

**WebSphere Business Integration Server
Express Plus**



Adapter for iSeries ユーザーズ・ガイド

バージョン 4.3.1

**WebSphere Business Integration Server
Express Plus**



Adapter for iSeries ユーザーズ・ガイド

バージョン 4.3.1

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、IBM WebSphere Business Integration Server Express バージョン 4.3.1、IBM WebSphere Business Integration Server Express Plus バージョン 4.3.1、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： WebSphere Business Integration Server Express Plus
Adapter for iSeries User Guide
Version 4.3.1

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 2004.8

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 2004. 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
本リリースの新機能	vii
バージョン 4.3.1	vii
バージョン 4.3	vii
第 1 章 概要	1
OS/400 システムの概要	1
アダプターの動作方法	4
ロケール依存データの処理	5
第 2 章 iSeries アダプターのインストールと構成	7
前提条件	7
iSeries アダプターと関連ファイルのインストール	8
インストール済みファイルの構造	8
コネクタの構成	9
複数のコネクタ・インスタンスの作成	10
コネクタの始動	13
コネクタの停止	15
第 3 章 コネクタ用ビジネス・オブジェクトの開発	17
コネクタ・ビジネス・オブジェクトの処理	17
コネクタ・ビジネス・オブジェクトの構造	17
ビジネス・オブジェクト Connection の XSD スキーマ	18
ビジネス・オブジェクト PAS400rpg2 の XSD スキーマ	20
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	22
付録 A. コネクタの標準構成プロパティ	25
標準コネクタ・プロパティの構成	25
標準プロパティの要約	26
標準構成プロパティ	29
付録 B. Connector Configurator Express	39
Connector Configurator Express の概要	39
Connector Configurator Express の始動	40
System Manager からの Configurator Express の実行	40
コネクタ固有のプロパティ・テンプレートの作成	41
新しい構成ファイルの作成	43
既存ファイルの使用	44
構成ファイルの完成	46
構成ファイル・プロパティの設定	46
構成ファイルの保管	52
構成の完了	53
グローバル化環境における Connector Configurator Express の使用	53
特記事項	55
特記事項	55

本書について

IBM[®] WebSphere Business Integration Server Express Plus は、InterChange Server Express、関連する Toolset Express、CollaborationFoundation、およびソフトウェア統合アダプターのセットで構成されています。Toolset Express に含まれるツールは、ビジネス・オブジェクトの作成、変更、および管理に役立ちます。プリパッケージされている各種アダプターは、お客様の複数アプリケーションにまたがるビジネス・プロセスに応じて、いずれかを選べるようになっています。標準的な処理のテンプレートである CollaborationFoundation は、カスタマイズされたプロセスを簡単に作成できるようにするためのものです。

本書では、Adapter for iSeries の構成、ビジネス・オブジェクトの開発、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
- iSeries 統合ファイル・システムの基本

関連文書

本書の対象製品の一連の関連文書には、WebSphere Business Integration Server Express のどのインストールにも共通する機能とコンポーネントの解説のほか、特定のコンポーネントに関する参考資料が含まれています。

関連文書は、<http://www.ibm.com/websphere/wbiserverexpress/infocenter> でダウンロード、インストール、および表示することができます。

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。これらの技術情報や速報は、WebSphere Business Integration のサポート Web サイト (<http://www.ibm.com/software/integration/websphere/support/>) で参照できます。適切なコンポーネント領域を選択し、「Technotes (技術情報)」セクションと「Flashes (速報)」セクションを参照してください。

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: <code><server_name><connector_name>tmp.log</code>)
/、¥	本書では、Windows のディレクトリー・パスの表記規則として円記号 (¥) を使用します。OS/400 および Linux では、ディレクトリー・パスにスラッシュ (/) を使用します。すべての WebSphere Business Integration Server Express システム製品のパス名は、ご使用のシステムにおいてこの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。
ProductDir	IBM WebSphere Business Integration Server Express for Adapters 製品のインストール先ディレクトリーを表します。各プラットフォームのデフォルトは、以下のとおりです。 Windows: IBM¥WebSphereServer OS/400: /QIBM/ProdData/WBIServer43/product Linux: /home/\${username}/IBM/WebSphereServer

本リリースの新機能

バージョン 4.3.1

本リリースでは、以下のオペレーティング・システムのサポートが追加されました。

- IBM OS/400 V5R2、V5R3
- Red Hat Enterprise Linux AS 3.0 (Update 1 を適用)
- SuSE Linux Enterprise Server 8.1
- Microsoft Windows 2003 (実動モードでの InterChange Server Express およびアダプターのみ)

このリリースには、Integrated Language Environment (ILE) プログラムおよび Report Program Generator (RPG) プログラムのサポートも追加されています。

バージョン 4.3

本書の最初のリリースです。

第 1 章 概要

この章では、WebSphere Business Integration Server Express および Express Plus で使用する iSeries コンポーネントのアダプターについて説明します。このアダプターは、IBM Toolbox for Java (JavaTM クラスのセット) を使用し、OS/400 オペレーティング・システムで稼働する iSeries 上で **Integrated Language Environment (ILE)** プログラムを実行する機能を提供します。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 Business Integration Server Express システム管理ガイド*」または「*IBM WebSphere Business Integration Server Express システム・インプリメンテーション・ガイド*」を参照してください。

