

**IBM WebSphere Business Integration
Adapters**



**Adapter for IndusConnect Framework
ユーザーズ・ガイド**

Adapter バージョン 1.0.1

**IBM WebSphere Business Integration
Adapters**



**Adapter for IndusConnect Framework
ユーザーズ・ガイド**

Adapter バージョン 1.0.1

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

本書は、IBM WebSphere Business Integration Adapter for IndusConnect Framework バージョン 1.0.1、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： IBM WebSphere Business Integration Adapters
Adapter for IndusConnect Framework User Guide
Adapter Version 1.0.1

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
本リリースの新機能	vii
第 1 章 アダプターの概要	1
概要	1
アダプターの概要	1
アダプターのアーキテクチャー	2
第 2 章 アダプターのインストールと構成	9
互換性	9
関連ソフトウェアのインストール	10
インストール・プロセスの概要	10
アダプターのインストール	10
ライブラリーおよびコネクタのインポート	12
新規ユーザー・プロジェクトの作成	13
アダプターの構成	14
ユーザー・プロジェクトの更新	18
サーバーへのプロジェクトの配置	19
ポーリング・パラメーターに関する考慮事項	20
構成および国際化対応	20
System Monitor の始動	22
コネクタの始動	22
WMQIB または WBIMB と共にアダプターを使用する方法	22
Latin 1 および Unicode 言語以外のメッセージ・ファイルの変換	24
第 3 章 ビジネス・オブジェクトについて	25
ビジネス・オブジェクト	25
ビジネス・オブジェクト構造の概要	26
IndusConnect Framework ビジネス・オブジェクトのマッピング	27
第 4 章 トラブルシューティングおよびエラー処理	29
トレース・メッセージ	29
プロファイルからアダプターを組み立てるときの問題	30
トラブルシューティングのヒント	31
エラー・ログを使用した問題の分離	32
エラー処理	33
第 5 章 アダプターの拡張およびカスタマイズ	35
XML 文法	35
付録 A. コネクタの標準構成プロパティー	37
新規プロパティーと削除されたプロパティー	37
標準コネクタ・プロパティーの構成	37
標準プロパティーの要約	39
標準構成プロパティー	42
付録 B. Connector Configurator	55

Connector Configurator の概要	55
Connector Configurator の始動	56
System Manager からの Configurator の実行	57
コネクタ固有のプロパティ・テンプレートの作成	57
新規構成ファイルの作成	60
既存ファイルの使用	61
構成ファイルの完成	62
構成ファイル・プロパティの設定	63
構成ファイルの保管	70
構成ファイルの変更	71
構成の完了	72
グローバル化環境における Connector Configurator の使用	72
付録 C. コネクタ構成の階層	75
付録 D. 特記事項	77
プログラミング・インターフェース情報	78
商標	79
索引	81

本書について

IBM(R) WebSphere(R) 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、およびメインフレーム・システムに統合コネクティビティを提供します。製品セットには、ビジネス・プロセスの統合に向けてコンポーネントをカスタマイズ、作成、および管理するためのツールとテンプレートが含まれています。

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

本書では、コネクタおよびアダプターという用語を同じような意味で使用しています。

対象読者

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

関連文書

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

以下のサイトから資料をインストールするか、オンラインで直接閲覧することができます。

- 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、以下を参照してください。

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

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

表記上の規則

本書では、以下のような規則を使用しています。

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

Adapter for IndusConnect Framework バージョン 1.0.1 に対応した本書の今回のリリースでは、以下の新しい情報が追加されました。

- 9 ページの『第 2 章 アダプターのインストールと構成』のコネクター構成手順では、Connector Configurator を使用して構成タスクを実行する方法についての説明が追加されました。さらに、この章ではソフトウェアの互換性に関するセクションの内容が見直され、サポートされるプラットフォームとソフトウェアのバージョンの変更が反映されました。
- エラー処理および文書エンコードに関連して、新しい構成パラメーターの説明が追加されました。
- 新しいセクション (22 ページの『WMQIB または WBIMB と共にアダプターを使用する方法』) が追加されました。ここでは、WMQI を統合ブローカーとして使用する場合に行う必要がある変更について説明しています。
- 本書の前のバージョンに記載されていた『付録 C. IndusConnect Framework 構成ファイル』は削除されました。
- 75 ページの『付録 C. コネクター構成の階層』は、コネクターのさまざまな構成プロパティがどのような関係でインプリメントされているかを示す目的で、本リリースから追加されました。
- 24 ページの『Latin 1 および Unicode 言語以外のメッセージ・ファイルの変換』が追加され、英語以外の言語によるメッセージの提供方法を説明しています。
- 本書の以前のバージョンに記載されていた『付録 D. コネクター機能リスト』は削除されました。

第 1 章 アダプターの概要

『アダプターの概要』

2 ページの『アダプターのアーキテクチャー』

概要

IBM WebSphere Business Integration Adapter for IndusConnect Framework により、Indus 統合製品 IndusConnect とのトランザクションが容易になります。Adapter for IndusConnect Framework を使用すれば、Indus International のコア・ソリューションと、ほかの優れたエンタープライズ・アプリケーションを統合することができます。このアダプターは、IndusConnect API を介して通信することにより、すべての IndusConnect Framework 統合ビジネス・オブジェクトをサポートしています。また、IndusConnect の標準ビジネス・オブジェクトのデータと WebSphere Business Integration の標準ビジネス・オブジェクトのデータを相互に変換する機能を備えています。このアダプターによって、Indus エンタープライズ資産管理 (EAM) ソリューションをほかのエンタープライズと容易に統合できます。IndusConnect Framework アプリケーションは、作業および資産の管理をサポートします。

アダプターの概要

アダプター (本書ではコネクタともいいます) は、アプリケーション固有のコンポーネントとコネクタ・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・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の間の仲介役として機能し、そのコードはすべてのコネクタに共通です。

コネクタは、WBI システムの変換サービスを提供し、データをコラボレーションと以下のもの間で移動します。

- アプリケーション。
- XML などの技術規格に対応し、WBI コネクタによって処理されるプログラマチック・エンティティ (リモート Web サーバーなど)。コネクタには以下の 2 つの特殊な終端があります。

