

**IBM WebSphere Business Integration  
Adapters**



## **Adapter for iSeries ユーザーズ・ガイド**

*バージョン 1.1.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、55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、コネクター・バージョン 1.0.x、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： IBM WebSphere Business Integration Adapters  
Adapter for iSeries User Guide  
Version 1.1.x

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第 1 刷 2004.1

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

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

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

© Copyright IBM Japan 2004

---

# 目次

本書について	v
対象読者	v
関連文書	v
表記上の規則	vi
<b>本リリースの新機能</b>	<b>vii</b>
バージョン 1.1.x	vii
以前のバージョン	vii
<b>第 1 章 概要</b>	<b>1</b>
iSeries システムおよび AS/400 システムの概要	1
アダプターの動作方法	4
<b>第 2 章 iSeries アダプターのインストールと構成</b>	<b>7</b>
互換性	7
前提条件	8
iSeries アダプターと関連ファイルのインストール	9
コネクターの始動	9
コネクターの構成	9
<b>第 3 章 コネクター用ビジネス・オブジェクトの開発</b>	<b>11</b>
コネクター・ビジネス・オブジェクトの処理	11
コネクター・ビジネス・オブジェクトの構造	11
ビジネス・オブジェクト Connection の XSD スキーマ	12
ビジネス・オブジェクト PAS400rpg2 の XSD スキーマ	14
ビジネス・オブジェクトの属性レベルのアプリケーション・テキスト	16
<b>付録 A. コネクターの標準構成プロパティ</b>	<b>19</b>
新規プロパティと削除されたプロパティ	19
標準コネクター・プロパティの構成	19
標準プロパティの要約	21
標準構成プロパティ	24
<b>付録 B. Connector Configurator</b>	<b>37</b>
Connector Configurator の概要	37
Connector Configurator の始動	38
System Manager からの Configurator の実行	39
コネクター固有のプロパティ・テンプレートの作成	39
新しい構成ファイルを作成	42
既存ファイルの使用	43
構成ファイルの完成	44
構成ファイル・プロパティの設定	45
構成ファイルの保管	52
構成ファイルの変更	52
構成の完了	53
グローバル化環境における Connector Configurator の使用	53
<b>特記事項</b>	<b>55</b>
プログラミング・インターフェース情報	56
商標	57



---

## 本書について

IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Business Integration Adapter ポートフォリオは、主要な e-business テクノロジー、エンタープライズ・アプリケーション、レガシー・システム、およびメインフレーム・システム向けに、統合コネクティビティーを提供します。本製品には、コンポーネントをカスタマイズ、作成、および管理するためのツールとテンプレートが含まれており、これにより、ビジネス・プロセスの統合を実現します。

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

---

## 対象読者

本書は、WebSphere Business Integration システムの一部としてコネクターを実装する WebSphere のコンサルタントおよびお客様を対象としています。本書の情報を利用するには、以下の分野に関する十分な知識が必要です。

- コネクター開発
- ビジネス・オブジェクト開発
- OS/400 アプリケーションのアーキテクチャー

---

## 関連文書

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

関連文書については、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、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 Broker (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、以下をご覧ください。

<http://www.ibm.com/software/webservers/appserv/library.html>

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

---

## 表記上の規則

本書では、以下のような規則を使用しています。

---

Courier フォント	コマンド名、ファイル名、入力情報、システムが画面に出力した情報など、記述されたとおりの値を示します。
<b>太字</b>	初出語を示します。
<i>イタリック、イタリック</i>	変数名または相互参照を示します。
青のアウトライン	オンラインで表示したときのみ見られる青のアウトラインは、相互参照用のハイパーリンクです。アウトラインの内側をクリックすると、参照先オブジェクトにジャンプします。
{ }	構文の記述行の場合、中括弧 { } で囲まれた部分は、選択対象のオプションです。1 つのオプションのみを選択する必要があります。
[ ]	構文の記述行の場合、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)
/ および ¥	本書では、ディレクトリー・パスに円記号 (¥) 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 WebSphere Business Integration システム製品のパス名は、使用システムで製品がインストールされた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	% 記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 です。これは、UNIX 環境変数 text の値を示します。
<i>ProductDir</i>	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

---

### バージョン 1.1.x

RPG (Remote Program Generator) 3 のサポートが追加されました。

バージョン 1.1 以降、Adapter for iSeries 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に関する情報が本書から移動しました。この情報の新しい掲載先については、第 2 章 (9 ページの『iSeries 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。

---

### 以前のバージョン

以前のバージョンでの変更は、以下のとおりです。

#### バージョン 1.0.x

バージョン 1.0.x は、IBM WebSphere Business Integration Adapter for iSeries の最初のリリースです。このアダプターは、iSeries システムまたは AS/400 システムで RPG (Remote Program Generator) 4 プログラムを実行する機能が提供されます。





---

## 第 1 章 概要

この章では、IBM WebSphere Business Integration Adapter for iSeries について説明します。このアダプターは、IBM Toolbox for Java (Java<sup>TM</sup> クラスのセット) を使用して、iSeries システム上または AS/400 システム上にある、**RPG** (Report Program Generator、モジュラー・プログラミング用の言語の 1 つ) で作成されたプログラムを実行する機能が提供されます。IBM Toolbox for Java には、これらのプログラムの操作や実行のためのクラスのセットが用意されています。アダプターは、これらのクラスと、着信ビジネス・オブジェクトからの情報を使用して、プログラムのパラメーター・リストを作成し、プログラムを実行します。Adapter for iSeries は現在、要求の処理のみをサポートしています。

アダプターは、アプリケーション固有のコンポーネントとコネクター・フレームワークで構成されています。アプリケーション固有のコンポーネントには、特定のアプリケーションに応じて調整されたコードが含まれます。コネクター・フレームワークは統合ブローカーとアプリケーション固有のコンポーネントとの仲介役として機能し、そのコードはいずれのアダプターにも共通です。コネクター・フレームワークは、統合ブローカーとアプリケーション固有のコンポーネントとの間で以下のようなサービスを提供します。

