMS プロパティの型は、メッセージ内のそのプロパティの値を設定するときこれらのメソッドのうちのどれを使用すればよいかを示します。

次の例では、Customer オブジェクトに JMSProperties 子オブジェクトを定義して、メッセージ・ヘッダーのユーザー定義フィールドにアクセスできるようにしています。

```
Customer (ASI = cw_mo_conn=MetaData)
  -- Id
  -- FirstName
  -- LastName
  -- ContactInfo
  -- MetaData
    -- OutputFormat = CUST
    -- OutputDestination = QueueA
    -- JMSProperties
      -- RoutingCode = 123 (ASI= name=RoutingCode;type=Int)
      -- Dept = FD (ASI= name=RoutingDept;type=String)
```

もう 1 つの例として、図 4 に、動的メタオブジェクトに含まれる属性 JMSProperties と、JMS メッセージ・ヘッダーの 4 つのプロパティ (ID、GID、RESPONSE、および RESPONSE_PERSIST) の定義を示します。これらの属性のアプリケーション固有情報には、名前と型がそれぞれ定義されています。例えば、属性 ID は、型が String の JMS プロパティ ID にマップされます。

	Pos	Name	Type	Key	Reqd	Card	App Spec Info
1	1	JMSProperties	TeamCenter_JMS_Properties	<input type="checkbox"/>	<input type="checkbox"/>	1	
1.1	1.1	ID	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		name=ID,type=String
1.2	1.2	GUID	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		name=GUID,type=String
1.3	1.3	RESPONSE	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		name=RESPONSE,type=Boolean
1.4	1.4	RESP_PERSIST	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		name=RESPONSE_PERSIST,type=Boolean
1.5	1.5	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>		
2	2	OutputFormat	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

図4. 動的メタオブジェクトの JMS プロパティ属性

動的メタオブジェクトの作成のステップ

動的メタオブジェクトを実装するには、以下の手順を実行します。

1. **Business Object Designer** を起動します。詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。
2. 処理するとき動的メタオブジェクトを作用させる必要があるトップレベル・ビジネス・オブジェクトを開きます。
3. 開いたトップレベル・オブジェクトに動的メタオブジェクトを子オブジェクトとして追加し、名前と値のペア `cw_mo_conn=<MO attribute>` をそのトップレベル・オブジェクトの ASI に追加します。ここで、`<MO attribute>` は、そのトップレベル・オブジェクトに含まれる、動的メタオブジェクトを表現する属性の名前です。例えば、次のようにします。

```
Customer (ASI = cw_mo_conn=MetaData)
  -- Id
  -- FirstName
  -- LastName
  -- ContactInfo
  -- MetaData
    -- OutputFormat = CUST
    -- OutputDestination = QueueA
```

コネクタは、上の定義にあてはまる要求を受信すると、その要求 (Customer オブジェクト) をフォーマットが CUST のメッセージに変換し、キュー QueueA にそのメッセージを入れます。

4. トップレベル・ビジネス・オブジェクトを保管します。

注: 複数のビジネス・オブジェクトで同じ動的メタオブジェクトを使用することも、それぞれで異なる動的メタオブジェクトを使用することもできます。また、動的メタオブジェクトを一切使用しなくてもかまいません。

第 6 章 データ・ハンドラーの構成

- 『データ・ハンドラーの構成の概要』
- 『データ・ハンドラーの指定』
- 68 ページの『メッセージ・フローの変更』

この章では、データ・ハンドラーの構成方法について説明します。

データ・ハンドラーの構成の概要

データ・ハンドラーはコネクタの中心的なコンポーネントです。コネクタは、データ・ハンドラーを呼び出して、ビジネス・オブジェクトからメッセージへの変換とメッセージからビジネス・オブジェクトへの変換を行います。

コネクタ固有のデータ・ハンドラー・プロパティに含まれる情報は、これらの変換において重要な役割を果たします。この情報の構成は、製品ファイルをインストールした後、製品を始動する前に行います。

注: 構成したデータ・ハンドラーは、入力キューにマップすることもできます。詳細と手順については、60 ページの『データ・ハンドラーの入力キューへのマッピングの概要』を参照してください。

この章で説明する作業

この章で説明する作業は、以下のとおりです。

表 22. データ・ハンドラーの構成: 作業ロードマップ

作業	関連手順 (参照先)	詳細情報 (参照先)
データ・ハンドラーの指定	『データ・ハンドラーの指定』	
メッセージ・フローの変更	68 ページの『メッセージ・フローの変更』	ご使用のメッセージ・ブローカーの資料

データ・ハンドラーの指定

このトピックでは、データ・ハンドラーの指定の概要と手順を示します。

データ・ハンドラーの指定の概要

データ・ハンドラーを構成するには、以下のコネクタ固有の構成プロパティに値を指定します。

- DataHandlerClassName
- DataHandlerConfigMO
- DataHandlerMimeType

これらのプロパティの詳細については、『付録 B. Adapter for WBI Message Broker のコネクタ固有のプロパティ』を参照してください。

データ・ハンドラーの指定のステップ

始める前に: XML データ・ハンドラーを使用する場合は、データ・ハンドラー関連のコネクター構成プロパティのデフォルト値をそのまま使用するだけで済みます。該当する 3 つのプロパティ (`DataHandlerClassName`、`DataHandlerConfigMO`、および `DataHandlerMimeType`) は、いずれもデフォルトで XML データ・ハンドラー用に構成されています。

データ・ハンドラーを構成するには、以下の手順を実行します。

1. **Connector Configurator** で、「コネクター固有プロパティ」タブをクリックします。
2. `DataHandlerClassName` に、構成するデータ・ハンドラーに対応するプロパティ値を指定します。
3. `DataHandlerConfigMO` に、構成するデータ・ハンドラーに対応するプロパティ値を指定します。
4. `DataHandlerMimeType` に、構成するデータ・ハンドラーに対応するプロパティ値を指定します。
5. **Connector Configurator** でプロパティを適用します。

注: 静的メタオブジェクトまたは動的メタオブジェクトで上記のコネクター固有のプロパティと同名のプロパティに値を指定すると、上記のプロパティに指定した値よりもそれらの値が優先されます。詳細については、54 ページの『メタオブジェクト・プロパティ』を参照してください。

メッセージ・フローの変更

このトピックでは、メッセージ・フローの変更の概要と手順を示します。

メッセージ・フローの変更の概要

WebSphere Business Integration Adapter は、メッセージ・ブローカーを使用する場合、WebSphere MQ メッセージ・フローを使用してデータの処理とルーティングを行います。キューごとに 1 つずつ定義されるメッセージ・フローは、対応するキューに入れられたメッセージのすべてを処理します。Message Brokers Toolkit を使用すると、メッセージ・フローを構成して、処理する予定のメッセージのタイプごとに異なる処理ステップを指定することができます。

各着信メッセージが構成されたデータ・ハンドラーに対応するフォーマットに変換されるように、メッセージ・フローを変更する必要があります。この変換は、メッセージがコネクターの入力キューに送信される前に行われる必要があります。下記の手順は、メッセージ・フローを XML 対応の構成に変更する方法を示しています。メッセージ・フローを他のデータ・フォーマットに対応させるために変更する場合は、ステップ 2 の 3 番目の箇条書き項目の XML を、そのフォーマットを表す MIME タイプで置き換えてください。

メッセージ・フローを XML 対応の構成に変更するステップ

始める前に: XML データ・ハンドラーを構成します。

メッセージ・フローを変更して XML 対応にするには、以下の手順を実行します。

1. **Message Brokers Toolkit** を起動します。
2. メッセージ・フローの終わりに計算ノードを追加します。
3. ESQL テキスト領域に、以下のようにフィールドを入力します。

```
Set OutputRoot = InputRoot;
```