コネクタ・コントローラーはコラボレーションと直接対話します。コネクタ・エージェントは、アプリケーションなどのプログラマチック・エンティティと直接対話します。

コネクタ・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。

- アプリケーション・イベントによって起動された場合は、ビジネス・オブジェクト (事前指定された情報のパケット) 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。

各コネクタは、アプリケーションのインターフェースに応じてアプリケーションと通信するため、固有です。アプリケーション・プログラミング・インターフェー

ス (API) がある場合は、コネクタはその API を使用できます。API のないアプリケーションのコネクタは、アプリケーションが提供する任意のメソッド (ユーザー出口や E メール・メッセージなど) を使用できます。

コラボレーションが対象とするアプリケーション・イベントを検出するために、コネクタは、アプリケーションをポーリングするか、アプリケーションのイベント・コールバック通知機構 (ある場合) を使用します。コネクタは、コラボレーションのコマンドでアプリケーションと対話したり、直前の要求の結果を検証したりすることもできます。

本書では、アプリケーション固有のコンポーネントとコネクタ・フレームワークについての情報を提供します。本書では、この 2 つのコンポーネントをまとめてコネクタと呼びます。

アダプターのアーキテクチャー

Adapter for IndusConnect Framework は、WebSphere Business Integration Adapter (WBIA) Framework と、WBIX と呼ばれるアダプター・フレームワーク拡張機能で構成されています。WebSphere Business Integration Extensions (WBIX) は、WBIA Framework 上に構築される一連のモジュラー・コンポーネントであり、実行時に構成され、インスタンス化されます。コンポーネントは、高水準の 2 つのトランスポートおよび 1 つ以上の論理モジュールです。論理モジュールは、発信元トランスポートからターゲット・トランスポートへの途中の内部ビジネス・オブジェクトを処理します。論理モジュールは、必要に応じて相互にチェーニングできます。**ルーティング・パス**は、あるトランスポートから特定の論理モジュールを経由してもう一方のトランスポートに至る特定のビジネス・オブジェクトのパスを定義します。XSLT 論理モジュールが処理できるようにするため、WBIX XML データ・ハンドラーを使用して、汎用ビジネス・オブジェクトとその XML 表現の間の変換を行います。図 1 に、アーキテクチャーの全体像を示します。

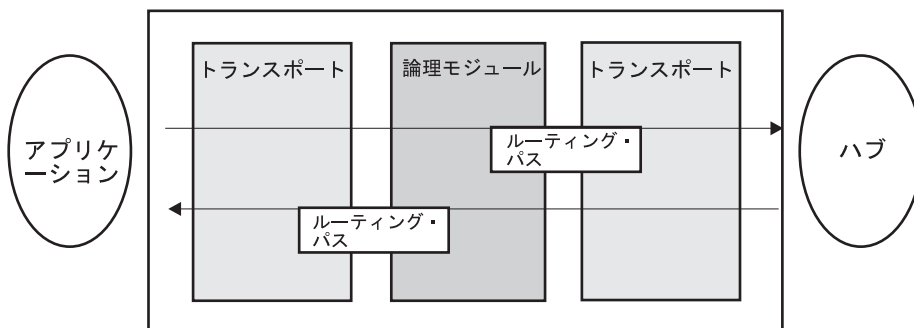


図 1. アダプター・アーキテクチャーの全体像

3 ページの図 2 に、より詳細なコンポーネントの関係を示します。これらの各コンポーネントの説明については後述します。

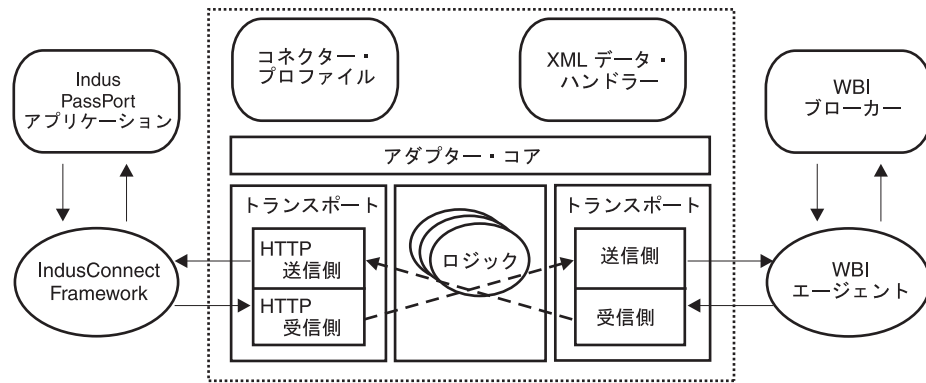


図 2. より詳しい説明図

アダプター・コンポーネント

以下にアダプター・コンポーネントを示します。

アダプター・コア

アダプター・コアは、アダプターのデータおよびロジックのフローを制御します。

コネクタ・プロファイル

コネクタ・プロファイルは、コネクタ構成とそのプロパティについての説明です。グラフィカル・ユーザー・インターフェースを備えたツールである Connector Configurator を使用して、これらの構成とプロパティを定義することができます。

WBIX XML データ・ハンドラー

WBIX XML データ・ハンドラーは、XSLT 論理モジュールが処理できるようにするため、汎用ビジネス・オブジェクトとその XML 表現の間の変換を行います。

イベント・ストア

イベント・ストアは、アダプターの処理に永続性を持たせるためのものです。デフォルトでは、イベントは、イベント・ストアに書き込まれるまでキューで保持されます。イベント・ストア・コンポーネントの詳細については、6 ページの『イベントのライフ・サイクル』および 7 ページの『イベント・ストア・ファイルの命名』を参照してください。

本書で解説するアダプターのイベント・ストアは、`com.ibm.wbix.adapter.transports.FileSystemEventStore` です。

トランスポート

トランスポートは、アダプターと統合ブローカーのインバウンド通信、アウトバウンド通信、およびパーシスタント通信のロジックをカプセル化したものです。トランスポートのアウトバウンド・コンポーネントは送信側です。トランスポートのインバウンド・コンポーネントは受信側です。パーシスタンス・コンポーネントはイベント・ストアです。