注: WebSphere Business Integration Server Express のアダプターは、いずれも、InterChange Server Express を統合ブローカーとして動作します。

本章の内容は、次のとおりです。

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

OS/400 システムの概要

IBM eServer iSeries システムは、ビジネスで複数の操作環境を同時に実行できる、高度に統合された、信頼性の高いサーバー・プラットフォームです。保全性とセキュリティに優れているため、多くの重要なアプリケーションで使用することができます。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。

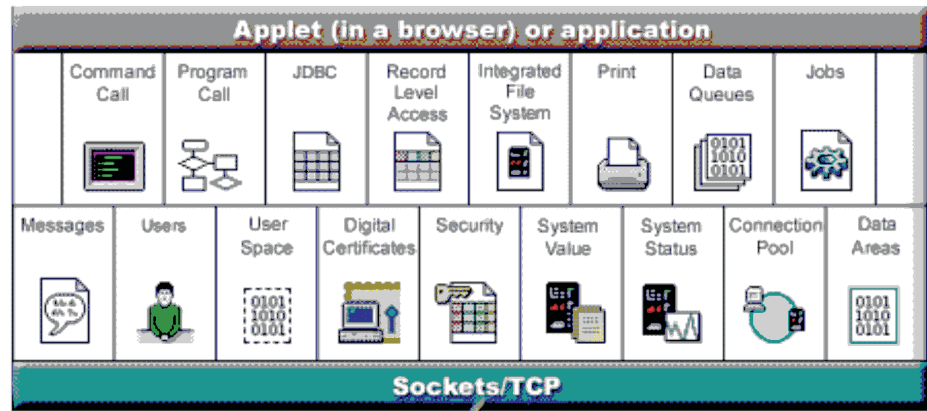
ホスト・サーバーは、3 ページの図 1 に示すように、アプリケーションを実行している装置 (クライアント PC など) からの要求を処理して、文書の印刷などのタスクを実行できるようにします。OS/400 コンピューターは、ファイル、データベース、アプリケーション、メール、プリント、マルチメディア、FAX、およびワイヤレス通信などの多くのタスクを一度に実行できる全機能サーバーです。タスク・サーバーは、システム上でそれぞれ個別のジョブとして実行されます。また、サーバー・ジョブは、それぞれソケット接続を介してデータ・ストリームを送受信します。

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、OS/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・クラスがサインオン情報の管理やソケット接続の作成や保守、およびデータの送受信を行う一方で、コマンド呼び出しクラスが OS/400 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 ILE プログラムを呼び出します。OS/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

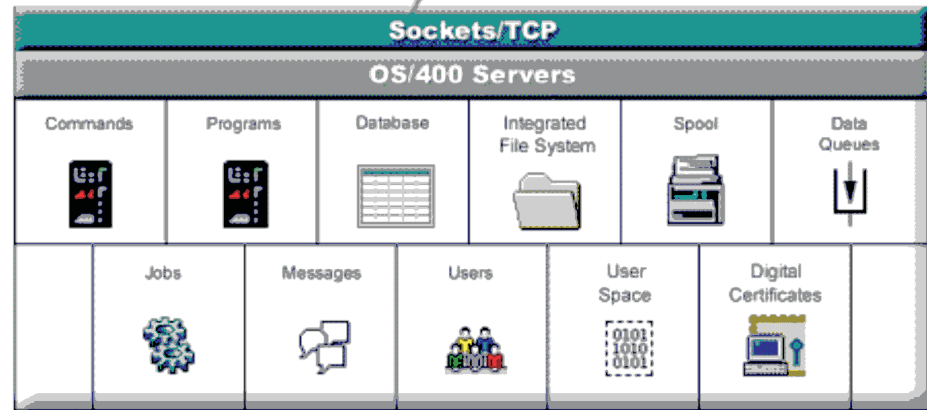


図1. OS/400 クライアント - サーバー・アーキテクチャーの概要

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、iSeries アダプターは、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を使用します。このサーバーは、OS/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。