これにより、メッセージが出力のためにコピーされます。

```
Set OutputRoot.MQHRF2.Format = 'SO-CR';
```

これにより、コネクタがこのフォーマットをチェックして、メッセージを適切に変換することが保証されます。

```
SET OutputRoot.Properties.MessageFormat = 'XML';
```

これは、メッセージがデリバリー時に XML に変換される必要があることを WebSphere Business Integration Message Broker に示します。

4. 「適用」をクリックして計算ノードを使用可能にします。

図 5 に、着信メッセージをコネクタが認識できるフォーマットに変換するように構成された計算ノードのサンプル・ビューを示します。この計算ノードが使用可能になった後は、オリジナルのメッセージを表す XML 文書がコネクタの入力キューに送信されます。

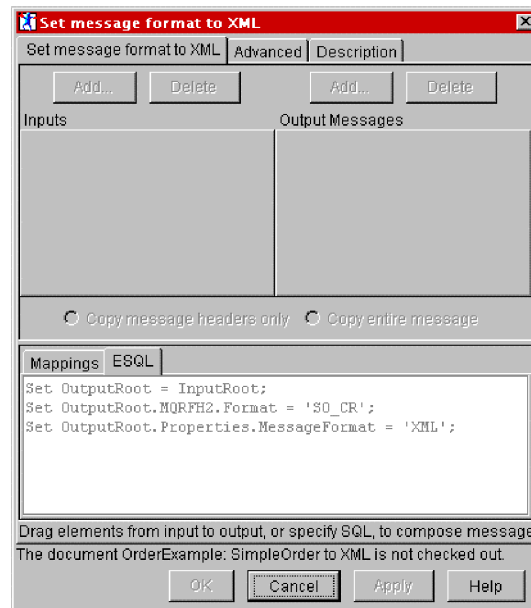


図 5. メッセージ・フォーマットを XML に設定

注: WebSphere Business Integration Message Broker Repository Manager 内でカスタム・フォーマットを定義している場合は、メッセージ・フォーマットを XML に設定するだけで既存のフォーマットを XML に変換できます。このフォーマットは、MQHRF2 とは異なります。OutputRoot.Properties.MessageFormat は MRM に関するものですが、OutputRoot.MQHRF2.Format はメッセージを受信するアプリケーションのメッセージ・フォーマットを指定するために使用されま

第 7 章 トラブルシューティング

- 『始動時の問題のトラブルシューティング』
- 72 ページの『イベント処理のトラブルシューティング』
- 72 ページの『サポートを受ける方法』

この章では、コネクタを始動または実行するときに発生する可能性がある問題について説明します。また、IBM のサポートを受ける方法についても説明します。

始動時の問題のトラブルシューティング

問題	考えられる処置/説明
初期化中にコネクタが予期しないエラーでシャットダウンし、次のメッセージが報告されました: Exception in thread "main" java.lang.NoClassDefFoundError: javax/jms/JMSException...	コネクタは、WebSphere MQ Java クライアント・ライブラリーからファイル jms.jar を見つけることができません。start_connector.bat 内の変数 MQSERIES_JAVA_LIB が WebSphere MQ Java クライアント・ライブラリー・フォルダーを指していることを確認します。
初期化中にコネクタが予期しないエラーでシャットダウンし、次のメッセージが報告されました: Exception in thread "main" java.lang.NoClassDefFoundError: com/ibm/mq/jms/MQConnectionFactory...	コネクタは、WebSphere MQ Java クライアント・ライブラリーからファイル com.ibm.mqjms.jar を見つけることができません。start_connector.bat 内の変数 MQSERIES_JAVA_LIB が WebSphere MQ Java クライアント・ライブラリー・フォルダーを指していることを確認します。
初期化中にコネクタが予期しないエラーでシャットダウンし、次のメッセージが報告されました: Exception in thread "main" java.lang.NoClassDefFoundError: javax/naming/Referenceable...	コネクタは、WebSphere MQ Java クライアント・ライブラリーからファイル jndi.jar を見つけることができません。start_connector.bat 内の変数 MQSERIES_JAVA_LIB が WebSphere MQ Java クライアント・ライブラリー・フォルダーを指していることを確認します。
初期化中にコネクタが予期しないエラーでシャットダウンし、次の例外が報告されました: java.lang.UnsatisfiedLinkError: no mqjbnd01 in shared library path	コネクタは、IBM MQSeries Java クライアント・ライブラリーから必要なランタイム・ライブラリー (mqjbnd01.dll [Windows] または libmqjbnd01.so [UNIX]) を見つけることができません。パスに WebSphere MQ Java クライアント・ライブラリーのフォルダーが含まれていることを確認します。
コネクタから次の例外が報告されました: MQJMS2005: failed to create MQQueueManager for ':'	次のプロパティーの値を明示的に設定します: HostName、Channel、および Port。

イベント処理のトラブルシューティング

問題

コネクターは、MQRFH2 ヘッダーを持つすべてのメッセージをデリバリーします。

コネクターは、コネクター・メタオブジェクト内でのメッセージ・フォーマットの定義にかかわらず、デリバリー時にすべてのメッセージ・フォーマットの 8 文字を超える部分を切り捨てます。

pollForEvents 時に、次の JMS 例外が報告された後でコネクターがシャットダウンしました: MQJMS1000:
Failed to create JMS message

pollForEvents 時に、次の JMS 例外が報告された後でコネクターがシャットダウンしました: MQJMS1052:
Unrecognised [sic] JMS Message class

考えられる処置/説明

MQMD WebSphere MQ ヘッダーを持つメッセージのみをデリバリーするには、出力キューの URI の名前に ?targetClient=1 を付加します。例えば、メッセージをキュー queue: //my.queue.manager/OUT に出力する場合は、URI を queue: //my.queue.manager/OUT?targetClient=1 に変更します。詳細は、13 ページの『第 2 章 コネクターのインストール』を参照してください。

これは WebSphere MQ MQMD メッセージ・ヘッダーの制限であり、コネクターの制限ではありません。

このエラーは MQ Java API に関するエラーであり、コネクター自身のエラーではありません。このエラーは、多くの場合、WebSphere Business Integration Message Broker 製品自体と同じマシン上でコネクターを実行しているときに発生します。この問題を解決するには、以下の手順を実行する必要があります。

1. ファイル Product_Dir¥Dependencies¥JRE_122_4.zip をフォルダー Product_Dir¥connectors¥WBIMB¥Dependencies¥jre_122_Re14 に unzip します。
2. Open Product_Dir¥connectors¥WBIMB¥start_MQSIV2.bat を開き、次の 2 つの行のコメントをはずします。oset PATH=%CONNDIR%¥Dependencies¥jre_122_Re14¥bin...oset JAVA=%CONNDIR%¥Dependencies¥jre_122_Re14¥lib¥rt.jar
3. コネクターを再始動します。

このエラーは、JMS 準拠のアプリケーションから発生したメッセージを WebSphere Business Integration Message Broker が変更したことによって発生することがあります。このエラーを訂正するには、WebSphere Business Integration Message Broker 計算ノードに次の SQL ステートメントを追加することにより、問題が発生したメッセージから残りの JMS 情報を除去します: SET OutputRoot.MQRFH2.jms = null;

サポートを受ける方法

始める前に: 本書の発行後に公開されたテクニカル・サポートの技術情報や速報に、本書の対象製品に関する重要な情報が記載されている場合があります。これらの技術情報や速報は、WebSphere Business Integration のサポート Web サイトで参照できます。