- ビジネス・オブジェクトを送信します。
- 始動メッセージおよび管理メッセージの交換を管理します。

本書では、アプリケーション固有のコンポーネントおよびコネクター・フレームワークについて説明します。ここでは、この 2 つのコンポーネントをまとめてアダプターと呼びます。

統合ブローカーとアダプターの関係の詳細については、「*IBM WebSphere InterChange Server システム管理ガイド*」または「*WebSphere MQ Integrator Broker 用インプリメンテーション・ガイド*」を参照してください。

本章の内容は、次のとおりです。

- 『iSeries システムおよび AS/400 システムの概要』
- 4 ページの『アダプターの動作方法』

---

## iSeries システムおよび AS/400 システムの概要

iSeries および AS/400 は、IBM がリレーショナル・データベース向けに販売している、主要なネットワーク・システムです。これらは、信頼性に優れ、数百のユーザーに対応することができる、ミッドレンジ・コンピューターです。特性として保全本性とセキュリティーに優れているので、「重要な」アプリケーションで使用することができます。

**RPG** (Report Program Generator) は、モジュラー・プログラミング用の言語です。通常、RPG プログラムは、iSeries システムや AS/400 システムではバッチ・アップロードに使用されます。

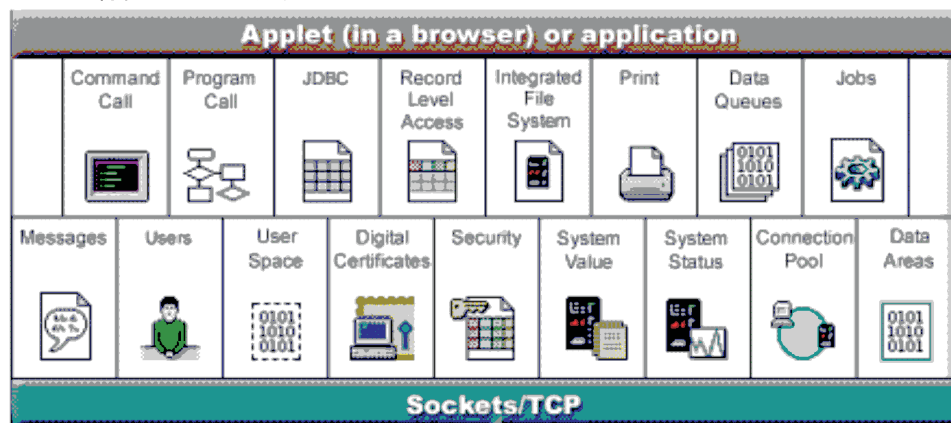
ホスト・サーバーは、3 ページの図 1 に示すように、アプリケーションを実行している装置 (クライアント PC など) からの要求を処理して、文書の印刷などのタスクを実行できるようにします。iSeries コンピューターと AS/400 コンピューターは機能の充実したサーバーであり、ファイル操作、データベース操作、アプリケーション実行、メール処理、印刷処理、マルチメディア処理、FAX 処理、無線通信などの多数のタスクを同時に実行することができます。タスク・サーバーは、システム上でそれぞれ個別のジョブとして実行されます。また、サーバー・ジョブは、それぞれソケット接続を介してデータ・ストリームを送受信します。

このようなホスト・サーバーの 1 つに、リモート・コマンドおよび分散プログラム呼び出しサーバーがあります。このサーバーは、iSeries システムまたは AS/400 システムでのプログラム実行を処理します。

IBM Toolbox for JAVA には多数のパッケージが含まれており、これらのパッケージはさまざまな機能に対応しています。例えば、アクセス・クラスは、サインオン情報の管理、ソケット接続の作成と維持、およびデータの送受信を行います。一方、コマンド呼び出しクラスは、iSeries と AS/400 のバッチ・コマンドを実行します。

IBM の Adapter for iSeries は、アクセス・クラスとプログラム呼び出しクラスを使用して RPG プログラムを呼び出します。iSeries (または AS/400) 形式と Java 形式の間で数値データと文字データを変換する機能は、データ変換クラスによって提供されます。

JVM が稼働しているクライアント



AS/400 システム

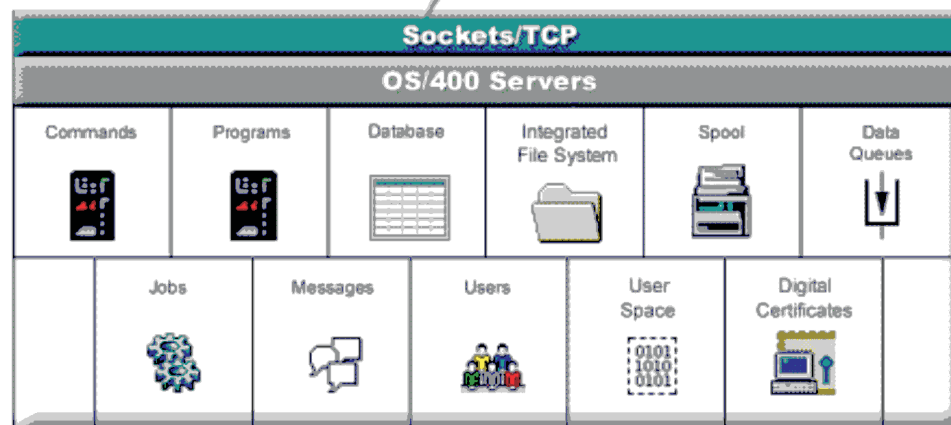


図 1. AS/400 クライアント - サーバー・アーキテクチャーの概要

AS/400 上で稼働する OS/400 はさまざまなタイプのタスクを処理することができますが、Adapter for iSeries から使用されるのはリモート・コマンドおよび分散プログラム呼び出しサーバーのみです。このサーバーは、AS/400 システムでのプログラム実行を処理します。