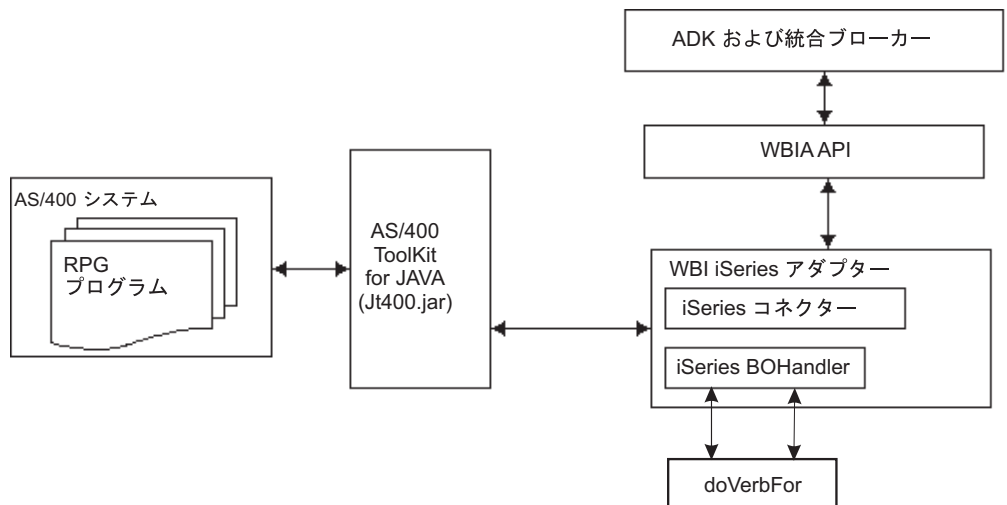


図2. Adapter for iSeries を使用した場合の接続のしくみ

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、InterChange Server Express からビジネス・オブジェクト要求を受信すると、ILE 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 OS/400 システムとの接続を確立して、ILE 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、OS/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、InterChange Server Express 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、OS/400 システムで実行される ILE プログラムが呼び出されます。

注: 基本的には OS/400 システム上のプログラムを実行するための呼び出しで、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

ロケール依存データの処理

アダプターは国際化され、2 バイト文字セットをサポートし、特定の言語でメッセージ・テキストを配信できるようになっています。アダプターは、ある文字コード・セット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) 内部の Java ランタイム環境では、Unicode 文字コード・セットでデータを表現します。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含む) の文字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。そのため、WebSphere Business Integration Server Express システム・コンポーネント間でデータを転送するときは、ほとんどの場合文字変換は必要ありません。

エラー・メッセージと通知メッセージを適切な言語で適切な国と地域に合わせて記録するには、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、25 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。

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。

- 『前提条件』
- 8 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』
- 10 ページの『複数のコネクター・インスタンスの作成』
- 13 ページの『コネクターの始動』
- 15 ページの『コネクターの停止』

前提条件

コネクターを使用するには、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/series/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。jt400.jar は現在、国際化対応をサポートしていません。

- WBIA.jar ファイル
 - CrossWorlds.jar ファイル
 - BIA_iSeries.jar ファイル
2. iSeries アダプターは、次のいずれかの OS/400 バージョンと接続するように設計されています。
- バージョン 5 リリース 1 から 3
 - バージョン 4 リリース 1 から 3
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。次のリンクから適切な PTF を入手する必要があります。

<http://www-1.ibm.com/servers/eserver/series/toolbox/hostservicepackdetail.htm>

iSeries アダプターと関連ファイルのインストール

アダプターのインストールについては、次のサイトの WebSphere Business Integration Server Express InfoCenter にある 「WebSphere Business Integration Server Express インストール・ガイド」を参照してください。

<http://www.ibm.com/websphere/wbiserverexpress/infocenter>

インストール済みファイルの構造

以下のサブセクションでは、Windows、OS/400、および Linux プラットフォーム上でのアダプターのインストール済みファイルの構造について説明します。

表 1 ではコネクターが使用する Windows ファイル構造を説明し、9 ページの表 2 では OS/400 ファイル構造を、9 ページの表 3 では Linux ファイル構造をそれぞれ説明します。

コネクター・コンポーネントのインストールについて詳しくは、次のガイドを参照してください。

- 「クイック・スタート・ガイド」
- 「IBM WebSphere Business Integration Server Express and Express Plus インストール・ガイド」

インストール済みファイルの構造 (Windows の場合)

表 1. コネクター用としてインストールされた Windows ファイル構造

%ProductDir% のサブディレクトリー	説明
connectors¥iSeries	コネクターの BIA_iSeries.jar、start_iSeries.bat および start_iSeries_service.bat が格納されます。
connectors¥messages	BIA_iSeriesAdapter.txt ファイルが格納されています。
repository¥iSeries	BIA_CN_iSeries.txt ファイルが格納されています。
¥lib	WBIA.jar ファイルが格納されています。
¥bin	CWConnEnv.bat ファイルが格納されています。

Windows の場合、インストーラーによって、「プログラム」>「IBM WebSphere Business Integration Express」>「アダプター」>「コネクター」を選択することにより表示可能なコネクター・ファイルのアイコンが追加されます。コネクターをすばやく始動するには、このファイルへのショートカットをデスクトップに作成してください。

インストール済みファイルの構造 (OS/400 の場合)

表 2. コネクタ用としてインストールされた OS/400 ファイルのファイル構造

%ProductDir% のサブディレクトリー	説明
connectors/iSeries	コネクタの BIA_iSeries.jar と start_iSeries.sh ファイルが格納されます。
connectors/messages	BIA_iSeriesAdapter.txt ファイルが格納されています。
repository/iSeries	BIA_CN_iSeries.txt ファイルが格納されています。
/lib	WBIA.jar ファイルが格納されています。
/bin	CWConnEnv.sh ファイルが格納されています。

OS/400 の場合、コネクタをすばやく始動するには、コンソール機能を使用してください。コンソールの詳細については、コンソールに付属のオンライン・ヘルプを参照してください。

インストール済みファイルの構造 (Linux の場合)