WebSphere Business Integration のサポート Web サイトを利用するには、以下の手順を実行します。

1. <http://www.ibm.com/software/integration/websphere/support/> にアクセスします。
2. 適切なコンポーネント領域を選択し、「Technotes」セクションと「Flashes」セクションをブラウズするか検索します。

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 Adapter のコネクター・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、以下の統合ブローカーで実行されるコネクター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)

コネクターによっては、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きには、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のある標準プロパティのリストが表示されます。

コネクター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

以下の標準プロパティは、本リリースで追加されました。

新規プロパティ

- XMLNamespaceFormat

削除されたプロパティ

- RestartCount

標準コネクター・プロパティの構成

アダプター・コネクターには 2 種類の構成プロパティがあります。

- 標準構成プロパティ
- コネクター固有の構成プロパティ

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。コネクター固有の構成プロパティの詳細は、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 の使用方法の詳細は、Connector Configurator に関する付録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。コネクタが UNIX システム上で動作している場合は、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

プロパティのフィールド長のデフォルトは 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は、他の項目の値をオーバーライドします)。

1. デフォルト
2. リポジトリ (統合ブローカーが WebSphere InterChange Server の場合のみ)
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、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 Broker で稼働している場合など、コネクタがスタンドアロン・モードで (System Manager から独立して) 稼働している場合は、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。

• 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。

• 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。

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 75 ページの表 23 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 23 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。

注: 表 23 の「注」列にある「Repository Directory は REMOTE」という句は、ブローカーが InterChange Server であることを示します。ブローカーが WMQI または WAS の場合には、リポジトリ・ディレクトリは LOCAL に設定されます。

表 23. 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> (ブローカーは ICS)
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		コンポーネント再始動	
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> (ブローカーは ICS)
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

表 23. 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
ControllerStoreAndForwardMode	true または false	true	動的	Repository Directory は <REMOTE> (ブローカーは ICS)
ControllerTraceLevel	0 から 5	0	動的	Repository Directory は <REMOTE> (ブローカーは ICS)
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、値は JMS のみ
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> でなければならぬ
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 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

表 23. 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
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> (ブローカーは ICS)
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> (ブローカーは ICS)
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> (ブローカーは ICS)
ListenerConcurrency	1 から 100	1	コンポーネント再始動	Delivery Transport は MQ でなければならぬ
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> でなければならぬ (ブローカーは ICS)
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
MessageFileName	パスまたはファイル名	CONNECTORNAMEConnector.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

表 23. 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
OADAutoRestartAgent	true または false	false	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
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	
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

表 23. 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
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	
RHF2MessageDomain	mrm、xml	mrm	コンポーネント再始動	Delivery Transport が JMS であり、かつ WireFormat が CwXML である場合のみ
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
WsifSynchronousRequestTimeout	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際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際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 (オブジェクト・リクエスト・ブローカー) 接続の数を制御します。

このプロパティのデフォルト値は 1 に設定されます。必要に応じてこの値を変更できます。

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、ドロップダウン・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。ドロップダウン・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。

デフォルト値はありません。

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 = /SOURCEQUEUE

また、MimeType、DHClass (データ・ハンドラー・クラス)、および DataHandlerConfigMOName (オプションのメタオブジェクト名) プロパティーを設定したデータ・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ー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。

これらのプロパティーはアダプター固有ですが、例の値は次のようになります。

- MimeType = text/xml
- DHClass = com.crossworlds.DataHandlers.text.xml
- DataHandlerConfigMOName = M0_DataHandler_Default

「データ・ハンドラー」タブのこれら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、コネクターはその 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。

このプロパティーは、DeliveryTransport プロパティーが値 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ー・コントローラーが検出した場合に、コネクター・コントローラーが実行する動作を設定します。

このプロパティーを 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、コネクター・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、コネクター・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、コネクター・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、コネクター・コントローラーはその要求を失敗させます。

このプロパティーを false に設定した場合、コネクター・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。

コネクター・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。

- RepositoryDirectory がリモートの場合は、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 (ASYNC) 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、メモリー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ーが使用されるため、(サーバー側の) コネクタ・コントローラーと (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、製品ディレクトリー配下の `%bin` ディレクトリーにあります。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・メモリーの使用量を最大 768 MB (3 セグメント * 256 MB) に制限します。プロセス・メモリー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>` の場合のみ適用できます。

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>` の場合のみ適用できます。

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>` の場合のみ適用できます。

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。ローカル・メッセージ・ブローカーに接続する場合は、デフォルト値を使用します。

リモート・メッセージ・ブローカーに接続すると、このプロパティは次の (必須) 値をとります。

`QueueMgrName:<Channel>:<HostName>:<PortNumber>`

各変数の意味は以下のとおりです。

`QueueMgrName`: キュー・マネージャー名です。

`Channel`: クライアントが使用するチャンネルです。

`HostName`: キュー・マネージャーの配置先のマシン名です。

`PortNumber`: キュー・マネージャーが `listen` に使用するポートの番号です。

例えば、次のようにします。

```
jms.MessageBrokerName = WBIMB.Queue.Manager:CHANNEL1:RemoteMachine:1456
```

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 トランスポートを使用するコネクタにのみ適用されません。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。

<i>ll</i>	2 文字の言語コード (普通は小文字)
<i>TT</i>	2 文字の国または地域コード (普通は大文字)
<i>codeset</i>	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、ドロップダウン・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。ドロップダウン・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>` の場合のみ適用できます。

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、製品ディレクトリーの `¥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

これは、前回のポーリングの終了から次のポーリングの開始までの間の間隔です。PollFrequency は、あるポーリング・アクションの終了から次のポーリング・アク

ションの開始までの時間をミリ秒単位で指定します。これはポーリング・アクション間の間隔ではありません。この論理を次に説明します。

- ポーリングし、PollQuantity の値により指定される数のオブジェクトを取得します。
- これらのオブジェクトを処理します。一部のアダプターでは、これは個別のスレッドで部分的に実行されます。これにより、次のポーリング・アクションまで処理が非同期に実行されます。
- PollFrequency で指定された間隔にわたって遅延します。
- このサイクルを繰り返します。

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 (整数)。
- ワード key。コネクターは、コネクターのコマンド・プロンプト・ウィンドウで文字 p が入力されたときに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。コネクター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: 一部のコネクターでは、このプロパティの使用が制限されています。このようなコネクターが存在する場合には、アダプターのインストールと構成に関する章で制約事項が説明されています。

PollQuantity

コネクター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ー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、このコネクター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。

電子メール・メッセージもイベントと見なされます。コネクターは、電子メールに関するポーリングを受けたときには次のように動作します。

コネクターは、1 回目のポーリングを受けると、メッセージの本文を選出します。これは、本文が添付とも見なされるからです。本文の MIME タイプにはデータ・ハンドラーが指定されていないので、コネクターは本文を無視します。

コネクターは PO の最初の添付を処理します。この添付の MIME タイプには対応する DH があるので、コネクターはビジネス・オブジェクトを Visual Test Connector に送信します。

2 回目のポーリングを受けると、コネクターは PO の 2 番目の添付を処理します。この添付の MIME タイプには対応する DH があるので、コネクターはビジネス・オブジェクトを Visual Test Connector に送信します。

これが受け入れられると、POの 3 番目の添付が届きます。

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ー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

コネクターが 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、コネクターが 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ー・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

コネクターによるコネクター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ーに対して使用する場合、コネクター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

コネクターによるコネクター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ーに対して使用する場合、コネクター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。

RHF2MessageDomain

WebSphere Message Brokers および WAS でのみ使用されます。

このプロパティにより、JMS ヘッダーのドメイン名フィールドの値を構成できます。JMS トランスポートを介してデータを WMQI に送信するときに、アダプター・フレームワークにより JMS ヘッダー情報、ドメイン名、および固定値 `mrm` が書き込まれます。この構成可能なドメイン名により、ユーザーは WMQI ブローカーによるメッセージ・データの処理方法を追跡できます。