アプリケーションには、インターフェースの統合によってさまざまなプロトコルが混在している場合があります (例えば、あるデータには SQL データ

ベース・インターフェースを使用し、別のデータには MQ メッセージング・インターフェースを使用するなど)。したがって、単一のアダプターで複数のトランスポートをロードして、単一のアダプターで異なる複数のプロトコル間の通信が可能になります。さらに、各トランスポートには複数の着信 (受信側) エンドポイントと発信 (送信側) エンドポイントを含めることができます (例えば、アプリケーションのメッセージング・キューごとに 1 つの受信側または送信側を設定することができます)。トランスポートは、一部の共通インターフェースに対して用意されています。トランスポート・コンポーネントの詳細については、『トランスポート・モジュール』を参照してください。

論理モジュール

論理モジュールは、インバウンド・トランスポート受信側からアウトバウンド・トランスポート送信側に至る途中のデータを処理します。このモジュールは、アプリケーションのデータ形式と統合ブローカーのデータ形式を相互に変換する機能を備えています。例えば、XSLT 論理モジュールは、インバウンドとアウトバウンドの形式が共に XML 形式である場合のために用意されています。

カスタム論理モジュールを作成すると、独自の要件を満たす処理を行うことができます。例えば、独自の XSL ファイルを作成して、独自の文法変換を定義することができます。また、論理モジュールは、必要に応じて相互にチェーンングすることができます。

本書で解説するアダプターの論理モジュールは、`com.ibm.wbix.adapter.logicmodules.XsltLogicModule` です。

トランスポート・モジュール

統合ブローカーとの通信には、WebSphere Business Integration の標準のトランスポート・モジュール (送信側用と受信側用の両方) が使用されます。このとき、メッセージの本文は、アダプター (具体的には WBI トランスポート) で認識可能な XML 形式を備えている必要があります。このメッセージ本文は、WebSphere Business Integration の Java オブジェクトに変換され、統合ブローカーに送信されて処理されます。トランスポート・コンポーネントには、統合ブローカーと通信するモジュールと、アプリケーションと通信するモジュールが含まれます。

本書で解説するアダプターの WebSphere Business Integration トランスポートは、以下のコンポーネントで構成されています。

com.ibm.wbix.datahandlers.XMLDataHandler

このデータ・ハンドラーは、WBI ビジネス・オブジェクトとアダプターの WBI XML を相互に変換します。

com.ibm.wbix.adapter.transports.wbi.WBISender

このモジュールは構成パラメーターを必要としません。

com.ibm.wbix.adapter.transports.wbi.WBIReceiver

このモジュールは構成パラメーターを必要としません。

アダプター・サブレット

サブレットのセットは、アダプターおよび IndusConnect Framework と共に用途に応じて提供されます。これらのサブレットは、IndusConnect の外部イベント・テーブルの抽象として使用します。抽象を使用することにより、アダプターは、HTTP を介した一定の方法で IndusConnect と対話することができるようになります。サブレットを使用すると、アダプターで以下を実行できます。

- イベント・テーブルからのイベントの検索
- イベント・テーブル内のイベント状況の更新
- イベント・テーブルからのイベントの削除

サブレットは、J2EE Web アプリケーション (.war ファイル) としてパッケージされています。このアプリケーションは、IndusConnect アプリケーションをホスティングするアプリケーション・サーバーにインストールする必要があります。サブレットのデータベース・アクセスは、IndusConnect の `apifw.properties` ファイルを使用して構成されます。このファイル場所を解決するには、「`passPort_properties`」という名前の Java™ システム・プロパティにサブレットを渡す必要があります。このプロパティの値は、`apifw.properties` ファイルが常駐するディレクトリです。この構成により、IndusConnect およびサブレットが同じパラメーターを持つ同一データベースにアクセスできるようになります。Web アプリケーションのインストール方法、およびシステム・プロパティの提供方法に関する説明については、アプリケーション・サーバーの資料を参照してください。