Adapter for iSeries を使用した場合のクライアントからサーバーへの接続のしくみを、4 ページの図 2 に示します。

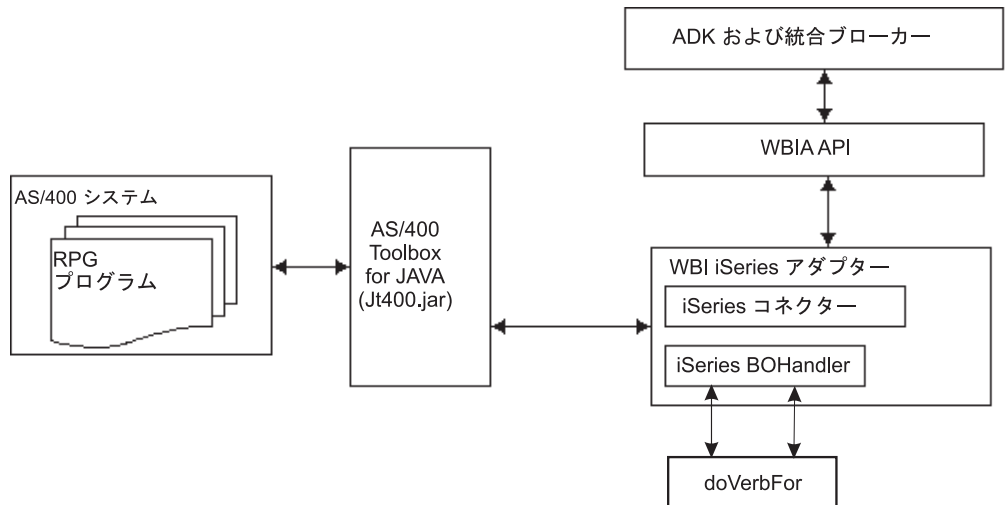


図2. Adapter for iSeries を使用した場合の接続のしくみ

## アダプターの動作方法

ここでは、アダプターでビジネス・オブジェクトがどのように処理されるかについて説明します。

### ビジネス・オブジェクトの処理

アダプターは、統合ブローカーからビジネス・オブジェクト要求を受信すると、RPG プログラムのパラメーター・リストを作成し、iSeries システムまたは AS/400 システムとの接続を確立して、RPG プログラムを実行します。

着信したビジネス・オブジェクトには、接続関連の子属性が含まれています。この属性の情報は、iSeries システムまたは AS/400 システムへの接続に使用されます。

### コネクタの動作

コネクタは、統合ブローカーとビジネス・オブジェクト・ハンドラーの間で、ビジネス・オブジェクトを受け渡します。ビジネス・オブジェクトは、以下の手順で処理されます。

1. コネクタがフレームワークに BOHandler を登録します。
2. フレームワークから BOHandler に BO 要求が送信されます。
3. BOHandler で、着信ビジネス・オブジェクトの属性の情報に基づいて RPG プログラムのパラメーター・リストが作成されます。
4. BOHandler から、iSeries システムまたは AS/400 システムで実行される RPG プログラムが呼び出されます。

**注:** これは、基本的には iSeries システム上または AS/400 システム上の RPG プログラムを実行するための呼び出しです。これらのシステムは、この呼び出しの後、成功または失敗を通知するメッセージを戻します。

5. BOHandler が RPG プログラムの実行結果をアダプター・フレームワークに戻します。また、戻されたパラメーターをビジネス・オブジェクトに取り込みます。

アダプターは、Java で作成されており、次の 2 つのコンポーネントで構成されています。

- コネクター
- BO ハンドラー



---

## 第 2 章 iSeries アダプターのインストールと構成

この章では、コネクターのインストールと構成のプロセスについて説明します。本章の内容は、次のとおりです。

- 『互換性』
- 8 ページの『前提条件』
- 9 ページの『iSeries アダプターと関連ファイルのインストール』
- 9 ページの『コネクターの構成』
- 9 ページの『標準コネクター・プロパティ』

---

### 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 iSeries バージョン 1.1 は、以下のバージョンのアダプター・フレームワークおよび統合ブローカーでサポートされています。

**アダプター・フレームワーク:** WebSphere Business Integration Adapter Framework、バージョン 2.1、2.2、2.3.x、および 2.4

**統合ブローカー:**

- WebSphere InterChange Server、バージョン 4.2.x
- WebSphere MQ Integrator、バージョン 2.1.0
- WebSphere MQ Integrator Broker、バージョン 2.1.0
- WebSphere Business Integration Message Broker、バージョン 5.0
- WebSphere Application Server Enterprise、バージョン 5.0.2 (WebSphere Studio Application Developer Integration Edition、バージョン 5.0.1 と併用)

例外については、「リリース情報」を参照してください。

**注:** 統合ブローカーおよびその前提条件のインストールに関する説明については、以下の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。

WebSphere Message Brokers (WebSphere MQ Integrator Broker、WebSphere MQ Integrator、および WebSphere Business 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 WebSphere Message Broker のインストールに関する資料を参照してください。この資料の一部は、<http://www.ibm.com/software/integration/mqfamily/library/manualsa/> の Web サイトで参照できます。

WebSphere Application Server については、「IBM WebSphere Business Integration Adapters 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。

---

## 前提条件

コネクタを使用するには、ご使用の環境に以下のものが用意されている必要があります。

1. 以下の Java 環境と JAR ファイルが必要です。

- JDK 1.3 以降
- JSSE (Java Secure Socket Extension) 1.0
- Jt400.jar ファイル

**注:** IBM Toolbox for Java (ライセンス製品 5722-JC1) V5R2 のファイルは、<http://www-1.ibm.com/servers/eserver/iseries/toolbox/downloads.htm> の Toolbox Web サイトからダウンロードできます。jt400.jar は、%Product\_dir%\%connectors%iSeries ディレクトリーにコピーする必要があります。