表 3. コネクタ用としてインストールされた Linux ファイル構造

%ProductDir% のサブディレクトリー	説明
connectors/iSeries	コネクタの BIA_iSeries.jar と start_iSeries.sh.files が格納されます。
connectors/messages	BIA_iSeriesAdapter.txt ファイルが格納されています。
repository/iSeries	BIA_CN_iSeries.txt ファイルが格納されています。
/lib	WBIA.jar ファイルが格納されています。
/bin	CWConnEnv.sh ファイルが格納されています。

Linux の場合、「connector_manager」コマンドを使用してコネクタを始動する必要があります。

コネクタの構成

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・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固有のプロパティも使用します。

標準コネクタ・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、25 ページの『付録 A. コネクタの標準構成プロパティ』を参照してください。

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ー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ー

コネクター固有の構成プロパティーは、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ーを使用すれば、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4 に、コネクターのコネクター固有の構成プロパティーを示します。各プロパティーの説明については、以降のセクションを参照してください。

表4. コネクター固有のプロパティー

名前	指定可能な値	デフォルト値	必須
ExecutionTimeout		なし	はい
UseDefaults	デフォルト値	なし	はい
PollQuantity	1 より大きい整数	1	いいえ

ExecutionTimeout

コネクターが iSeries システムの pgm プログラムを呼び出す場合のタイムアウトです。

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 プロパティーが 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。

PollQuantity

コネクターがポーリングする項目数を指定します。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ーがある場合、標準プロパティーの値は、このコネクター固有のプロパティーの設定値によりオーバーライドされます。

複数のコネクター・インスタンスの作成

このアダプター（または、WebSphere Business Integration Server Express や Express Plus と共に提供される任意のアダプター）に追加のインスタンスを作成すると、配置できるアダプターの合計数を制限するライセンス機能は、そのアダプターのインスタンスを個別のアダプターとしてカウントされます。

以下に示すステップを実行することによって、コネクタの複数のインスタンスを作成して実行するように、ご使用のシステムを設定することができます。次のようにする必要があります。

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

新規ディレクトリの作成

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

- Windows プラットフォームの場合:

```
ProductDir¥connectors¥connectorInstance
```

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

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

```
ProductDir¥repository¥connectorInstance
```

- OS/400 プラットフォームの場合:

```
/QIBM/UserData/WBIServer43/WebSphereICSName/connectors/connectorInstance
```

ここで、`connectorInstance` はコネクタ・インスタンスを固有に識別し、`WebSphereICSName` はコネクタの実行に使用する `InterChange Server Express` インスタンスの名前です。`QWBIDFT` がデフォルトの名前です。

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

```
/QIBM/UserData/WBIServer43/WebSphereICSName/repository/connectorInstance
```

ここで、`WebSphereICSName` はコネクタの実行に使用する `InterChange Server Express` インスタンスの名前です。

- Linux プラットフォームの場合:

```
ProductDir/connectors/connectorInstance
```

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

```
ProductDir/repository/connectorInstance
```

connector_manager のパラメーターとして ICS サーバー名を指定できます。例えば、「connector_manager -start connName WebSphereICSName [-cConfigFile]」と指定します。

ビジネス・オブジェクト定義の作成

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

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

- Windows:

```
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。

- OS/400:

```
/QIBM/UserData/WBIServer43/WebSphereICSName/repository  
/initialConnectorInstance
```

WebSphereICSName は、コネクタを実行する InterChange Server Express サーバーのインスタンスの名前です。

作成した追加ファイルは、

```
/QIBM/UserData/WBIServer43/WebSphereICSName/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。

- Linux:

```
/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。

コネクタ定義の作成

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

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

始動スクリプトの作成

始動スクリプトを作成するには、以下のステップを実行します。

1. 初期コネクターの始動スクリプトをコピーし、コネクタ・ディレクトリーの名前を含む名前を付けます。

`dirname`

(Linux の場合のみ) 始動スクリプト `CONJAR` を

「`CONJAR=${CONDIR}/BIA_${CONNAME}.jar`」から

「`CONJAR=${CONDIR}/BIA_iSeries.jar`」に変更する必要があります。

2. この始動スクリプトを、11 ページの『新規ディレクトリーの作成』で作成したコネクタ・ディレクトリーに格納します。
3. (Windows の場合のみ) 始動スクリプトのショートカットを作成します。
4. (Windows の場合のみ) 初期コネクターのショートカット・テキストをコピーし、新規コネクタ・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。
5. (OS/400 の場合のみ) 次の情報を使用して、コネクタのジョブ記述を作成します。

```
CRTDUPOBJ OBJ(QWBII SRSC) FROMLIB(QWBISVR43) OBJTYPE(*JOB) TOLIB(QWBISVR43)
NEWOBJ(newiSeriesname)
```

`newiSeriesname` は、新規 E メール・コネクタのジョブ説明に使用する 10 文字の名前です。

6. (OS/400 の場合のみ) 新規コネクタをコンソールに追加します。コンソールの詳細については、コンソールに付属のオンライン・ヘルプを参照してください。

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。

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・ディレクトリーに存在していなければなりません。例えば、Windows の場合は以下を使用します。

```
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 5 に示すように、オペレーティング・システム・プラットフォームで定められています。