以下に、アダプター・サブレットおよびそのパラメーターの説明を示します。

printEventTable

外部イベント・テーブル (TIDAPIEV) 内のイベントをすべて表示します。

getEventXML

イベントの完全なアウトバウンド・イベント XML 詳細を取得します。複数の文書が戻される場合があります。

eventID

イベントを識別する固有 ID (前述のタイム・スタンプ)。

注: `eventID` を設定しないと、テーブル内のすべての文書に対して XML が戻されます。

status (この引き数と等しい状況を持つすべてのイベントを) イベント状況ごとにフィルターに掛けます。

注: `status` パラメーターおよび `notStatus` パラメーターは、互いに排他的です。

notStatus

(この引き数と等しくないすべてのイベント状況を) イベント状況ごとにフィルターに掛けます。

removeEvent

外部イベント・テーブル (TIDAPIEV) からイベントを除去します。

eventID

イベントを識別する固有 ID (前述のタイム・スタンプ)。 `eventID = all` に指定すると、テーブル内のすべてのイベントが除去されます。

setStatus

イベント・テーブル内のイベントの状況を設定します。

eventID

イベントを識別する固有 ID (前述のタイム・スタンプ)。

status このイベントの状況として設定するストリング値。

イベントのライフ・サイクル

アプリケーションの Indus PassPort コンポーネントで生成されたイベントは、以下のようにして統合ブローカーに伝搬されます。

注: 以下の説明では、リモート・イベント・ストアは Indus PassPort イベント・テーブルを参照します。

1. Indus PassPort が、状況 PRE_POLL_STATUS のリモート・イベント・ストアにアプリケーション・イベントをパブリッシュします。
2. アダプター・サブレットが、次のポーリングで、パブリッシュされたイベントをアダプターに戻します。
3. アダプターが、状況 EVENT_RECEIVED のローカル・イベント・ストアにイベントを追加します。状況定数のリストについては、7 ページの表 1 を参照してください。
4. アダプターが、リモート・イベント・ストアのイベントの状況を POST_POLL_STATUS に更新します。
5. 状況変更対象のローカル・イベント・ストア内のイベントについて、以下の処理が発生します。
 - 変換による状況の変更
 - WBIA のイベント・ポーリングによる状況の変更
6. 処理とポーリングが完了すると、ローカル・イベントの状況が適切な状況値に変更されます。例えば、WBIA のイベント・ポーリングが完了した後は、ERROR_POSTING_EVENT または SUCCESS になります。
7. 最終のローカルの状況値がエラーの場合は、リモート・イベント・ストア内のイベントが ERROR_COMPLETION_STATUS に設定されます。成功した場合は、リモート・イベント・ストア内のイベントが SUCCESS_COMPLETION_STATUS に設定されます。
8. イベントがアーカイブされます。
 - アーカイブを有効にすると、ローカル・イベントがアーカイブ・ストアにコピーされます。
 - DeleteOnArchive オプションを設定すると、イベントがローカルおよびリモートのイベント・ストアから削除されます。

現行の構成は以下のとおりです。

PRE_POLL_STATUS=R

POST_POLL_STATUS = 9

ERROR_COMPLETION_STATUS = 7

SUCCESS_COMPLETION_STATUS = 8

イベント・ストア・ファイルの命名

IndusEventStore の FileSystemEventStore コンポーネントでは、提供されたデフォルトの永続イベント・ストア・インプリメンテーションとして、ローカル・ファイル・システムを使用します。アダプターを開始すると、イベント・ストアおよびイベント・ストア・アーカイブごとに個別のディレクトリーが作成されます。

アダプターは、イベントがストアに追加されるたびに、そのイベントの内容を新しいログ・ファイルに書き込みます。各イベント・レコードには、命名規則に従って名前が付けられます。この名前は、固有 ID の後にイベント状況を付加し、その間をアンダースコア () で区切ったもの (*uniqueID_status value* など) です。固有 ID は、メッセージのイベント ID を基に、ハッシュ関数を計算して生成されます。

このイベント・ストアを使用する場合は、受信側で生成されるイベント ID に、ご使用のファイル・システムで許可されていない特殊文字が含まれないようにしてください。Adapter for IndusConnect Framework では、イベント ID が、Indus PassPort イベント・テーブルからのアウトバウンド・イベントのタイム・スタンプになります。

イベント状況を変更する場合は、通常、イベント・ストアが更新された状況を反映するようにファイルを名前変更します。ただし、イベント本体の変更を示す状況変更の場合、例えば、IN_PROCESSING を READY_FOR_POLL に変更した場合、イベント・ストアはファイル内容を再書き込みします。

状況定数を表 1 に示します。エラー定数については、表 2 に示します。

表 1. 状況定数

状況定数	値
EVENT_RECEIVED	2147483647
IN_PROCESSING	2147483646
EVENT_PROCESSED	2147483645
READY_FOR_POLL	0
IN_PROGRESS	1
SUCCESS	3

エラー定数は以下のとおりです。

表 2. エラー定数

エラー定数	値
ERROR_IN_LOGIC_PROCESSING	-2147483647
UNSUBSCRIBED	2
ERROR_PROCESSING_EVENT	-1
ERROR_POSTING_EVENT	-2
ERROR_OBJECT_NOT_FOUND	-3

第 2 章 アダプターのインストールと構成

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

- 『互換性』
- 10 ページの『関連ソフトウェアのインストール』
- 10 ページの『インストール・プロセスの概要』
- 10 ページの『アダプターのインストール』
- 12 ページの『ライブラリーおよびコネクターのインポート』
- 13 ページの『新規ユーザー・プロジェクトの作成』
- 14 ページの『アダプターの構成』
- 18 ページの『ユーザー・プロジェクトの更新』
- 19 ページの『サーバーへのプロジェクトの配置』
- 20 ページの『構成および国際化対応』
- 22 ページの『System Monitor の始動』
- 22 ページの『コネクターの始動』
- 22 ページの『WMQIB または WBIMB と共にアダプターを使用する方法』
- 24 ページの『Latin 1 および Unicode 言語以外のメッセージ・ファイルの変換』

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。このバージョンの Adapter for the IndusConnect Framework をサポートするアダプター・フレームワークおよび統合ブローカーは、以下のとおりです。

- **アダプター・フレームワーク:** WebSphere Business Integration Adapter Framework バージョン 2.4
- **統合ブローカー:**
 - WebSphere InterChange Server バージョン 4.2.1 および 4.2.2
 - WebSphere MQ Integrator Broker バージョン 2.1 (CSD05 を適用済み)
 - 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)
 - WebSphere Business Integration Message Broker バージョン 5.0 (CSD02 を適用済み)

例外については、「リリース情報」を参照してください。

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

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

関連ソフトウェアのインストール

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

以下のアプリケーション・プラットフォームのいずれか:

- Windows 2000 (Service Pack 4 を適用済み)
- AIX 5.1 (保守レベル 4 を適用済み)
- AIX 5.2 (保守レベル 1 を適用済み)
- HP-UX 11.11
- Sun Solaris 7.0 または Solaris 8.0

1.0 バージョンから、Adapter for IndusConnect Framework は Microsoft Windows NT 上ではサポートされなくなりました。

以下のサポート・ソフトウェア:

- IndusConnect Framework バージョン 9.0.3

インストール・プロセスの概要

インストール処理は、サーバーへのコンポーネントのインストールおよびクライアントへの製品のインストールから構成されます。クライアントとサーバーは同じ物理システムにすることができます。本書では、コンポーネントをサーバーまたはクライアントのいずれかとします。

以下のステップに従って、Adapter for IndusConnect Framework をインストールします。

1. 統合ブローカーのインストール。
2. IndusConnect Framework 用アダプターのインストール。
3. IndusConnect Framework 用アダプターの構成および初期化。

アダプターのインストール

アダプター・インストール・ソフトウェアは、CD 上で、または Web から ESD (Electronic Software Delivery) をダウンロードして使用できます。IBM WebSphere Business Integration IndusConnect Framework 用セットアップ・ファイルをシステムにダウンロードするには、パスポート・アドバンテージを使用します。製品の部品