サンプル・ヘッダーを以下に示します。

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

デフォルト値は `mrm` ですが、このプロパティには `xml` も設定できます。このプロパティは、`DeliveryTransport` が JMS に設定されており、かつ `WireFormat` が `CwXML` に設定されている場合にのみ表示されます。

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、81 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 を指定する関連 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。

WsifSynchronousRequestTimeout

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. Adapter for WBI Message Broker のコネクタ固有のプロパティ

この付録では、Adapter for WebSphere Business Integration Message Broker のコネクタ固有のプロパティについて説明します。Connector Configurator を使用してアダプターを構成するときには、これらのプロパティの説明を参照してください。

コネクタ固有のプロパティの概要

コネクタ固有の構成プロパティは、コネクタ・エージェントが実行時に必要とする情報を提供します。また、コネクタ固有の構成プロパティを使用すると、コネクタ・エージェントのコード変更や再ビルドを行わなくても、エージェント内の静的情報またはロジックを変更できます。

コネクタ固有のプロパティ

表 24 に、このコネクタに固有な構成プロパティを示します。プロパティの説明については、以下の各セクションを参照してください。

表 24. コネクター固有の構成プロパティ

名前	指定可能な値	デフォルト値	必須
ApplicationPassword	ログイン・パスワード		いいえ
ApplicationUserName	ログイン・ユーザー ID		いいえ
ArchiveQueue	正常に処理されたメッセージのコピーが送信されるキュー	MQWFCNN.ARCHIVE	いいえ
CCSID	キュー・マネージャー接続用文字セット	ヌル	はい
Channel	MQ サーバー・コネクター・チャンネル		はい
ConfigurationMetaObject	構成メタオブジェクトの名前		はい
DataHandlerClassName	データ・ハンドラー・クラス名	com.crossworlds.DataHandlers. text.xml	はい
DataHandlerConfigMO	データ・ハンドラー・メタオブジェクト	MO_DataHandler_Default	はい
DataHandlerMimeType	ファイルの MIME タイプ	text/xml	いいえ
DefaultVerb	コネクターがサポートしている任意の動詞。	Create	
EnableMessageProducerCache	true または false	true	いいえ
ErrorQueue	未処理のメッセージのキュー	MQWFCNN.ERROR	いいえ
HostName	WebSphere MQ サーバー		いいえ
InDoubtEvents	FailOnStartup Reprocess IgnoreLogError	Reprocess	いいえ
InputQueue	ポーリング・キュー	MQWFCNN.IN	はい
InProgressQueue	進行中イベント・キュー	MQWFCNN.IN_PROGRESS	いいえ
PollQuantity	InputQueue プロパティ内で指定された各キューから検索されるメッセージの数	1	いいえ
Port	WebSphere MQ リスナー用に設定されたポート		いいえ
ReplyToQueue	コネクターからの要求発行時に応答メッセージがデリバリーされるキュー	MQWFCNN.REPLYTO	いいえ
SessionPoolSizeForRequests	要求処理中に使用されるセッションをキャッシュする最大プール・サイズ	10	いいえ
UnsubscribedQueue	アンサブスクライブされたメッセージが送信されるキュー	MQWFCNN.UNSUBSCRIBED	いいえ
UseDefaults	true または false	false	

ApplicationPassword

WebSphere Business Integration Message Broker にログインするために、UserID とともに使用されるパスワード。

デフォルト = 設定値なし

ApplicationPassword の値がブランクのままか、または除去された場合、コネクターは WebSphere Business Integration Message Broker によって提供されるデフォルトのパスワードを使用します。

ApplicationUserName

WebSphere Business Integration Message Broker にログインするために、Password とともに使用されるユーザー ID。

デフォルト = 設定値なし

ApplicationUserName の値がブランクのままか、または除去された場合、コネクタは WebSphere Business Integration Message Broker によって提供されるデフォルトのユーザー ID を使用します。

ArchiveQueue

正常に処理されたメッセージのコピーが送信されるキューです。

デフォルト = MQWFCONN.ARCHIVE

CCSID

キュー・マネージャー接続用文字セット。このプロパティの値は、キュー URI 内の CCSID プロパティの値と一致している必要があります。39 ページの『キューの Uniform Resource Identifier (URI) の設定』を参照してください。

デフォルト = ヌル

Channel

コネクタが WebSphere Business Integration Message Broker と通信するときに使用する MQ サーバー・コネクタ・チャンネルです。

デフォルト = 設定値なし

Channel の値がブランクのままか、または除去された場合、コネクタは WebSphere Business Integration Message Broker によって提供されるデフォルトのサーバー・チャンネルを使用します。

ConfigurationMetaObject

コネクタの構成情報を含むメタオブジェクトの名前です。

デフォルト = 設定値なし

DataHandlerClassName

ビジネス・オブジェクトとの間でのメッセージ変換に使用するデータ・ハンドラー・クラスです。

デフォルト = com.crossworlds.DataHandlers.text.xml

DataHandlerConfigMO

構成情報を提供するためにデータ・ハンドラーに渡されるメタオブジェクトです。

デフォルト = MO_DataHandler_Default

DataHandlerMimeType

使用すると、特定の MIME タイプに基づいたデータ・ハンドラーを要求できます。

デフォルト = text/xml

DefaultVerb

ポーリング時にデータ・ハンドラーによって動詞が設定されなかった場合、着信ビジネス・オブジェクトの内部に設定する動詞を指定します。

デフォルト = Create

EnableMessageProducerCache

要求メッセージを送信するために、アダプターがメッセージ・プロデューサーのキャッシュを有効にすることを指定する `boolean` プロパティ。

デフォルト = true

ErrorQueue

処理されなかったメッセージが送信されるキューです。

デフォルト = `queue://CrossWorlds.QueueManager/ERROR`

HostName

WebSphere Business Integration Message Broker をホスティングしているサーバーの名前です。

デフォルト = 設定値なし

InDoubtEvents

コネクターの予期しないシャットダウンのために、処理が完了していない進行中イベントの処理方法を指定します。初期化中に進行中キューにイベントが見つかった場合に実行するアクションを、以下の 4 つから選択してください。

- `FailOnStartup`: エラー・ログを記録して即時にシャットダウンします。
- `Reprocess`: 残りのイベントを先に処理してから、入力キューのメッセージを処理します。
- `Ignore`: 進行中キューのすべてのメッセージを無視します。
- `LogError`: エラー・ログを記録しますが、シャットダウンはしません。

デフォルト = `Reprocess`

InputQueue

コネクターが新規メッセージをポーリングするメッセージ・キューです。コネクターは、セミコロンによって区切られた複数のキュー名を受け入れます。例えば、`MyQueueA`、`MyQueueB`、および `MyQueueC` の 3 つのキューをポーリングするには、コネクター構成プロパティ `InputQueue` の値を `MyQueueA;MyQueueB;MyQueueC` にします。

コネクターはラウンドロビン方式でキューをポーリングし、各キューから `pollQuantity` で指定された値を最大数とするメッセージを検索します。例えば、`pollQuantity` の値が 2 で、`MyQueueA` 内に 2 つ、`MyQueueB` 内に 1 つ、`MyQueueC` 内に 5 つのメッセージがそれぞれ格納されている場合、コネクターは次のようにメッセージを検索します。

pollQuantity が 2 に設定されているため、コネクタは、pollForEvents への 1 回の呼び出しごとに各キューからそれぞれ最大 2 つのメッセージを検索します。最初のサイクル (2 サイクルのうち 1 サイクル目) では、コネクタは MyQueueA、MyQueueB、および MyQueueC の各キューからそれぞれ 1 番目のメッセージを検索します。これによって、ポーリングの第 1 ラウンドが完了します。pollQuantity が 1 に設定されている場合、コネクタはこの時点で停止します。この例では pollQuantity が 2 に設定されているため、コネクタは第 2 ラウンドのポーリングを開始し、MyQueueA と MyQueueC の各キューからそれぞれ 1 つずつのメッセージを検索します。このとき、MyQueueB は空になっているためスキップされます。すべてのキューを 2 回ずつポーリングすると、メソッド pollForEvents への呼び出しは完了します。このメッセージ検索の順序を以下に示します。