- WBIA.jar ファイル
- CrossWorlds.jar ファイル
- BIA\_iSeries.jar ファイル

2. Adapter for iSeries は、以下のいずれかのバージョンの OS/400 を介して AS/400 に接続するように設計されています。

- バージョン 4 リリース 1 から 3
- 累積プログラム一時修正 (PTF) レベル C7105370 以降を適用済みのバージョン 3 リリース 7
- PTF レベル C7126320 以降を適用済みのバージョン 3 リリース 2

3. OS/400 のホスト・サーバー・オプションがインストールされ、実行されている必要があります。

**注:** OS/400 データ・キュー・サーバーで peek 関数が正常に実行されるようにするには、PTF を適用する必要があります。以下のリストを参照して、適切な PTF を適用してください。

- V4R2、PTF は不要です。
- V4R1、5769SS1: PTF SF42813
- V3R7、5716SS1: PTF SF42398
- V3R2、5763SS1: PTF SF43295

4. RPG IV が実行されている必要があります。



---

## iSeries アダプターと関連ファイルのインストール

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

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

---

### コネクターの始動

コネクターの始動と停止、およびコネクターの一時始動ログ・ファイルについては、「*WebSphere MQ Integrator Broker 用インプリメンテーション・ガイド*」を参照してください。WebSphere ICS を使用している場合は、「*IBM WebSphere InterChange Server システム管理ガイド*」を参照してください。

---

### コネクターの構成

Adapter for iSeries の構成には、次のセクションで説明する標準コネクター・プロパティを使用します。また、その次のセクションで説明するコネクター固有のプロパティも使用します。

#### 標準コネクター・プロパティ

標準構成プロパティにより、すべてのコネクターによって使用される情報が提供されます。標準構成プロパティの資料については、19 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。

**重要:** 本書で解説するコネクターはすべての統合ブローカーをサポートしているため、関係する構成プロパティは、すべてのブローカーに適合する構成プロパティのみになります。

コネクターを実行する前に、少なくとも以下の標準コネクター構成プロパティを設定しておく必要があります。

- AgentTraceLevel
- ApplicationName
- ControllerStoreAndForwardMode
- ControllerTraceLevel
- DeliveryTransport

#### コネクター固有のプロパティ

コネクター固有の構成プロパティは、コネクターが実行時に必要とする情報を提供します。また、コネクター固有の構成プロパティを使用すれば、コネクター・エージェント内の静的な情報やロジックを、エージェントの再コーディングや再ビルドを行わずに変更することができます。

表 1 に、コネクターのコネクター固有の構成プロパティを示します。各プロパティの説明については、以降のセクションを参照してください。

表 1. コネクタ固有のプロパティ

名前	指定可能な値	デフォルト値	必須
ApplicationName	iSeriesAdapter	なし	はい
UseDefaults	デフォルト値	なし	はい
MessageFileName	BIA_iSeriesAdapter.txt	BIA_iSeriesAdapter.txt	いいえ

## ApplicationName

各コネクタに指定する必要がある、固有の名前です。

## UseDefaults

通常、プログラムの入力パラメーターの一部は、値が一定です。これらを表す属性は、デフォルト値を持つように設計することができます。UseDefaults プロパティが true に設定されている場合、デフォルト値が指定されていなければ、アダプターはエラーにより停止し、VerbProcessingFailedException をスローします。UseDefaults が設定されていない場合や false に設定されている場合は、デフォルト値が指定されていなければ、MaxLength に指定されている長さになるようにスペースを埋め込んだストリングが属性の値としてアダプターで作成されます。

## MessageFileName

エラー・メッセージ・ファイルの名前とパスです。エラー・メッセージ・ファイルがメッセージの標準位置である %CROSSWORLDS%\connectors\messages にない場合に指定します。メッセージ・ファイルの名前が完全修飾パスと共に指定されていない場合、そのメッセージ・ファイルは、HOME 環境変数または始動パラメーター user.home に指定されているディレクトリーにあるものと見なされます。コネクタ・メッセージ・ファイルが存在しない場合は、ファイル BIA\_iSeriesAdapter.txt がメッセージ・ファイルとして使用されます。

---

## 第 3 章 コネクタ用ビジネス・オブジェクトの開発

この章では、コネクタによって受け渡されるビジネス・オブジェクトの構造について説明します。また、必須の属性や、コネクタでのビジネス・オブジェクトの処理についても説明します。本章の内容は、次のとおりです。

- 『コネクタ・ビジネス・オブジェクトの処理』
- 『コネクタ・ビジネス・オブジェクトの構造』

---

### コネクタ・ビジネス・オブジェクトの処理

コネクタは、統合ブローカーと AS/400 システムの間で、ビジネス・オブジェクトを受け渡します。

コネクタは、統合ブローカーからビジネス・オブジェクトを渡されると、以下の処理を実行します。

1. 接続関連の子属性の情報を使用して、AS/400 システムに接続します。
2. BO の属性に基づいて、RPG プログラムのパラメーター・リストを作成します。
3. BO に対応する RPG プログラムを実行します。
4. プログラムの実行結果 (成功または失敗) を戻します。

---

### コネクタ・ビジネス・オブジェクトの構造

Adapter for iSeries のビジネス・オブジェクトは、フラット・ビジネス・オブジェクトです。各属性は、入力、出力、または入出力パラメーターになります。属性の 1 つは、Business Object Designer 用のキーになっている必要があります。

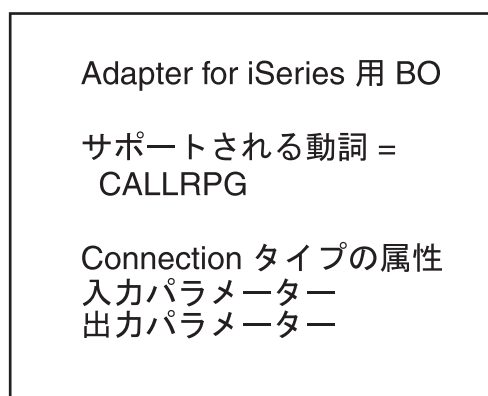


図 3. iSeries の親ビジネス・オブジェクト

また、タイプが Connection の子属性もあります。これには、AS/400 マシンへの接続に関する情報である HostName、UserName、および Password が含まれています。