番号も ESD イメージを参照します。Adapter for IndusConnect Framework のインストール手順については、「*WebSphere Business Integration Adapters* インストール・ガイド」で詳しく説明しています。この資料は、「リリース情報」と共に、InfoCenter (<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>) から入手できます。

Windows システムへのアダプターのインストール

ESD イメージからインストールする場合は、ファイル・ダウンロード・ユーティリティーによって、一連のインストール・ファイルを含む実行可能ファイルがマシンにコピーされます。アダプターをインストールするには、以下のステップを実行します。

1. セットアップ・ディレクトリーまたは CD から、ご使用のプラットフォームに適したセットアップ・ファイル `setupwin32.exe` を実行します。InstallShield が表示されます。
2. 言語を選択し、「OK」をクリックします。
3. 「次へ」をクリックして、画面を順に進めます。

注: ファイルの場所を選択するように求められます。

- ICS を実行している場合、ファイルは必ず ICS のインストール先と同じ製品ディレクトリー(例: `C:\IBM\WebSphereICS`) にインストールしてください。以下のステップでは、ICS ディレクトリーを *ICSdir* と呼びます。
- WebSphere MQ Integrator Broker または WebSphere Business Integration Message Broker を実行している場合、ファイルは必ず WebSphere Business Integration Adapter パッケージのインストール先と同じ製品ディレクトリーにインストールしてください。デフォルトは `C:\IBM\WebSphereAdapters` です。

「参照」ボタンを使用し、正しいディレクトリーの場所を探して選択します。

UNIX システムへのアダプターのインストール

アダプターを UNIX プラットフォームにインストールする場合は、以下のステップを実行します。

1. ご使用のシステムに適したセットアップ・ファイルを実行します。
 - AIX[®] をご使用の場合、`setupAIX.bin` を実行します。
 - HP-UX をご使用の場合、`setupHP.bin` を実行します。
 - Solaris をご使用の場合、`setupsolarisSparc.bin` を実行します。
2. 以下のようにファイルのディレクトリーを指定します。
 - a. ICS を実行している場合、ファイルは必ず ICS のインストール先と同じ製品ディレクトリーにインストールしてください。デフォルトは `/opt/IBM/WebSphereICS` です。
 - b. WebSphere[®] MQ Integrator Broker または WebSphere Business Integration Message Broker を実行している場合、ファイルは必ず WebSphere Business Integration Adapter パッケージのインストール先と同じ製品ディレクトリーにインストールしてください。デフォルトは `/opt/IBM/WebSphereAdapters` です。

インストール済みファイルの構造

表3 に、アダプターで使用されるファイル構造と、システムにインストールされるファイルをリストします。

注:

1. 本書では、ディレクトリー・パスとして円記号 (¥) を使用しています。UNIX システムの場合には、円記号をスラッシュ (/) に置き換えてください。
2. すべての製品パス名は、システム上の、製品のインストール先ディレクトリーを基準とした相対パス名です。

表3. システムにインストールされたアダプターのファイル

%Indus% のサブディレクトリー	
	説明
	内容: <ul style="list-style-type: none">• CustDataHandler.jar - WBIX 固有の XML データ・ハンドラーを含む .jar ファイル。• Indus.jar - アプリケーション固有のアダプター・クラスを含む .jar ファイル。• start_Indus.bat および start_Indus.sh - アダプター用の開始スクリプト。• WBIX.jar - WBIX フレームワーク・アダプター・クラスを含む .jar ファイル。• indusicl.zip - (System Manager 内の) カスタマー・ユーザー・プロジェクトへのインポートに必要な ICL .zip ファイル。• Indus.war - IndusConnect マシンにインストールするアダプター・サーブレットを含む Web アプリケーション・アーカイブ。
translations	オブジェクト・マッピング変換用のディレクトリー。変換マネージャー構成ファイルである schemaTypes.xml を含みます。
translations¥xslt	実際の変換マッピング・ファイル (XSL スタイルシート) を含むディレクトリー。
lib	アダプター用の Java ライブラリー・クラスを含むディレクトリー。

ライブラリーおよびコネクターのインポート

アダプター用のライブラリーをインポートする必要があります。以下のステップでは、統合コンポーネント・ライブラリーを *wbieuicl* と呼びます。この名前は、独自のライブラリー名で置換できます。

以下のステップを実行し、ライブラリーをインポートします。

1. WebSphere Business Integration System Manager を始動します。「スタート」→「プログラム」→「**IBM WebSphere InterChange Server**」→「**IBM WebSphere Integration Toolset**」→「管理」→「**System Manager**」を選択します。

注: この手順を初めて実行するときには、「ウィンドウ」→「パースペクティブを開く」→「その他」→「**System Manager**」を選択し、「OK」を選択する必要があります。

「プロジェクト」ウィンドウに以下のものが表示されます。

ユーザー・プロジェクト
統合コンポーネント・ライブラリー

2. 「統合コンポーネント・ライブラリー」を右マウス・ボタンでクリックします。
3. 「新規統合コンポーネント・ライブラリー」を選択します。
4. 「プロジェクト名 (Project name)」フィールドにライブラリー名 *wbieuicl* を入力し、*wbieuicl* という名前のフォルダーを作成します。
5. 「完了」をクリックします。
6. 「ファイル」 → 「インポート」をクリックし、「インポート」ウィンドウを開きます。
7. 「Zip ファイル」を選択します。
8. 「次へ」をクリックします。
9. 「Zip ファイル」フィールドで、*indus icl.zip* にブラウズします。ICS を実行している場合、このファイルは *ICSdir/connectors/Indus* にあります。
10. 「フォルダー (Folder)」フィールドで、*wbieuicl* 用に作成したフォルダーにブラウズします。
11. 「完了」をクリックし、すべてのファイルをフォルダーにインポートします。
12. *wbieuicl* フォルダーを右マウス・ボタンでクリックします。
13. 「最新表示」を選択します。表示に、インポート済みのすべてのビジネス・オブジェクトが表示されます。