1. MyQueueA から 1 つのメッセージ
2. MyQueueB から 1 つのメッセージ
3. MyQueueC から 1 つのメッセージ
4. MyQueueA から 1 つのメッセージ
5. MyQueueB は空なのでスキップ
6. MyQueueC から 1 つのメッセージ

デフォルト = queue://CrossWorlds.QueueManager/IN

InProgressQueue

処理中にメッセージが保持されるメッセージ・キューです。コネクタがこのキューなしで動作するように構成するには、System Manager を使用してコネクタ固有のプロパティからデフォルトの InProgressQueue 名を除去します。これを実行すると、始動時に、イベントの保留中にコネクタをシャットダウンするとイベント・デリバリーが正常に実行されない可能性があるという警告のプロンプトが出力されます。

デフォルト = queue://CrossWorlds.QueueManager/IN_PROGRESS

PollQuantity

pollForEvents スキャン中に InputQueue プロパティ内で指定された各キューから検索されるメッセージの数

デフォルト = 1

Port

WebSphere Business Integration Message Broker リスナー用に設定されたポートです。

デフォルト = 設定値なし

ReplyToQueue

コネクタからの要求発行時に応答メッセージがデリバリーされるキューです。

デフォルト = queue://CrossWorlds.QueueManager/REPLYTO

SessionPoolSizeForRequests

要求処理中に使用されるセッションをキャッシュする最大プール・サイズ。

デフォルト = 10

UnsubscribedQueue

サブスクライブされていないメッセージが送信されるキューです。

デフォルト = queue://CrossWorlds.QueueManager/UNSUBSCRIBED

注: *WebSphere Business Integration Message Broker によって提供される値は誤っていたり不明である可能性があるため、常にチェックする必要があります。値が誤っていたり不明な場合は、値を暗黙的に指定してください。

UseDefaults

Create 操作において、UseDefaults に true が設定されている場合、コネクタは、各 isRequired ビジネス・オブジェクト属性に対して、デフォルト値または有効値が指定されているかどうかをチェックします。デフォルト値または有効値が指定されている場合は、Create 操作は成功します。このパラメーターに false が設定されている場合は、コネクタは、有効値のみをチェックします。有効値が指定されていない場合は、Create 操作は失敗します。デフォルト値は false です。

付録 C. チュートリアル

- 『チュートリアルの概要』
- 100 ページの『始める前に』
- 101 ページの『環境のセットアップ』
- 104 ページの『シナリオの実行』

このチュートリアルは、以下のことを示すことを目的として記述されています。

- Adapter for WebSphere Business Integration Message Broker を WebSphere Business Integration Message Broker のメッセージ・フローと統合するにはどうすればよいか。
- アダプターはどのようにしてレガシー・アプリケーションとの間でメッセージを送受信するのか。

このチュートリアルのシナリオは、アダプターの機能の基本を示すことを目的としています。

表記規則のガイドについては、本書のまえがきを参照してください。

チュートリアルの概要

このチュートリアルでは、メッセージ・ブローカー経由でアダプターに XML メッセージを送信するレガシー・アプリケーション (ユーティリティによりシミュレートしたもの) を使用します。メッセージにはさまざまなカタログ項目の変更が含まれます。このレガシー・メッセージは、WebSphere Business Integration Message Broker に関連付けられているデータウェアハウス・フローとメッセージ変換によって処理されるのではなく、単純なカラー・チェック・フローによって処理されます。処理されたメッセージは、アダプターに送信されます。

アダプターでは、カラー処理されたメッセージを WBI Message Broker から受け取ります。アダプターは XML データ・ハンドラーを使用してそのメッセージをビジネス・オブジェクト `Sample_WBIMB_LegacyItem_XMLDoc` に変換し、さらに統合ブローカーにこのオブジェクトを渡します。このチュートリアルでは、Visual Test Connector を使用してポート・コネクターをシミュレートし、新規作成されたビジネス・オブジェクトを取得してその内容を確認します。

また、逆に Visual Test Connector からビジネス・オブジェクトを発行します。発行されたオブジェクトは、Adapter for WBI Message Broker でレガシー・メッセージに変換されます。このメッセージはアダプターから WebSphere Business Integration Message Broker アプリケーションにデリバリーされ、ここで処理された後 (メッセージ・フォーマットが `LI_UP` に更新されます)、レガシー・アプリケーション (シミュレートされたもの) に転送されます。このシナリオについては、WMQI に対応するように構成されたアダプターを使用して説明します。

このシナリオを実行するために必要な条件は、アダプターとアダプター・フレームワークに加えて以下のものがインストールされていることだけです。

- メッセージ・フローのサンプル、ドメイン構成マネージャー・ファイルのサンプル、および配置可能な bar ファイルが含まれる WebSphere Business Integration Message Broker プロジェクト (メッセージ・フロー・プロジェクトおよびサーバー・プロジェクト)。
- ポート・コネクタのリポジトリと Visual Test Connector (ADK のインストールに組み込まれています)。ポート・コネクタは、特定のコードを基礎としないアダプター定義で構成されており、そのためシミュレーション・シナリオに適しています。Visual Test Connector は、インターフェースをテストするためのツールです。

この付録で説明する作業

この付録では、以下の作業について説明します。

表 25. アダプターのデモンストレーション: 作業ロードマップ

作業	関連手順 (参照先)
インストール内容のチェックリストの確認	『始める前に』
環境のセットアップ	101 ページの『環境のセットアップ』
シナリオの実行	104 ページの『シナリオの実行』

始める前に

このチュートリアルを開始する前に、以下のことを確認してください。

- IBM WebSphere 製品がインストールされており、かつこの製品の操作に慣れていること。
- WebSphere MQ 5.1 以降がインストールされていること。
- WebSphere Business Integration Message Broker 5.0 以降がインストールされていること。
- WebSphere MQ クライアント・ライブラリー (Java 用) がインストールされていること。
- Adapter for WebSphere Business Integration Message Broker がインストールされていること (構成方法については、このチュートリアルで説明します)。
- WebSphere MQ のアダプター・キュー・マネージャーに `crossworlds.queue.manager` という名前 (インストール時に設定されるデフォルト値) が付けられていること。これ以外の名前が付けられている場合は、本書に記載されている「`crossworlds.queue.manager`」を、そのキュー・マネージャー名に読み替えてください。また、以下の手順を実行する必要があります。
 1. Message Brokers Toolkit を開き、メッセージ・フローのサンプルに含まれるすべての MQOutput ノードの QueueManagerName 属性の値を、`crossworlds.queue.manager` に変更します。
 2. サーバー・プロジェクト (アダプターとともにインストールされたサンプルに含まれています) 内の `LocalDomain.configmgr` ファイルを開き、キュー・マネージャー名 (ブローカーが対応するキュー・マネージャーの名前) を `crossworlds.queue.manager` に変更します。