表 5. 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
Windows	<code>start_connName.bat</code>
Linux	<code>connector_manager -start connName [-cConfigFile]</code> により、環境変数が設定され、 <code>start_connName.sh</code> 始動スクリプトが自動的に開始されます。始動スクリプトを手動で実行する必要はありません。
OS/400	<code>start_connName.sh</code>

コマンド行の始動オプションなどのコネクターの始動方法の詳細については、「システム管理ガイド」を参照してください。

始動スクリプトの起動 (Windows の場合)

Windows プラットフォームでは、以下の方法でコネクターの始動スクリプトを起動できます。

- System Monitor から

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。

- 「スタート」メニューから

– 「プログラム」 > 「IBM WebSphere Business Integration Express」 > 「アダプター」 > 「コネクター」 > 「ご使用のコネクター名」を選択します。

デフォルトでは、プログラム名は「IBM WebSphere Business Integration Express」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、ご使用のコネクターへのデスクトップ・ショートカットを作成することもできます。

– コネクターは、Windows システムの Windows サービスとして始動するように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、コネクターが始動します。

- コマンド行から

```
start_connName connName WebSphereICSName [-cconfigFile ]
```

ここで、connName はコネクターの名前であり、WebSphereICSName は InterChange Server Express インスタンスの名前です。

デフォルトでは、InterChange Server Express インスタンスの名前は WebSphereICS です。

始動スクリプトの起動 (OS/400 の場合)

OS/400 プラットフォームでは、以下の方法でコネクターの始動スクリプトを起動できます。

- Windows から

WebSphere Business Integration Server Express Console がインストールされているマシンから、「プログラム」 > 「IBM WebSphere Business Integration Console」 > 「コンソール」を選択します。次に、OS/400 システム名または IP アドレスと、*JOBCTL 特殊権限を持つユーザー・プロファイルおよびパスワードを指定します。アダプターのリストから connName アダプターを選択し、「アダプターを始動」ボタンを選択します。

- OS/400 コマンド行から

バッチ・モード: CL Command および QSH を実行し、QSHHELL 環境から QIBM/ProdData/WBIServer43/bin/submit_adapter.sh connName WebSphereICSName pathToConnNameStartScript jobDescriptionName を実行する必要があります。ここで、connName はコネクター名、WebSphereICSName は InterChange Server Express のサーバー名 (デフォルトは QWBIDFT)、pathToConnNameStartScript は

コネクタ始動スクリプトの絶対パス、*jobDescriptionName* は QWBISVR43 ライブラリーで使用するジョブ説明の名前です。

対話モード: CL Command および QSH を実行し、QShell 環境から
/QIBM/UserData/WBIServer43/WebSphereICSName/connectors
/connName/start_connName.sh connName WebSphereICSName[-cConfigFile] を実行する必要があります。ここで、connName はコネクタ名、WebSphereICSName は Interchange Server Express インスタンスの名前です。

注: TCP/IP サーバーを始動するには、QShell 環境でスクリプト
/QIBM/ProdData/WBIServer43/bin/add_autostart_adapter.sh connName
WebSphereICSName pathToConnNameStartScript jobDescriptionName を使用します。ここで、connName はコネクタ名、WebSphereICSName は InterChange Server Express のサーバー名 (デフォルトは QWBIDFT)、pathToConnNameStartScript はコネクタ始動スクリプトの絶対パス、jobDescriptionName は QWBISVR43 ライブラリーで使用するジョブ説明の名前です。

- System Monitor から

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。

始動スクリプトの起動 (Linux の場合)

Linux プラットフォームでは、以下の方法でコネクタの始動スクリプトを起動できます。

```
connector_manager -start connName WebSphereICSName [-cConfigFile]
```

ここで、connName はコネクタの名前であり、WebSphereICSName は InterChange Server Express インスタンスの名前です。デフォルトでは、InterChange Server Express インスタンスの名前は WebSphereICS です。

コネクタの停止

コネクタを停止する方法は、コネクタが始動された方法によって異なります。

コネクタの停止 (Windows から)

Windows プラットフォームでは、以下の方法でコネクタを停止できます。

- System Monitor から

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。

- 「コネクタ」ウィンドウをアクティブにして、「q」と入力して Enter を押しします。
- コネクタが Windows のサービスとして始動された場合は、コントロール・パネル (「コントロール パネル」>「管理ツール」>「サービス」>「CWConnectorWBliSeriesAdapter」) を使用してコネクタを停止できます。

コネクターの停止 (OS/400 から)

OS/400 プラットフォームでは、以下の方法でコネクターを停止できます。

- コンソールまたはコマンド行から

コンソールを使用して、または OS/400 コマンド入力から QSHELL で「submit_adapter.sh」スクリプトを使用してコネクターを始動した場合は、CL コマンド `WRKACTJOB SBS(QWBISVR43)` を使用して Server Express 製品に対するジョブを表示します。リストをスクロールして、コネクターのジョブ記述に一致するジョブ名を持つジョブを探し出します。例えば、e-Mail Connector では、jobname は QWBIEMAILC です。

このジョブに対してオプション 4 を選択し、F4 を押して ENDJOB コマンドのプロンプトを取得します。次に、オプション・パラメーターとして *IMMED を指定し、Enter を押します。