新規ユーザー・プロジェクトの作成

以下のステップでは、プロジェクトを **INDUS** プロジェクトと呼びます。任意のプロジェクト名を指定できますが、名前はインストール・ステップおよび構成ステップを通じて一貫していなければなりません。

System Manager を使用して以下のステップを実行します。

1. ご使用のシステムに ICS がインストールされている場合は、「ユーザー・プロジェクト」を右マウス・ボタンでクリックし、「新規 ICS プロジェクト (New ICS project)」を選択して、ステップ 4 に進みます。
2. 「ユーザー・プロジェクト」を右マウス・ボタンでクリックし、「新規ユーザー・プロジェクト」を選択します。
3. 「新規 WAS プロジェクト (New WAS project)」または「新規 Message Brokerプロジェクト (New Message broker project)」を選択します。
4. プロジェクト名 **INDUS** を指定します。
5. 下のウィンドウで、*wbieuicl* ライブラリー・フォルダー名の横にあるチェック・ボックスを選択します。12 ページの『ライブラリーおよびコネクターのインポート』のステップ 4 で別のライブラリー名を指定した場合は、ここではそのライブラリー名を使用してください。
6. 「完了」をクリックし、すべてのビジネス・オブジェクトを持つフォルダー **INDUS** を作成します。

アダプターの構成

アダプターのコネクタ・コンポーネントには、2種類の構成プロパティがあります。1つは標準の構成プロパティで、ほとんどのアダプターに適用されます。もう1つはコネクタ固有の構成プロパティで、ご使用のアダプターにのみ適用されます。このコネクタ固有の構成プロパティを使用すると、コーディングやビルドをやり直さずにコネクタ内部の静的な情報やロジックを変更することができます。コネクタは、始動時に構成値を取得します。コネクタを実行する前に、必ずこれらのプロパティの値を設定してください。

コネクタ・プロパティを構成するには、Connector Configurator ツールを使用します。詳細については、55ページの『付録 B. Connector Configurator』を参照してください。このツールは、コネクタを構成するためのグラフィカル・ユーザー・インターフェースを提供します。

コネクタの構成プロパティの値を指定すると、Connector Configurator は、アダプターのリポジトリにその値を保管するか (ICS を使用している場合)、構成ファイルを生成してアダプターのローカル・リポジトリに格納します (WebSphere MQ Integrator Broker または WebSphere Application Server を使用している場合)。

プロパティの指定

このセクションでは、プロパティを指定するときに実行する必要があるステップについて説明します。この手順で使用するフォルダーのプロジェクト名は、*INDUS* です。

プロパティを指定するには、WebSphere Business Integration System Manager を始動して、以下のステップを実行します。

1. 「ユーザー・プロジェクト」フォルダーを展開します。
2. 「InterChange Server プロジェクト」を展開します。
3. 「INDUS」フォルダーを展開します。
4. 「コネクタ」フォルダーを展開します。
5. 「IndusConnector::wbieuc1」をダブルクリックします。
6. 「標準のプロパティ」タブをクリックし、構成プロパティを追加または変更します。
7. アダプターの AgentTraceLevel を指定します。値 **5** を選択すると、詳細なトレース・レベルが生成されます。ゼロ (**0**) を選択した場合は、トレース情報は生成されません。
8. ご使用のアプリケーションによっては、DeliveryTransport プロパティに以下のどのトランスポートを指定するかを選択するように求められることがあります。

MQ

IDL (デフォルトのトランスポート・メソッド)

JMS

そのほかの標準プロパティの構成については、デフォルト値を受け入れます。

続いて、トランスポート関連のコネクタ固有のプロパティを定義します。

9. 「コネクター固有プロパティ」タブをクリックします。
10. 「wbiadapter」をクリックして展開します。
11. 「transports」をクリックして展開します。
12. 「Indus」を展開します。
13. 「senders」を展開します。
14. 「IndusSenderReceiver」を展開します。
15. 「parameters」を展開します。
16. 表 4 のリストのプロパティに、適切な値を指定します。

表 4. Adapter for IndusConnect Framework のコネクター固有の構成プロパティ (トランスポート関連)

プロパティ/パラメーター	説明	値	デフォルト値
sendUrl	送信側コンポーネントがイベントを送信する先の URL です。この URL は完全修飾されていなければならない、HTTP 経由で解決可能でなければなりません。	このパラメーターは、APIFWAdapterServlet などの IndusConnect 処理サーブレットを指している必要があります。	使用不可
pollingInterval	IndusConnect の新しいイベントを検索するために poll() メソッドを呼び出すまでの待機時間 (ミリ秒単位)。ポーリングの詳細については、20 ページの『ポーリング・パラメーターに関する考慮事項』を参照してください。	整数値 (ミリ秒単位)。ポーリングを停止する場合は「no」。	30000
prePollStatus	ポーリング時に検索するイベントの状況レベルです。この状況のイベントは poll() メソッドで戻されます。	このオプションの値は、単一の文字でなければなりません。	R
postPollStatus	ポーリング時に、ポーリング後にイベントを設定する状況レベルです。poll() メソッドで戻されるすべてのイベントの状況レベルがこの値に設定されます。	このオプションの値は、1 桁の整数でなければなりません。	9
apifwVersion	APIFW ビジネス・オブジェクトのバージョン この値は、処理要求に対する DTD およびヘッダーを構成するために使用します。	ストリング	V090000
dtdLocation	IndusConnect Framework サーバーのビジネス・オブジェクト DTD 文書のロケーション。この値は、処理要求に対する DTD ヘッダーを構成するために必要です。	IndusConnect Framework のホストとなっているサーバーにある、IndusConnect.DTD ファイルが格納されている xml ディレクトリーの名前 (ストリング) です。	使用不可
environment	apiFW に必要な Indus PassPort 環境名。IndusConnect Framework アプリケーション環境。この値は、処理要求に対する DTD ヘッダーを構成するために必要です。		D903

表 4. Adapter for IndusConnect Framework のコネクタ固有の構成プロパティ (トランスポート関連) (続き)