3. Business Object Designer を開き、メタオブジェクトのサンプルである Sample_WBIMB_MO_Config に指定されている出力キュー・マネージャーを crossworlds.queue.manager に変更します。
- WebSphere MQ Explorer を含めて、WebSphere MQ の (基本インストールではなく) フルインストールが行われていること。WebSphere MQ Explorer は必須ではありません。ただし、このツールを使用すると、キューの設定とメッセージの検証を容易に実行できます。
 - WebSphere MQ クライアント・ライブラリー (Java 用) がインストールされていること。

環境のセットアップ

このトピックでは、環境のセットアップの概要と手順を示します。

環境のセットアップの概要

このセクションでは、このチュートリアルを実行するために必要な環境を準備する方法について説明します。なお、この後の説明の *sample_folder* という変数は、インストール済みのファイル構造に含まれている samples/Legacy Item フォルダを表しています。詳細については、16 ページの『インストールの検証』を参照してください。ビジネス・オブジェクト・リポジトリは、*sample_folder* 内に .xsd ファイルとして用意されています。

このチュートリアルでは、Adapter for WebSphere Business Integration Message Broker と Visual Test Connector の間での単純なビジネス・オブジェクト交換の例を示します。この交換は、WebSphere Business Integration Message Broker 環境内で行われます。

環境のセットアップのステップ

このチュートリアルで使用する環境をセットアップするには、以下の構成作業を行う必要があります。

1. **キューの定義:** このチュートリアルでは、キュー・マネージャーに 8 つのキューが定義されていることが必要です。必要なキューを作成するには、コマンド行から「RUNMQSC」と入力し、以下のコマンドを実行します。

- DEFINE QL('Samples/WBIMB/Item/LegacyApp')
- DEFINE QL('Samples/WBIMB/Item/WBIMBConnector')
- DEFINE QL('Samples/LegacyApp/Item/WBIMB')
- DEFINE QL('Samples/WBIMBConnector/Item/WBIMB')
- DEFINE QL('Samples/WBIMB/FAIL')
- DEFINE QL('Samples/WBIMBConnector/UNSUBSCRIBED')
- DEFINE QL('Samples/WBIMBConnector/ERROR')
- DEFINE QL('Samples/WBIMBConnector/ARCHIVE')

次に、WBI Message Broker を構成するために、Adapter for WBI Message Broker とポート・コネクタが必要とするキューを以下のように定義します。

- DEFINE QL('WBIMBConnector/ADMININQUEUE')
- DEFINE QL('WBIMBConnector/ADMINOUTQUEUE')

- DEFINE QL('WBIMBConnector/DELIVERYQUEUE')
- DEFINE QL('WBIMBConnector/FAULTQUEUE')
- DEFINE QL('WBIMBConnector/REQUESTQUEUE')
- DEFINE QL('WBIMBConnector/RESPONSEQUEUE')
- DEFINE QL('WBIMBConnector/SYNCHRONOUSREQUESTQUEUE')
- DEFINE QL('WBIMBConnectorSYNCHRONOUSRESPONSEQUEUE')
- DEFINE QL('WBIMBConnectorMONITORQUEUE')
- DEFINE QL('PortConnector/ADMININQUEUE')
- DEFINE QL('PortConnector/ADMINOUTQUEUE')
- DEFINE QL('PortConnector/DELIVERYQUEUE')
- DEFINE QL('PortConnector/FAULTQUEUE')
- DEFINE QL('PortConnector/REQUESTQUEUE')
- DEFINE QL('PortConnector/RESPONSEQUEUE')
- DEFINE QL('PortConnector/SYNCHRONOUSREQUESTQUEUE')
- DEFINE QL('PortConnector/SYNCHRONOUSRESPONSEQUEUE')

2. **アダプターの構成: Connector Configurator** で「ファイル」->「開く」->「ファイルから」を選択し、*sample_folder* にある *WBIMBConnector.cfg* をロードします。アダプター構成プロパティの値が以下に示す値と一致していることを確認します。または、一致するように変更します。Connector Configurator の使用方法の詳細については、67 ページの『第 6 章 データ・ハンドラーの構成』を参照してください。コネクタ固有のプロパティの詳細については、93 ページの『付録 B. Adapter for WBI Message Broker のコネクタ固有のプロパティ』を参照してください。
- 標準プロパティに関しては、以下のプロパティを設定する必要があります。

- **BrokerType:** WMQI に設定します。
 - **RepositoryDirectory:** *sample_folder* ディレクトリーに設定します。
 - **DuplicateEventElimination:** true に設定します。
 - **MonitorQueue:** WBIMBConnector/MONITORQUEUE に設定します。
- 以下のコネクタ固有プロパティを設定します。
- **ConfigurationMetaObject:** *Sample_WBIMB_MO_Config* に設定します。
 - **DataHandlerConfigMO:** *Sample_WBIMB_MO_DataHandler* に設定します。
 - **DataHandlerMimeType:** text/xml に設定します。
 - **ErrorQueue:**
queue://crossworlds.queue.manager/Samples/WBIMBConnector/ERROR に設定します。
 - **InputQueue:**
queue://crossworlds.queue.manager/Samples/WBIMB/Item/WBIMBConnector に設定します。
 - **UnsubscribedQueue:**
queue://crossworlds.queue.manager/Samples/WBIMBConnector/UNSUBSCRIBED に設定します。

- ArchiveQueue:
queue://crossworlds.queue.manager/Samples/WBIMBConnector/ARCHIVE に設定します。
3. **ポート・コネクタの構成: Connector Configurator** を使用して、以下の標準プロパティを設定します。
 - BrokerType: WMQI に設定します。
 - RepositoryDirectory: *sample_folder* ディレクトリーに設定します。
 - RequestQueue: WBIMBConnector/DELIVERYQUEUE (Adapter for WBI Message Broker の DeliveryQueue プロパティの値) に設定します。
 - DeliveryQueue: WBIMBConnector/REQUESTQUEUE (Adapter for WBI Message Broker の RequestQueue プロパティの値) に設定します。
 4. **サポートするビジネス・オブジェクトの構成:** ビジネス・オブジェクトを使用するには、まずアダプターでそのビジネス・オブジェクトがサポートされていることが必要です。**Connector Configurator** で、Adapter for WBI Message Broker の「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、表 26 のビジネス・オブジェクトを追加します。さらに、「メッセージ・セット ID」を、サポートするビジネス・オブジェクトのそれぞれに固有の値に設定します。

表 26. JMS アダプターでサポートするサンプル・ビジネス・オブジェクト

ビジネス・オブジェクト名	メッセージ ID
Sample_WBIMB_MO_Config	1
Sample_WBIMB_MO_DataHandler	2
Sample_WBIMB_LegacyItem	3
Sample_WBIMB_LegacyItem_XMLDoc	4

Connector Configurator で、*sample_folder* にあるポート・コネクタの定義 PortConnector.cfg を開き、表 27 のサポートするビジネス・オブジェクトとメッセージ ID を追加します。

表 27. ポート・コネクタでサポートするサンプル・ビジネス・オブジェクト

ビジネス・オブジェクト名	メッセージ ID
Sample_WBIMB_LegacyItem	1
Sample_WBIMB_LegacyItem_XMLDoc	2

5. **メッセージ・フロー・プロジェクトの新規作成**
 - a. **Message Brokers Toolkit** を開き、メッセージ・フロー・プロジェクトを新規作成します。作成したメッセージ・フロー・プロジェクトに、*sample_folder/MSG_FLOW_PROJECT* ディレクトリーのメッセージ・フローと ESQLE ファイルをすべてインポートします。
 - b. **サーバー・プロジェクト**を新規作成し、作成したサーバー・プロジェクトに *sample_folder/Sample_WBIMB_Project* のファイルをすべてインポートします。
 - c. 「ブローカー管理 (**Broker Administration**)」パースペクティブの「ドメイン (**Domains**)」ビューの下に表示されたドメインに接続します。bar ファイル *Sample_WBIMB_bar.bar* を、「ブローカー管理 (**Broker**