図 4. iSeries の子ビジネス・オブジェクト

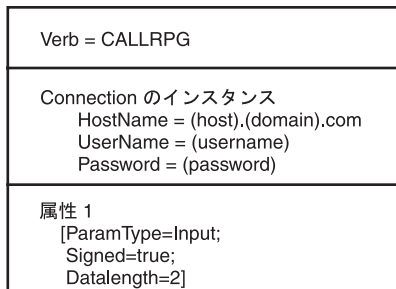


図 5. ビジネス・オブジェクトの例

上記の図の大括弧で囲まれている情報は、アプリケーション固有の情報を表しています。

ビジネス・オブジェクトを作成するには、Business Object Designer を使用します。ビジネス・オブジェクト定義を作成してから、必須の属性を追加します。その後で、ビジネス・オブジェクトがサポートされるようにコネクタを構成します。

## ビジネス・オブジェクト Connection の XSD スキーマ

```

<?xml version="1.0" encoding="utf-8" standalone="no" ?>
- <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.ibm.com/websphere/crossworlds
2002/BOSchema/Connection"
xmlns:bx=
"http://www.ibm.com/websphere/crossworlds/2002/BOSchema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <xsd:annotation>
<xsd:documentation>Fri Jun 06 21:20:50 IST 2003
</xsd:documentation>
</xsd:annotation>
- <xsd:element name="Connection">
- <xsd:annotation>
- <xsd:appinfo>

```

```

<bx:boDefinition version="3.0.0" />
</xsd:appinfo>
</xsd:annotation>
- <xsd:complexType>
- <xsd:sequence>
- <xsd:element name="UserName" default="[insert username]" minOccurs="1">
- <xsd:annotation>
- <xsd:appinfo>
<bx:boAttribute>
<bx:attributeInfo isForeignKey="false" isKey="true" />
</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
<xsd:maxLength value="255" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
- <xsd:element name="Password" default="[insert password]" minOccurs="1">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boAttribute>
<bx:attributeInfo isForeignKey="false" isKey="true" />
</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
<xsd:maxLength value="255" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
- <xsd:element name=
"HostName" default="(hostname).(domain name).com" minOccurs="1">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boAttribute>
<bx:attributeInfo isForeignKey="false" isKey="true" />
</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
<xsd:maxLength value="255" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

```

<xsd:element name=
"ObjectEventId" type="xsd:string" minOccurs="0" />
</xsd:sequence>
<xsd:attribute name="version" type="xsd:token" default="0.0.0" />
<xsd:attribute name="delta" type="xsd:boolean" default="false" />
- <xsd:attribute name="verb" use="required">
- <xsd:simpleType>
- <xsd:restriction base="xsd:NMTOKEN">
<xsd:enumeration value="CALLRPG" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

---

## ビジネス・オブジェクト PAS400rpg2 の XSD スキーマ

```

<?xml version="1.0" encoding="utf-8" standalone="no" ?>
- <xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.ibm.com/websphere
/crossworlds/2002/BOSchema/PAS400rpg2"
xmlns:Connection=
"http://www.ibm.com/websphere/crossworlds/2002
/BOSchema/Connection" xmlns:bx="http://www.ibm.com/websphere/crossworlds/2002
/BOSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <xsd:annotation>
<xsd:documentation>Fri Jun 06 21:20:50 IST 2003</xsd:documentation>
</xsd:annotation>
<xsd:import namespace=
"http://www.ibm.com/websphere/crossworlds/2002
/BOSchema/Connection" schemaLocation="Connection.xsd" />
- <xsd:element name="PAS400rpg2">
- <xsd:annotation>
- <xsd:appinfo>
<bx:boDefinition version="3.0.0" />
</xsd:appinfo>
</xsd:annotation>
- <xsd:complexType>
- <xsd:sequence>
- <xsd:element name="attr1" minOccurs="1">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boAttribute>
<bx:appSpecificInfo>
ParamType=Input;PackedDec=true;DecimalPositions=0
</bx:appSpecificInfo>
<bx:attributeInfo isForeignKey="false" isKey="false" />

```

```

</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
<xsd:maxLength value="3" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
- <xsd:element name="attr2" minOccurs="1">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boAttribute>
<bx:appSpecificInfo>
ParamType=Input;PackedDec=true;DecimalPositions=0
</bx:appSpecificInfo>
<bx:attributeInfo isForeignKey="false" isKey="false" />
</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
<xsd:maxLength value="2" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
- <xsd:element name="attr3" minOccurs="0">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boAttribute>
<bx:appSpecificInfo>
ParamType=Inout;PackedDec=true;DecimalPositions=0
</bx:appSpecificInfo>
<bx:attributeInfo isForeignKey="false" isKey="false" />
</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
<xsd:maxLength value="3" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
- <xsd:element name="Connection" minOccurs="1">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boAttribute>
<bx:appSpecificInfo>ParamType=Input</bx:appSpecificInfo>