プロパティ/パラメーター	説明	値	デフォルト値
username	アダプターが要求を処理する際に使用するアプリケーション (Indus Passport など) のユーザー名。	ストリング この値は、 apifw.properties apiUsername プロパティに一致している必要があります。このプロパティについては、IndusConnect Framework の資料を参照してください。	使用不可
DefaultEncoding	サーバーから受信した文書に ContentEncoding ヘッダーが設定されていない場合に、その文書で使用されているとみなされるデフォルトのエンコード方式。	Java の有効なエンコード方式 ストリング	UTF-8
DefaultSendMethod	URL のアクセス方法としてデフォルトで GET を使用するか POST を使用するかを示すパラメーター。	GET または POST	POST
generateErrorObject	IndusConnect Framework から受信したエラー情報や警告情報からビジネス・オブジェクトを別途生成する必要があるかどうかをアダプターに対して指示するフラグ。生成されたオブジェクトは、indusErrorObject という名前になります。	True または False	False
OverrideServerEncoding	サーバーで設定された ContentEncoding ヘッダーを DefaultEncoding プロパティの値でオーバーライドするかどうかを指示するフラグ。	True または False	False

17. 「eventstore」を展開します。
18. 「parameters」を展開します。
19. 表 5 のリストのプロパティに、適切な値を指定します。

表 5. Adapter for IndusConnect Framework のコネクタ固有の構成プロパティ (イベント・ストア関連)

プロパティ/パラメーター	説明	値	デフォルト値
fetchURL	このストアがイベントを検索する URL。	{IBM servlet location}/getEventXML	使用不可
fetchIDParameter	リモート・サイトが検索するイベントの識別に使用するパラメーター名。URL に置く場合、このパラメーターの値は fetchRecords メソッドに渡された eventID 引き数になります。		eventID
fetchStatusParameter	イベント・ストアが設定状況のイベントの検索に使用するパラメーター名。URL に置く場合、このパラメーターの値は fetchRecords メソッドに渡された status 引き数になります。		status

表 5. Adapter for IndusConnect Framework のコネクタ固有の構成プロパティ (イベント・ストア関連) (続き)

プロパティ/パラメーター	説明	値	デフォルト値
postPollStatus	ポーリング後に設定するイベント状況。この値は、イベントを HTTP イベント・ストアからファイル・システム・イベント・ストアにコピーする時点を示すために使用します。	このオプションの値は、IndusSenderReceiver クラスで指定した値に一致する 1 桁の整数でなければなりません。	9
updateURL	イベント・ストアのイベントを更新可能な URL。	{IBM servlet location}/setStatus	使用不可
updateIDParameter	リモート・サイトが更新するイベントの識別に使用するパラメーター名。URL に置く場合、このパラメーターの値は setEventStoreStatus メソッドに渡された eventID 引き数になります。		eventID
useGetMethod	URL へのアクセス方法として GET を使用する必要があるかどうかをイベント・ストアに対して指示します。実稼働環境では、この値を FALSE に変更します。	TRUE (GET を使用) または FALSE (POST を使用)	TRUE
eventDir	イベント・ストアが存在するディレクトリー。この引き数は、相対値にすることも完全修飾することもできます。		eventStore
archive	アーカイブ・ストアが存在するディレクトリー。アーカイブをインプリメントしない場合は、この引き数をヌルにするか、省略します。		archive
archiveFailureFatal	システムがアーカイブの初期化に失敗した場合に、致命的例外を戻すかどうかを示すブール値。	True または False	False
successCompletionStatus	アダプターがイベントを正常に消費した後 (統合ブローカーにイベントを正常に渡した後など) に設定するイベント状況。	このオプションの値は、1 桁の整数でなければなりません。	8
errorCompletionStatus	アダプターがイベントの消費に失敗した後 (ロジック処理で障害が発生したか、統合ブローカーへのイベントの送信に失敗した後など) に設定するイベント状況。	このオプションの値は、1 桁の整数でなければなりません。	7
deleteOnArchive	イベントのアーカイブが要求されたときにイベントを削除するかどうかを示すブール値。	True または False	False
updateStatusParameter	リモート・サイトが指定のイベントの新規状況の識別に使用するパラメーター名。URL に置く場合、このパラメーターの値は setEventStoreStatus メソッドに渡された status 引き数になります。		status
deleteURL	ストアからのイベントの削除をサポートする URL。		使用不可

表 5. Adapter for IndusConnect Framework のコネクタ固有の構成プロパティ (イベント・ストア関連) (続き)

プロパティ/パラメーター	説明	値	デフォルト値
deleteIDParameter	リモート・サイトが削除するイベントの識別に使用するパラメーター名。URLに置く場合、このパラメーターの値は deleteEvent メソッドに渡された eventID 引き数になります。		eventID

20. 「**logicmodules**」を展開します。
21. 「**TranslationManager**」を展開します。
22. 「**parameters**」を展開します。

アダプターは、1 次変換メカニズムとして XSLT 論理モジュールを使用してアダプターの XML メッセージと WBI XML の間の変換を行います。XML 形式間の変換は直接行われ、チューニングされることはありません。デフォルトとは異なる特性を持つように論理モジュールを構成するには、表 6 のリストのプロパティに独自の値を指定します。

表 6. Adapter for IndusConnect Framework のコネクタ固有の構成プロパティ (論理モジュール関連)

プロパティ/パラメーター	説明	値	デフォルト値
schemaConfig	アプリケーションのデータ形式と統合ブローカーのデータ形式を相互に変換できるようにします。	repository_dir¥schemaTypes.xml	translations ¥schemaTypes.xml
repository	xslt サブディレクトリーは、トップレベルの変換ディレクトリーに対して、リポジトリー内のネーム解決のルートとして指定されます。	repository_dir¥xslt	translations¥xslt
Indus	ルーティング・パスを構成するために使用します。		CW
CW	ルーティング・パスを構成するために使用します。	この値は、ルーティング・パスに一致していなければなりません。	Indus

23. ステップ 8 (14 ページ) でデリバリー・トランスポートとして MQ を選択した場合は、以下のようにしてキュー・マネージャー名を指定する必要があります。
 - a. 「**メッセージング**」タブをクリックします。
 - b. 「**キュー・マネージャー**」フィールドにキュー・マネージャー名を指定します。デフォルトは **server.queue.manager** です。
24. 変更後に、「**ファイル**」 → 「**保管**」 → 「**プロジェクトに**」をクリックします。
25. Connector Configurator を終了します。