- start_connName.sh の使用

OS/400 コマンド・エントリーから QSHELL の start_connName.sh スクリプトを使用してコネクターを始動した場合、F3 キーを押すだけで QSHELL 環境を終了し、アダプターを終了することができます。

- System Monitor から

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、または削除を実行できます。

コネクターの停止 (Linux から)

Linux システムでは、コネクターはバックグラウンドで実行されるので、個別のウィンドウはありません。代わりに、以下のコマンドを実行してコネクターを停止します。

```
connector_manager connName -stop serverName
```

ここで、connName はコネクターの名前で、serverName は InterChange Server Express のインスタンスです。

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、InterChange Server Express と OS/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、InterChange Server Express 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、OS/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。

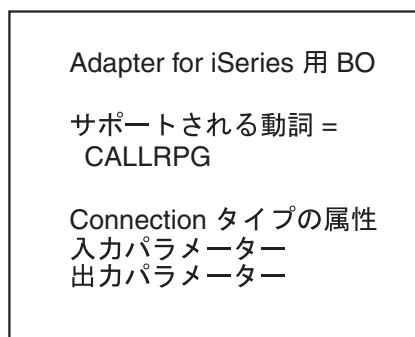


図 3. iSeries の親ビジネス・オブジェクト

また、タイプが Connection の子属性もあります。これには、OS/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

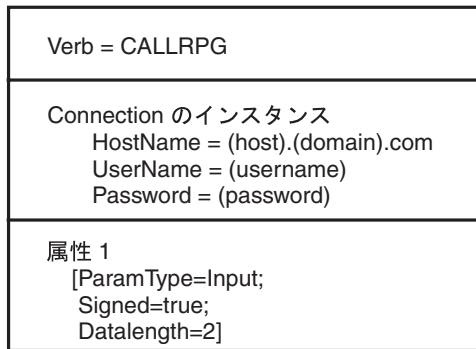


図5. ビジネス・オブジェクトの例

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 Express 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 6. 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 6. 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、プロパティは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。

OS/400 用の Toolbox からのデータ変換

OS/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、OS/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 7. 変換されるデータ型とデータ変換クラス

OS/400 のデータ型	WebSphere Business Integration Express のデータ型	データ変換クラス
OS/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	OS400Bin2
OS/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	OS400Bin4
OS/400 形式の符号付き 2 バイト浮動小数点数	Float	OS400Float4
OS/400 形式の符号付き 4 バイト浮動小数点数	Double	OS400Float8
OS/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	OS400UnsignedBin2

表 7. 変換されるデータ型とデータ変換クラス (続き)

OS/400 のデータ型	WebSphere Business Integration Express のデータ型	データ変換クラス
OS/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	OS400UnsignedBin4
OS/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	OS400PackedDecimal
OS/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	OS400ZonedDecimal
文字データ	String (MaxLength に文字データの最大長を指定します)	OS400Text
日付データ	String (MaxLength に日付データの最大長を指定します)	OS400Text

付録 A. コネクタの標準構成プロパティ

この付録では、WebSphere InterChange Server Express で動作する、WebSphere Business Integration Server Express のアダプターに含まれるコネクタ・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。

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 Express から統合ブローカーを選択すると、ご使用のアダプターに対し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。

標準コネクタ・プロパティの構成

アダプター・コネクタには 2 つのタイプの構成プロパティがあります。

- 標準構成プロパティ
- コネクタ固有のプロパティ

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

Connector Configurator Express の使用

コネクタ・プロパティの構成は Connector Configurator Express から行います。Connector Configurator Express には、System Manager からアクセスします。Connector Configurator Express の使用方法の詳細については、付録の『Connector Configurator Express』を参照してください。

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
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。
- **コンポーネント再始動**
System Manager でコネクター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**
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 Express」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 8 は、標準コネクター構成プロパティの早見表です。コネクター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。

コネクターを実行する前に、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 8. 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 IDL
AgentTraceLevel	0 から 5	0	動的	
ApplicationName	アプリケーション名	コネクター・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	ICS		
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

表 8. 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>
ControllerTraceLevel	0 から 5	0	動的	Repository Directory は <REMOTE>
DeliveryQueue		CONNECTORNAME/DELIVERYQUEUE	コンポーネント再始動	JMS トランスポートのみ
DeliveryTransport	IDL または JMS	IDL	コンポーネント再始動	
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。
EnableOidForFlowMonitoring	true または false	false	コンポーネント再始動	
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	crossworlds.queue.manager	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>
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	

表 8. 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
LogAtInterchangeEnd	true または false	false	コンポーネント再始動	
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE>
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 でなければならない。
OADAutoRestartAgent	true または false	false	動的	Repository Directory は <REMOTE>
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE>
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE>
PollEndTime	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	
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	
RequestQueue	メタデータ・リポジトリの場所		エージェント再始動	<REMOTE> に設定する
RequestQueue	有効な JMS キュー名	CONNECTORNAME/REQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
ResponseQueue	有効な JMS キュー名	CONNECTORNAME/RESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
RestartRetryCount	0 から 99	3	動的	
RestartRetryCount	適切な正数 (単位: 分) 1 から 2147483547	1	動的	