Administration) パースペクティブの「ブローカー管理ナビゲーター (Broker Administration Navigator)」パネルから、ブローカーのデフォルトの実行グループに配置します。

6. コネクタ始動スクリプトの構成

Windows:

- a. Adapter for WBI Message Broker のショートカットのプロパティを開きます。
- b. リンク先の最後の引き数として、`-c` に `<WBIMBConnector.cfg` ファイルの絶対パスとファイル名> を続けたものを追加します。例えば、次のようにします。

```
-cProduct_Dir%connectors%WBIMB%LegacyItem%  
WBIMBConnector.cfg
```

UNIX:

- a. ファイル

`Product_Dir/bin/connector_manager_WebSphereBusinessIntegrationMessageBroker` を開きます。AGENTCONFIG_FILE プロパティの値として、`-c` に `<WBIMBConnector.cfg` ファイルの絶対パスとファイル名> を続けたものを設定します。例えば、次のようにします。

```
AGENTCONFIG_FILE=-cProduct_Dir/connectors/WebSphereBusinessIntegrationMessageBroker  
/samples/LegacyItem/WBIMBConnector.cfg
```

シナリオの実行

このトピックでは、シナリオの実行の概要と手順を示します。

シナリオの実行の概要

このチュートリアルには、要求処理シナリオとイベント処理シナリオがあります。これらのシナリオを実行するには、101 ページの『環境のセットアップ』に説明されているとおりにサンプルを構成する必要があります。

シナリオの実行のステップ

シナリオを実行するには、以下の手順を実行します。

1. **Adapter for WebSphere Business Integration Message Broker** を始動します (このアダプターがまだ実行されていない場合)。詳細については、44 ページの『コネクタの始動』を参照してください。
2. **Visual Test Connector** を始動します (このツールがまだ実行されていない場合)。詳細については、「*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。
3. **WebSphere Business Integration Message Broker アプリケーション・ブローカー** を始動します (このブローカーがまだ実行されていない場合)。詳細については、「*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。
4. ポート・コネクタをシミュレートします。**Visual Test Connector** を使用して、以下の手順でポート・コネクタのプロファイルを定義してください。

- a. 「**Visual Test Connector**」メニューから「ファイル」->「プロファイルを作成/選択」を選択し、次に「コネクタ・プロファイル」メニューから「ファイル」->「新規プロファイル」を選択します。
 - b. *sample_folder*にあるポート・コネクタの構成ファイル *PortConnector.cfg* を選択し、コネクタ名とブローカー・タイプを構成してから「**OK**」をクリックします。
 - c. 作成したプロファイルを選択し、「**OK**」をクリックします。
5. 要求処理シナリオをテストします。**Visual Test Connector** を使用して、以下の手順でビジネス・オブジェクトのインスタンスを新規作成し、送信してください。
- a. ビジネス・オブジェクト *Sample_WBIMB_LegacyItem_XMLDoc* のインスタンスを新規作成します。これを行うには、「**ビジネス・オブジェクト・タイプ**」ドロップダウン・ボックスでこのビジネス・オブジェクトを選択し、次に「**BO インスタンス**」で「**作成**」を選択します。このビジネス・オブジェクトは、XML データ・ハンドラーが必要とする XML ラッパーでしかありません。実際のデータを格納するには、このテスト用オブジェクトの内部に、子オブジェクト **LegacyItem** のインスタンスを新規作成する必要があります。デフォルト値は、必要に応じて変更してください。
 - b. 「**ビジネス・オブジェクトを送信**」をクリックしてメッセージを送信します。
6. **メッセージのデリバリーを確認**します。WebSphere MQ Explorer または同様のアプリケーションを使用してキュー `queue://crossworlds.queue.manager/Samples/WBIMB/Item/LegacyApp` を開き、フォーマットが **LI_UP** の新しいレガシー項目メッセージがアダプターから届いているかどうかを確認します。届いていれば、ビジネス・オブジェクトのペイロードが、WebSphere Business Integration Message Broker アプリケーション経由でレガシー・メッセージとして正常にデリバリーされていることになります。メッセージが見つからない場合は、キュー `queue://crossworlds.queue.manager/Samples/WBIMB/FAIL` を調べて、WebSphere Business Integration Message Broker アプリケーションがメッセージを処理できなかったのかどうかを確認します。処理されていないときは、「*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、トレースを使用可能にします。これで、WebSphere Business Integration Message Broker アプリケーションのどの部分でエラーが発生したのかを判別できるようになります。
7. **イベント処理シナリオをテスト**します。ユーティリティ `sample_folder\mqsiput.exe` を使用して、レガシー項目メッセージを WebSphere Business Integration Message Broker アプリケーション経由でアダプターにデリバリーします。mqsiput ユーティリティの構文は次のとおりです。
- ```
mqsiput [queue] [queue manager] < [message file].
```
- デリバリー・メッセージのサンプル *LegacyItem.txt* は、*sample\_folder* ディレクトリにあります。このチュートリアルでは、次のように引き数を指定する必要があります。
- ```
mqsiput Samples/LegacyApp/Item/WBIMB crossworlds.queue.manager < LegacyItem.txt
```
- コマンド行でこのとおりに入力してください。正常に実行されると、項目メッセ

ージのサンプルが、レガシー・アプリケーションをシミュレートしている WebSphere Business Integration Message Broker アプリケーションにデリバリーされます。

注: ブローカーの実行グループには、ループバック・メッセージ・フローを追加することもできます。(このフローは、IBM Adapter for WebSphere Business Integration Message Broker のサンプル・メッセージ・フローとともに提供されています。) ループバック・メッセージ・フローを追加すると、アダプターから送信されたどのメッセージも、WebSphere Business Integration Message Broker アプリケーションで処理された後、アダプターに戻されるようになります。

WebSphere Business Integration Message Broker アプリケーションがメッセージを入力キューにデリバリーすると、アダプターはそのメッセージを取得して `Sample_WBIMB_LegacyItem_XMLDoc` ビジネス・オブジェクトに変換しようとします。アダプターにメッセージをポーリングさせる上で重要なことは、メッセージ・フォーマットを、`Sample_WBIMB_LegacyItem_XMLDoc` ビジネス・オブジェクトに関連付けられているフォーマット値 (メタオブジェクト `Sample_WBIMB_MO_Config` に指定されています) と確実に一致させることです。アダプターは、メッセージのフォーマットが `LI_UP` であることを識別すると、データ・ハンドラーを呼び出して、そのメッセージを、動詞が `Update` のビジネス・オブジェクト `Sample_WBIMB_LegacyItem_XMLDoc` に変換します。新しく作成されたビジネス・オブジェクトは、続いて `WBI Broker` にデリバリーされ、その後 `Visual Test Connector` に再デリバリーされます。

8. **メッセージのデリバリーを確認します。** これまでに説明したステップをすべて完了すると、シナリオが実践されます。その結果、Adapter for WBI Message Broker ではメッセージを取得して `Sample_WBIMB_LegacyContact` ビジネス・オブジェクトに変換できるようになり、また `Sample_WBIMB_LegacyContact` ビジネス・オブジェクトをメッセージに変換できるようになります。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アーカイブ 10
アダプター 1
 アダプターと関連ファイルのインストール 15
 依存関係 15
 インストール 13
 始動 44
 処理の概要 4
 停止 45
 ブローカーの互換性 14
 WBI Message Broker との相互作用 4
 WebSphere MQ Java クライアント・ライブラリー要件 15
 WebSphere MQ の互換バージョン 15
アダプター環境 13
アダプターの実行
 作業ロードマップ 43
アダプター用にインストールされた
 Windows ファイル構造 16
アダプター・フレームワーク 2
アプリケーション・タイムアウト 46
アンサブスクライブされたビジネス・オブジェクト 46
イベント処理
 概要 8
イベント・ポーリング 9
インストール
 検証 16
 作業ロードマップ 13
 統合ブローカー 13
インストールの検証 16
エラー
 アプリケーション・タイムアウト 46
 アンサブスクライブされたビジネス・オブジェクト 46
 コネクタがアクティブでない 47
 処理 46
 データ・ハンドラーによる変換 47
 入力フォーマットの多重定義 47
 JMS プロパティ 47
応答メッセージ
 メッセージ選択子によるフィルター操作 51