```

```

<bx:attributeInfo isForeignKey="false" isKey="true" />
<bx:childObjectInfo relationship="Containment" version="3.0.0" />
</bx:boAttribute>
</xsd:appinfo>
</xsd:annotation>
- <xsd:complexType>
- <xsd:sequence>
<xsd:element ref="Connection:Connection" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ObjectEventId" type="xsd:string" minOccurs="0" />
</xsd:sequence>
<xsd:attribute name="version" type="xsd:token" default="0.0.0" />
<xsd:attribute name="delta" type="xsd:boolean" default="false" />
- <xsd:attribute name="verb" use="required">
- <xsd:simpleType>
+ <xsd:restriction base="xsd:NMTOKEN">
- <xsd:enumeration value="CALLRPG">
- <xsd:annotation>
- <xsd:appinfo>
- <bx:boVerb>
<bx:appSpecificInfo>
/QSYS.LIB/XYZ.LIB/RPG3.PGM
</bx:appSpecificInfo>
</bx:boVerb>
</xsd:appinfo>
</xsd:annotation>
</xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

---

## ビジネス・オブジェクトの属性レベルのアプリケーション・テキスト

以下に示す情報は、ビジネス・オブジェクトの属性レベルのアプリケーション・テキストの一部です。

表 2. BO 属性

プロパティ	値	説明
ParamType	Input/Output/Inout	属性が表すパラメーターのタイプを示します。
Offset	任意の整数値	バイト配列内でパラメーター値の開始位置からのオフセットを示します。



表 2. BO 属性 (続き)

プロパティ	値	説明
Signed	true/false	データ型が integer、short、または long の場合に、符号付きかどうかを示します。このプロパティが設定されていない場合、値は符号なしと見なされます。
DataLength	任意の整数値	データ型が integer、short、または long の場合に適用されます。データ型が符号の有無を指定できるものである場合に、データ長の識別に使用されます。このプロパティが設定されていない場合、デフォルト値の 4 が使用されます。
DecimalPositions	任意の整数値	データ型がゾーン 10 進数またはパック 10 進数である場合に適用されます。このプロパティの値は、小数部の桁数を表します。
PackedDec	true/false	true に設定されている場合、属性はパック 10 進数を表します。
ZonedDec	true/false	true に設定されている場合、属性はゾーン 10 進数を表します。

## iSeries 用または AS/400 用の Toolbox からのデータ変換

iSeries 用または AS/400 用の Toolbox には、データ変換クラスが含まれています。次の表に、iSeries と AS/400 のデータ型とそれに対応する IBM WebSphere Business Integration のデータ型の組み合わせを、使用するデータ変換クラスと共に示します。

表 3. 変換されるデータ型とデータ変換クラス

iSeries/AS400 のデータ型	IBM WBI のデータ型	データ変換クラス
AS/400 形式の符号付き 2 バイト数値	Integer、アプリケーション固有の情報 - Signed=true; DataLength=2	AS400Bin2
AS/400 形式の符号付き 4 バイト数値	Integer、アプリケーション固有の情報 - Signed=true; DataLength=4	AS400Bin4
AS/400 形式の符号付き 2 バイト浮動小数点数	Float	AS400Float4
AS/400 形式の符号付き 4 バイト浮動小数点数	Double	AS400Float8
AS/400 形式の符号なし 2 バイト数値	Integer、アプリケーション固有の情報 - Signed=false; DataLength=2	AS400UnsignedBin2

表 3. 変換されるデータ型とデータ変換クラス (続き)

iSeries/AS400 のデータ型	IBM WBI のデータ型	データ変換クラス
AS/400 形式の符号なし 4 バイト数値	Integer、アプリケーション固有の情報 - Signed=false; DataLength=4	AS400UnsignedBin4
AS/400 形式のパック 10 進数	String (MaxLength 属性プロパティに桁数を指定する必要があります)、アプリケーション固有の情報 - DecimalPositions=<小数部の桁数>; PackedDec=true	AS400PackedDecimal
AS/400 形式のゾーン 10 進数	String (MaxLength 属性プロパティに桁数を指定する必要があります)、アプリケーション固有の情報 - DecimalPositions=<小数部の桁数>; ZonedDec=true	AS400ZonedDecimal
文字データ	String (MaxLength に文字データの最大長を指定します)	AS400Text
日付データ	String (MaxLength に日付データの最大長を指定します)	AS400Text

---

## 付録 A. コネクターの標準構成プロパティ

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

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (これらをまとめて WebSphere Message Brokers (WMQI) と呼びます。)
- WebSphere Application Server (WAS)

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

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

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

---

### 新規プロパティと削除されたプロパティ

本リリースには、次の標準プロパティが追加されました。

#### 新規プロパティ

- XMLNamespaceFormat

#### 削除されたプロパティ

- RestartCount
- RHF2MessageDomain

---

### 標準コネクタ・プロパティの構成

Adapter コネクタには 2 つのタイプの構成プロパティがあります。

- 標準構成プロパティ
- コネクタ固有のプロパティ

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

## Connector Configurator の使用

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

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

## プロパティ値の設定と更新

プロパティ・フィールドのデフォルトの長さは 255 文字です。

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

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

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

### • 動的

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

### • コンポーネント再始動

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

### • サーバー再始動

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

### • エージェント再始動 (ICS のみ)

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

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

## 標準プロパティの要約

表 4 は、標準コネクタ構成プロパティの早見表です。コネクタによっては使用されないプロパティがあります。また、標準プロパティの依存関係の決定には `RepositoryDirectory` が関係するので、使用する統合ブローカーによってプロパティの設定が異なる可能性があります。

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

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

表 4. 標準構成プロパティの要約 (続き)

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

表 4. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE> でなければならぬ
MessageFileName	パスまたはファイル名	InterchangeSystem.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	CONNECTORNAME/MONITORQUEUE	コンポーネント再始動	JMS トランスポートのみ: DuplicateEvent Elimination は True でなければならぬ
OADAutoRestartAgent	True または False	False	動的	Repository Directory は <REMOTE> でなければならぬ
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE> でなければならぬ
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE> でなければならぬ
PollEndTime	HH:MM	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒) no (ポーリングを使用不可にする) key (コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力された場合にのみポーリングする)	10000	動的	
PollQuantity	1 から 500	1	エージェント再始動	JMS トランスポートのみ: Container Managed Events を指定
PollStartTime	HH:MM(HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
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 4. 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	
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> の場合は CwBO
WsifSynchronousRequest Timeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	WAS のみ
XMLNamespaceFormat	short、long	short	エージェント再始動	WebSphere MQ Message Brokers および WAS のみ

## 標準構成プロパティ

このセクションでは、各標準コネクタ構成プロパティの定義を示します。

### AdminInQueue

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

デフォルト値は CONNECTORNAME/ADMININQUEUE です。



## AdminOutQueue

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

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

## AgentConnections

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用されます。

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

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

## AgentTraceLevel

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

## ApplicationName

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

## BrokerType

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

## CharacterEncoding

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

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

## ConcurrentEventTriggeredFlows

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用されます。

コネクタ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・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 対応コネクタが、保証付きイベント・デリバリーを提供できるようになります。保証付きイベント・デリバリーでは、イベントはソース・キューから除去され、単一 JMS トランザクションとして宛先キューに配置されます。

デフォルト値は No value です。

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

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

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

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

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

## ControllerStoreAndForwardMode

RepositoryDirectory が <REMOTE> の場合のみ適用されます。

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

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

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

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

デフォルト値は true です。

## ControllerTraceLevel

RepositoryDirectory が <REMOTE> の場合のみ適用されます。

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

## DeliveryQueue

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルト値は CONNECTORNAME/DELIVERYQUEUE です。

## DeliveryTransport

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

- ICS がブローカー・タイプの場合は、DeliveryTransport プロパティの指定可能な値は MQ、IDL、または JMS であり、デフォルトは IDL になります。

- `RepositoryDirectory` がローカル・ディレクトリーの場合は、指定可能な値は `JMS` のみです。

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

## WebSphere MQ および IDL

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

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

- 非同期 (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、メモリ制限が発生することもあります。

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