表 8. 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
SourceQueue	有効な JMS キュー名	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	有効な JMS キュー名	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	有効な JMS キュー名	CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
WireFormat	CwBO	CwBO	エージェント再始動	

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

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

コネクター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。コネクターを実行する前に、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を指定します。ICS を指定する必要があります。

CharacterEncoding

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

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

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

ConcurrentEventTriggeredFlows

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。

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。

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 Express 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。

ControllerStoreAndForwardMode

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

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 WebSphere InterChange Server Express へビジネス・オブジェクトが送信される時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、IDL (CORBA IIOP) または JMS (Java Messaging Service) です。デフォルトは IDL です。

DeliveryTransport プロパティに指定されている値が 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。

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すると、jms.MessageBrokerName、jms.FactoryClassName、jms.Password、jms.UserName などの追加の JMS プロパティが Connector Configurator Express 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: WebSphere InterChange Server Express で動作しているコネクタで 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があります。

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。

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。

EnableOidForFlowMonitoring

このプロパティを true に設定すると、アダプター・フレームワークは、フロー・モニターを使用できるようにするため、着信 **ObjectEventId** を外部キーとしてマークします。

デフォルト値は false です。

FaultQueue

コネクターでメッセージを処理中にエラーが発生すると、コネクターは、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)。

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)。

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)。

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、このコネクター・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として選択するときは (DeliveryTransport を参照)、このコネクター・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。

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・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、本書の『Connector Configurator Express』の使用方法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

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、フロー制御で使用されます。

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

MessageFileName

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

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

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

MonitorQueue

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

デフォルト値は CONNECTORNAME/MONITORQUEUE です。

OADAutoRestartAgent

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)」を参照してください。

デフォルト値は false です。

OADMaxNumRetry

異常シャットダウンの後で MQ により起動される OAD がコネクター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

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。コネクター・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、コネクター・コントローラーはコネクター・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。このプロパティが使用されるかどうかを特定のコネクターについて判断するには、該当するアダプター・ガイドのインストールと構成についての章を参照してください。

PollQuantity

コネクタ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。

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

WebSphere InterChange Server Express からコネクタへビジネス・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 `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。

この値は `<REMOTE>` に設定する必要があります。これは、コネクタが InterChange Server Express リポジトリからこの情報を取得するためです。

ResponseQueue

`DeliveryTransport` が JMS の場合のみ適用されます。

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。WebSphere InterChange Server Express は、要求を送信した後、JMS 応答キューで応答メッセージを待機します。

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、31 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。設定値は CwB0 です。

付録 B. Connector Configurator Express

この付録では、Connector Configurator Express 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。

この付録では、次のトピックについて説明します。

- 39 ページの『Connector Configurator Express の概要』
- 40 ページの『Connector Configurator Express の始動』
- 41 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』
- 43 ページの『新しい構成ファイルの作成』
- 46 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 Express の使用』

Connector Configurator Express の概要

Connector Configurator Express では、WebSphere InterChange Server Express 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。

Connector Configurator Express 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、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、ロギングとトレース、およびデータ・ハンドラーに関するパラメーターを指定する必要があります。

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持つプロパティ) 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、すべてのコネクタで使用されるので、新規に定義する必要はありません。構成ファイルを作成すると、Connector Configurator Express によって標準プロパティがそのファイルに挿入されます。ただし、Connector Configurator Express 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 Express の「標準のプロパティ」ウィンドウには、現在ご使用の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、41 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 Express は、Windows 環境でのみ実行できます。別の環境でコネクタを実行する場合には、Windows で Connector Configurator Express を使用して構成ファイルを変更し、このファイルを別の環境へコピーしてください。

Connector Configurator Express の始動

Connector Configurator Express は、以下の 2 種類のモードで始動し、実行することができます。

- 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 Express の実行

Connector Configurator Express をブローカーと連携させずに別個に実行して、コネクタ構成ファイルを編集することができます。

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 Business Integration Server Express」>「Toolset Express」>「開発」>「Connector Configurator Express」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。

Connector Configurator Express 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 (46 ページの『構成ファイルの完成』を参照)。

System Manager からの Configurator Express の実行

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

Connector Configurator Express 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 Express」をクリックします。「Connector Configurator Express」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。

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。

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、41 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。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。入力した情報が、Connector Configurator Express によって、Connector Configurator Express 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 XML 文書として保管されます。

新しい構成ファイルの作成

コネクター構成ファイルを作成するには、コネクター固有のテンプレートから作成するか、既存の構成ファイルを変更します。

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。

- 以下のフィールドを含む「**新規コネクタ**」ダイアログ・ボックス表示されません。

- **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 Express 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**

デフォルトのブローカーは ICS です。この値は変更できません。

- **コネクタ固有プロパティ・テンプレートを選択**

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。

- 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。

- 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 Express 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する必要があります。

- この章で後述する手順に従って、「Connector Configurator Express」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

既存ファイルを使用してコネクタを構成するには、Connector Configurator Express でそのファイルを開き、構成を修正してから、構成ファイル (`*.cfg`) として保管する必要があります。