オブジェクトの作成
 作業ロードマップ 49

[カ行]

キュー
 サンプル・シナリオの定義 101
検索 9
構成
 コネクタ 19
 サンプル・シナリオの定義 102
 データ・ハンドラー 67
 WBI Message Broker および
 InterChange Server の標準的な構成 3
コネクタ
 アダプター との違い 2
 構成 19
 サポートされる動詞 5
 実行 43
コネクタがアクティブでない 47
コネクタ固有のプロパティ 93
コネクタの実行 43
コネクタの始動 44
コネクタの停止 45
コネクタ用としてインストールされた
 UNIX ファイルのファイル構造 17
コネクタ・コントローラ 2
コネクタ・フレームワーク 2
コラボレーション
 要求処理 5

[サ行]

作業
 インストール 13
 オブジェクトの作成 49
 コンピューターの実行 43
 データ・ハンドラーの構成 67
作業ロードマップ 1
始動ファイル
 構成 43
進行中のキュー 8, 9
静的メタオブジェクト 54
 作成 58

[タ行]

データ・ハンドラー
 およびメッセージ要求 5

データ・ハンドラー (続き)

 構成 67
 構成の作業ロードマップ 67
 指定 67
 入力キューへのマッピング 60
データ・ハンドラーの入力キューへのマッピング 60
同期
 サポートされる動詞 5
 メッセージ要求 5, 6
統合ブローカー 3
動詞
 サポートされる 5
 同期要求でサポートされる動詞 5
 非同期要求でサポートされる動詞 5
動的メタオブジェクト 54
 作成 60
動的メタオブジェクト・ヘッダー属性 61
トレース 48

[ナ行]

入力キュー
 データ・ハンドラーへのマッピング 60
入力フォーマットの多重定義 47

[ハ行]

ビジネス・オブジェクト 49
 作成 50
 サポートされる動詞 5
 変更 51
非同期
 サポートされる動詞 5
 メッセージ要求 5, 6
フィードバック・コード
 応答メッセージ内 7
フォーマット
 イベント・ポーリングによって開始される変換 3
ブローカーの互換性 14

[マ行]

メタオブジェクト 49
 および動的の JMS ヘッダー 62
 作成 53
 静的と動的との違い 54
 静的の作成 58

メタオブジェクト (続き)

動的

ポーリング時の取り込み 62

動的作成のステップ 65

動的の作成 60

動的のヘッダー属性 61

プロパティ 54

メタオブジェクト属性

読み取りおよび書き込み 61

メッセージ

イベント・ポーリングによって開始さ

れるルーティング 3

メッセージ記述子ヘッダー

参照: MQMD

メッセージ選択子 51

メッセージ要求

およびデータ・ハンドラー 5

概要 5

処理 5

同期 5

同期処理 6

非同期 5

非同期処理 6

メッセージ・フロー

変更 68

メッセージ・フローの変更 68

メッセージ・フローを変更するステップ

68

[ヤ行]

要求

参照: メッセージ要求、要求処理

要求メッセージ

MQMD フィールドの値 6

用語 1

[ラ行]

リカバリー 10

ローカライズされたデータ 15

ロケール依存データ 15

A

ABON_APPRESPONSETIMEOUT 46

Adapter Development Kit (ADK) 2

AIX 14

AppID 61

ApplicationPassword プロパティ 94

ApplicationUserName プロパティ 94

APP_RESPONSE_TIMEOUT 47

ArchiveQueue 10

ArchiveQueue プロパティ 95

B

Business Object Designer 51

C

CCSID プロパティ 95

Channel プロパティ 95

ConfigurationMetaObject プロパティ 95

CONNECTOR_NOT_ACTIVE 47

correlationID 8, 52

Create (オブジェクトの動詞) 5

D

DataHandlerClassName 55

DataHandlerClassName プロパティ 95

DataHandlerConfigMO 55

DataHandlerConfigMO プロパティ 95

DataHandlerMimeType 55

DataHandlerMimeType プロパティ 95

DefaultVerb プロパティ 96

Delete (オブジェクトの動詞) 5

DeliveryCount 61

DeliveryMode 61

Destination 61

doVerbFor() メソッド 5

E

EnableMessageProducerCache プロパティ

96

ErrorQueue プロパティ 96

event 9

Exists (オブジェクトの動詞) 5

Expiration 61

F

fail on startup 10

G

gotAppEvents() メソッド 9, 10, 54

GroupID 61

GroupSeq 61

H

HostName プロパティ 96

HP-UX 14

I

ignore 10

InDoubtEvents 10

InDoubtEvents プロパティ 96

InputFormat 56

InputQueue 56

InputQueue プロパティ 96

J

Java TM Message Service (JMS) 4

Java 仮想マシン 15

JavaTM Message Service (JMS) 3

JMS

メタオブジェクト・プロパティ 57

JMS プロパティ 62

エラー 47

JMS ヘッダー

および動的メタオブジェクトの属性

62

JMSProperties 61

L

log error 10

M

MessageID 61

MQMD

応答メッセージ記述子ヘッダー 7

メッセージ要求フィールドの値 6

Expiry 6

Format 6

MsgType 6

Persistence 6

ReplyToQ 6

Report 6

N

NO_SUBSCRIPTION_FOUND 46

O

Object Discovery Agent 50

OutputFormat 56

OutputQueue 56

P

pollForEvents() 9

pollForEvents() メソッド 8

Port プロパティ 97
Priority 61

R

Redelivered 61
ReplyToQueue 8, 61
 同期処理 6
ReplyToQueue プロパティ 97
reprocess 10
ResponseTimeout 57
 同期処理 6
response_selector 51
Retrieve by Content (オブジェクトの動詞) 5
Retrieve (オブジェクトの動詞) 5

S

selectorstring 51
SessionPoolSizeForRequests プロパティ 98
Solaris 14

T

TimeoutFatal 46, 57
TimeStamp 61
Type 61

U

UnsubscribedQueue 46
UnsubscribedQueue プロパティ 98
Update (オブジェクトの動詞) 5
UseDefaults プロパティ 98
UserID 61

W

WBI Message Broker
 アダプターとの相互作用 4
 XML に対するネイティブ・サポート 50
WebSphere Application Server Enterprise
 アダプターの互換バージョン 14
WebSphere Business Integration Adapter Framework
 アダプターの互換バージョン 14
WebSphere Business Integration システム 3
WebSphere Integration Message Broker バージョン 2.2 3

WebSphere InterChange Server
 アダプターの互換バージョン 14
WebSphere MQ
 アダプターの互換バージョン 15
WebSphere MQ Java クライアント・ライブラリー 15
WebSphere MQ キュー・マネージャー 4
Windows 2000 14

X

XML
 アダプターで使用する推奨フォーマット 4
 および WBI Message Broker のサポート 50
 メッセージ・フローを変更するステップ 68
XML ODA 50
XML データ・ハンドラー 67

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証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

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。

著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

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
Lotus
Lotus Domino
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。



WebSphere Business Integration Adapter Framework V2.4.0



Printed in Japan