この環境では、WebSphere MQ クライアント内でメモリが使用されるため、(サーバー側の) コネクタ・コントローラーと (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> の場合にのみ適用されます。

デフォルト値は 128M です。

## JvmMaxNativeStackSize

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

デフォルト値は 128K です。

## JvmMinHeapSize

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

デフォルト値は 1M です。

## jms.FactoryClassName

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

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

## jms.MessageBrokerName

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

デフォルト値は crossworlds.queue.manager です。

## jms.NumConcurrentRequests

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

デフォルト値は 10 です。

## jms.Password

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

デフォルトはありません。

## jms.UserName

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

デフォルトはありません。

## ListenerConcurrency

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

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

## Locale

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

ロケール名は、次の書式で指定します。

`ll_TT.codeset`

ここで、以下のように説明されます。

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

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

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

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

## LogAtInterchangeEnd

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用されます。

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>` の場合のみ適用されます。

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

## MessageFileName

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

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

**注:** 特定のコネクタについて、コネクタ独自のメッセージ・ファイルがあるかどうかを判別するには、該当するアダプターのユーザーズ・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機能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 により起動される 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 により起動される 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 により起動される 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には、OADAutoRestartAgent プロパティを true に設定する必要があります。

デフォルト値は 10 です。

## PollEndTime

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

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

## PollFrequency

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

- ポーリング・アクション間のミリ秒数。
- ワード key。コネクタは、コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力されたとき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。コネクタはポーリングを実行しません。このワードは小文字で入力します。

デフォルト値は 10000 です。

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

## PollQuantity

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

## PollStartTime

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

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

## RequestQueue

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

デフォルト値は CONNECTOR/REQUESTQUEUE です。

## RepositoryDirectory

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

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

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

## ResponseQueue

DeliveryTransport が JMS の場合のみ適用され、RepositoryDirectory が <REMOTE> の場合のみ必要です。

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

## RestartRetryCount

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

デフォルト値は 3 です。

## RestartRetryInterval

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

デフォルト値は 1 です。

## SourceQueue

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

JMS イベント・ストアを使用する JMS 対応コネクタでの保証付きイベント・デリバリーをサポートするコネクタ・フレームワークに、JMS ソース・キューを指定します。詳細については、26 ページの『ContainerManagedEvents』を参照してください。

デフォルト値は CONNECTOR/SOURCEQUEUE です。

## SynchronousRequestQueue

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルトは CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE です。

## SynchronousResponseQueue

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルトは CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE です。

## SynchronousRequestTimeout

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルト値は 0 です。

## WireFormat

トランスポートのメッセージ・フォーマットです。

- RepositoryDirectory がローカル・ディレクトリーの場合は、設定は CwXML になります。
- RepositoryDirectory の値が <REMOTE> の場合は、設定は CwB0 になります。

## WsifSynchronousRequest Timeout

WAS 統合ブローカーでのみ使用されます。

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

デフォルト値は 0 です。

## XMLNameSpaceFormat

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

XML 形式のビジネス・オブジェクト定義でユーザーがショート・ネーム・スペースとロング・ネーム・スペースを指定できる強力なプロパティです。

デフォルト値は `short` です。

---

## 付録 B. Connector Configurator

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

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

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

**注:**

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

この付録では、次のトピックについて説明します。

- 37 ページの『Connector Configurator の概要』
- 38 ページの『Connector Configurator の始動』
- 39 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』
- 42 ページの『新しい構成ファイルを作成』
- 45 ページの『構成ファイル・プロパティの設定』
- 53 ページの『グローバル化環境における Connector Configurator の使用』

---

### Connector Configurator の概要

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

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (これらをまとめて WebSphere Message Brokers (WMQI) と呼びます。)
- WebSphere Application Server (WAS)

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

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

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

Connector Configurator の実行モードと使用する構成ファイルのタイプは、実行する統合ブローカーによって異なります。例えば、使用している統合ブローカーが WMQI の場合、Connector Configurator を System Manager から実行するのではなく、直接実行します (38 ページの『スタンドアロン・モードでの Configurator の実行』を参照)。

コネクタ構成プロパティには、標準の構成プロパティ (すべてのコネクタにもつプロパティ) と、コネクタ固有のプロパティ (特定のアプリケーションまたはテクノロジーのためにコネクタに必要なプロパティ) 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、39 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

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

---

## Connector Configurator の始動

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

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

### スタンドアロン・モードでの Configurator の実行

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

これを行うには、以下のステップを実行します。

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

- 「システム接続統合ブローカー (System Connectivity 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、WAS のいずれかを選択します。

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方法が便利です (44 ページの『構成ファイルの完成』を参照)。

---

## System Manager からの Configurator の実行

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

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

1. System Manager を開きます。
2. 「System Manager」ウィンドウで、「統合コンポーネント・ライブラリー」アイコンを展開し、「コネクタ」を強調表示します。
3. System Manager メニュー・バーから、「ツール」>「**Connector Configurator**」をクリックします。「Connector Configurator」ウィンドウが開き、「新規コネクタ」ダイアログ・ボックスが表示されます。
4. 「システム接続統合ブローカー (System Connectivity 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、WAS のいずれかを選択します。

既存の構成ファイルを編集するには、以下のステップを実行します。

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

---

## コネクタ固有のプロパティ・テンプレートの作成

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

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

- テンプレートの新規作成については、39 ページの『新規テンプレートの作成』を参照してください。
- 既存のファイルを使用する場合には、既存のテンプレートを変更し、新しい名前でのこのテンプレートを保管します。

## 新規テンプレートの作成

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

テンプレートは以下のように作成します。

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

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

### 一般特性の指定

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

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

プロパティの一般特性の選択を終えたら、「値」タブをクリックします。

### 値の指定

「値」タブを使用すると、プロパティの最大長、最大複数値、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、以下のステップを実行します。



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

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

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

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

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

## 依存関係の設定

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

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