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の `¥repository` ディレクトリ内には、このようなファイルが格納されていることがあります (通常、このファイルの拡張子は `.txt` です。例えば、XML コネクタの場合は `CN_XML.txt` です)。
- InterChange Server Express リポジトリ・ファイル
以前にコネクタの InterChange Server Express インプリメンテーションの際に使用された定義が、そのコネクタの構成に使用されたりリポジトリ・ファイルに残されていることがあります。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 Express でそのファイルを開き、構成を修正してから、再度保管する必要があります。

ディレクトリから `*.txt`、`*.cfg` または `*.in` ファイルを開くには、以下のステップを実行します。

1. Connector Configurator Express 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)
- InterChange Server Express リポジトリ (*.in、*.out)

これまでリポジトリ・ファイルを使用してコネクタを構成していた場合は、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 Express 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 Express」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。

Connector Configurator Express では、以下のセクションに記載されているプロパティの値を設定する必要があります。

- 47 ページの『標準コネクタ・プロパティの設定』
- 47 ページの『アプリケーション固有の構成プロパティの設定』
- 48 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』
- 50 ページの『関連付けられたマップ』
- 51 ページの『トレース/ログ・ファイル値の設定』

注: コネクタが JMS メッセージングを使用するものである場合、データをビジネス・オブジェクトに変換するデータ・ハンドラーを構成できるように、追加のカテゴリが表示されることがあります。詳細については、52 ページの『データ・ハンドラー』を参照してください。

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、または既存のコネクタ構成ファイルを開くと、Connector Configurator Express に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。

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のプロパティは、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。

「標準プロパティ」と「コネクタ固有プロパティ」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」フィールドは構成可能です。
- 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。

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 Express 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 Express 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. 『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ー・プロパティ名を変更すると、コネクターに障害が発生する可能性があります。コネクター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。

コネクター・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、各コネクターのアダプター・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『コネクターの標準構成プロパティ』の 25 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。

コネクター・プロパティはほとんどが静的なプロパティであり、それらの更新メソッドはコンポーネント再始動です。変更を有効にするには、変更したコネクター構成ファイルを保管した後、コネクターを再始動する必要があります。

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 Express 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。

サポートされるビジネス・オブジェクトを指定するときには、指定するビジネス・オブジェクトとそのオブジェクトに対応するマップが、システムに存在していなければなりません。ビジネス・オブジェクト定義 (データ・ハンドラー・メタオブジェクトのビジネス・オブジェクト定義を含みます) とマップ定義は、統合コンポーネント・ライブラリー (ICL) プロジェクトに保管されている必要があります。ICL プロジェクトの詳細については、「*User Guide for WebSphere Business Integration Server Express*」を参照してください。

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。詳細につ

いては、本書のビジネス・オブジェクトに関する章と、「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 Express」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 Express」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ず。
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、コネクター定義が変更され、削除されたビジネス・オブジェクトはコネクターのこのインプリメンテーションで使用不可になります。コネクター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、コネクター・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・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 Express」ウィンドウでは、「エージェント・サポート」を選択しても問題ないかどうかの検証は行われません。

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。

関連付けられたマップ

各コネクタは、ビジネス・オブジェクト定義とそれらに関連付けられたマップのうち現在 InterChange Server Express でアクティブであるものを示す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 Express」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。InterChange Server Express は、ブート時、各コネクターのサポートされるビジネス・オブジェクトのそれぞれ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 Express」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 InterChange Server Express 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

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

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

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 Express は、そのファイルに含まれるロギングとトレースに関する値をデフォルト値として使用します。これらの値は、Connector Configurator Express 内で変更できます。

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、優先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。(コネクターが、Connector Configurator Express をインストールした Windows プラットフォームで実行されていない場合は、最初に、システム上のファイルの格納場所にドライブをマップする必要があります。)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。このタブは、アダプターが保証付きイベント・デリバリーを利用するものである場合に使用可能になります。

これらのプロパティーに使用する値については、標準プロパティーに関する付録の『ContainerManagedEvents』の説明を参照してください。

構成ファイルの保管

構成ファイルの作成とそのファイルに含まれるプロパティーの設定が完了したら、使用するコネクターに応じた適切な場所にそのファイルを配置する必要があります。ICL プロジェクトに構成を保管し、保管されたファイルを System Manager から InterChange Server Express へロードしてください。

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

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

System Manager でのプロジェクトの使用方法和、配置の詳細については、「*User Guide for IBM WebSphere Business Integration Server Express*」を参照してください。

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、コネクターが使用する始動ファイルを開き、コネクター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ー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 Express の使用

Connector Configurator Express はグローバル化されており、構成ファイルと統合ブローカーの間での文字変換を処理できます。Connector Configurator Express 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 Express 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、製品ディレクトリーの `¥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>
  </ValidValues>
  <DefaultValue>en_US</DefaultValue>
</Property>
```

特記事項

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）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。

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。

注: 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 および Pentium 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。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

System Manager には、Eclipse Project (<http://www.eclipse.org/>) により開発されたソフトウェアが含まれています。



WebSphere Business Integration ServerExpress V4.3.1 および WebSphere Business Integration Server Express Plus V4.3.1



Printed in Japan