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

>= (より大か等しい)

<= (より小か等しい)

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

---

## 新しい構成ファイルを作成

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

ブローカーを選択するには、以下のステップを実行します。

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

また、以下の作業も実行できます。

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

## コネクター固有のテンプレートからの構成ファイルの作成

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

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

- **名前**

コネクターの名前を入力します。名前では大文字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ーのファイル名に対応した一意の名前でなければなりません。

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

- **システム接続**

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

- 「コネクタ固有プロパティ・テンプレート」を選択します。

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

使用するテンプレートを選択し、「OK」をクリックします。

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

- コネクタ定義ファイル。  
コネクタ定義ファイルは、特定のコネクタのプロパティと、適用可能なデフォルト値がリストされたテキスト・ファイルです。コネクタの配布パッケージの `¥repository` ディレクトリー内には、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りポジトリー・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。

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

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

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

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

---

## 構成ファイルの完成

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

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

1. 「標準のプロパティ」タブで、BrokerType プロパティの値フィールドを選択します。ドロップダウン・メニューで、値 ICS、WMQI、または WAS を選択します。
2. 選択したブローカーに関連付けられているプロパティが「標準のプロパティ」タブに表示されます。ここでファイルを保管するか、または 47 ページの

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

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

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

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

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

---

## 構成ファイル・プロパティの設定

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

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

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

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

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

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

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

標準プロパティとコネクタ固有プロパティの違いは、以下のとおりです。

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

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

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

## 標準コネクター・プロパティの設定

標準のプロパティの値を変更するには、以下のステップを実行します。

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

## アプリケーション固有の構成プロパティの設定

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

1. グリッドの左上端の部分で右マウス・ボタンをクリックします。ポップアップ・メニュー・バーが表示されます。プロパティを追加するときは「追加」をクリックします。子プロパティを追加するには、親の行番号を右マウス・ボタンでクリックし、「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. 46 ページの『標準コネクタ・プロパティの設定』の説明に従い、変更内容を保管するかまたは破棄するかを選択します。

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

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

### コネクタ・プロパティの暗号化

「プロパティを編集」ウィンドウの「暗号化」チェック・ボックスにチェックマークを付けると、アプリケーション固有のプロパティを暗号化することができます。値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、暗号化解除された値が表示されます。

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

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

### 更新メソッド

付録『コネクタの標準構成プロパティ』の 20 ページの『プロパティ値の設定と更新』にある更新メソッドの説明を参照してください。

## サポートされるビジネス・オブジェクト定義の指定

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

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

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

## ご使用のブローカーが ICS の場合

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

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

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

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

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

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

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



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

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

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

ほとんどのコネクターの場合、選択可能な項目は「最大限の努力」のみです。

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

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

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

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

WebSphere Business Integration Message Broker 5.0 を使用している場合、「メッセージ・セット ID」はオプションのフィールドです。値を指定するときは、固有の値を指定しなくてもかまいません。ただし、WebSphere MQ Integrator 2.1 または WebSphere MQ Integrator Broker 2.1 を使用している場合は、固有の ID を指定する必要があります。

### ご使用のブローカーが WAS の場合

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

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

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

## 関連付けられているマップ (ICS のみ)

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

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

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

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

「**関連付けられたマップ**」タブには以下のフィールドが表示されます。

- **ビジネス・オブジェクト名**

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

- **関連付けられたマップ**

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

- **明示的**

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

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

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

以下のステップを実行して、マップを明示的にバインドします。

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

## リソース (ICS)

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

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

## メッセージング (ICS)

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

## トレース/ログ・ファイル値の設定

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

ログとトレースの値を変更するには、以下のステップを実行します。

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

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

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

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

## データ・ハンドラー

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

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

---

## 構成ファイルの保管

コネクターの構成が完了したら、コネクター構成ファイルを保管します。Connector Configurator では、構成中に選択したブローカー・モードで構成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 Integration Component Library に `*.con` 拡張子付きファイルとして保管します。
- 指定したディレクトリーに保管します。
- スタンドアロン・モードで、ディレクトリー・フォルダーに `*.cfg` 拡張子付きファイルとして保管します。

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

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

---

## 構成ファイルの変更

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

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

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

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

---

## 構成の完了

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

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

---

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

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

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

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

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

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

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

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

---

## 特記事項

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

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

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Burlingame Laboratory Director  
IBM Burlingame Laboratory  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

---

## プログラミング・インターフェース情報

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

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  
Domino  
Lotus  
Lotus Notes  
MQIntegrator  
MQSeries  
Tivoli  
WebSphere

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

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

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



WebSphere Business Integration Adapter Framework V2.4.0