ユーザー・プロジェクトの更新

ユーザー・プロジェクトを更新するには、以下のステップを実行します。

1. 「ユーザー・プロジェクト」で、「INDUS」フォルダーを右マウス・ボタンでクリックします。
2. 「プロジェクトを更新」を選択します。
3. 「wbieuicl」フォルダーを展開します。
4. 「コネクター」フォルダーを展開します。
5. IndusConnector にチェックマークを付けます。
6. 「完了」をクリックします。
7. 「上書きの確認」プロンプトで「はい」をクリックします。

サーバーへのプロジェクトの配置

System Manager の「InterChange Servers」ウィンドウには、すでにローカル・サーバーのインスタンスが存在します。ここに記載するステップでは、サーバー・インスタンスを *INDUSserver* と呼びます。以下のステップを実行します。

1. WebSphere InterChange Server が実行されていることを確認します。
2. 「InterChange Server インスタンス」フォルダーを展開します。
3. 「*INDUSserver*」を右マウス・ボタンでクリックします。

注: サーバー・インスタンスが「サーバー」の下に表示されない場合は、まず以下のようにしてサーバーを登録する必要があります。

- a. 「InterChange Server インスタンス」を右マウス・ボタンでクリックします。
 - b. 「サーバーを登録」を選択します。
 - c. 必要な情報を指定して、「新規サーバーを登録」ダイアログを完了します。
 - d. 「OK」をクリックします。
4. 「接続」を選択します。
 5. 「ユーザー名」フィールドと「パスワード」フィールドにそれぞれ情報を指定します。
 6. 「OK」をクリックします。
状況が「接続しました」に変化します。
 7. 「ユーザー・プロジェクト」を展開します。
 8. 「InterChange Server プロジェクト」を展開します。
 9. 「INDUS」を選択します。
 10. 「INDUS」を右マウス・ボタンでクリックし、「ユーザー・プロジェクトの配置 (Deploy user project)」を選択して、プロジェクト・リストを開きます。
 11. *INDUS* にチェックマークを付けます。
 12. 「スキーマを作成」および「コンパイル」を選択します。
 13. 宛先サーバーとして「*INDUSserver*」を選択します。
 14. 「次へ」をクリックします。
 15. 「INDUS」フォルダーが表示された場合は、チェック・ボックスを選択します。

16. 「完了」をクリックします。このステップは、完了までに数分かかる場合があります。
17. 「OK」をクリックします。
18. 以下のようにして、InterChange Server をシャットダウンし、再始動します。
 - a. 「INDUSserver」を右マウス・ボタンでクリックします。
 - b. 「シャットダウン」を選択します。
 - c. 「正常」を選択します。
 - d. 「OK」をクリックします。
 - e. 「OK」をクリックします。
 - f. InterChange Server を始動します。

ポーリング・パラメーターに関する考慮事項

統合ブローカーに伝送する必要があるビジネス・オブジェクトの有無を IndusConnector がポーリングする頻度は、ポーリング・パラメーターに基づいて制御されます。アプリケーションがコラボレーション処理の一端でバッチ・モードでの要求を行う場合は、ポーリングの頻度を低くすることができます。ポーリング頻度を高に設定すると、パフォーマンスが低下する場合があります。コラボレーションの両端で、アプリケーション間のポーリングのバランスを取るよう to してください。ポーリング間隔の制御には、コネクタ固有のプロパティである `pollingInterval` を使用します。

サーバーがアダプターからイベントを受信する頻度は、コネクタの標準プロパティである `PollFrequency` に基づいて制御されます。ただし、このプロパティの値を変更する必要はありません。

構成および国際化対応

データ・ハンドラーはアダプターから呼び出されます。データ・ハンドラーは構成を必要とします。この構成には、数値と日付の形式の変換を定義するロケール固有のプロパティがあります。 `DataHandlerMetaObjectName` プロパティはデータ・ハンドラーを使用するアダプター内で定義する必要があります。このプロパティの値は、各 MIME タイプに適切なデータ・ハンドラーを定義するメタ・ビジネス・オブジェクトの名前です。 `text/xml` タイプに指定されたメタオブジェクトは、WBIX XML データ・ハンドラーおよびその構成オプションを指定します。

これらのオプションを構成するには、ビジネス・オブジェクト `MO_Indus_XMLConfiguration` を編集する必要があります。 `System Manager` を使用し、以下のパラメーターに応じて `Business Object Designer` のプロパティを編集します。編集が完了したら、このオブジェクトをサーバーに再展開する必要があります。

オプションは以下のとおりです。

UseDefaults

GBO にデフォルト値を取り込むかどうかを示すブール値。パラメーターがメタオブジェクト内に存在しない場合は、デフォルトの振る舞いとしてデフォルトを使用します。

AppDateFormat

データ・ハンドラーがアプリケーションとの間でのやりとりに使用する日付形式。Indus アダプターのデフォルトの日付形式は、yyyy-MM-dd hh:mm:ss です。

フォーマット設定について詳しくは、『日付形式』のセクションを参照してください。

EnforceRequired

要件を強制するかどうかを指定します。**TRUE** を指定すると、データを含まない必須フィールドを含むイベントがデータ・ハンドラーによって拒否されます。**FALSE** を指定すると、データを含まない必須フィールドもデータ・ハンドラーによって受け入れられます。

ClassName

Java クラス: com.ibm.wbix.datahandlers.XMLDataHandler

ICSDateFormat

データ・ハンドラーが統合ブローカーとの間でのやりとりに使用する日付形式。Indus アダプターのデフォルトの日付形式は、MM/dd/yyyy です。

フォーマット設定について詳しくは、『日付形式』のセクションを参照してください。

XmlEncoding

ICS オブジェクトから XML を生成する場合に使用するエンコードを指定します。有効なエンコードは以下のとおりです。

US-ASCII

7 ビット ASCII。ISO646-US や、Unicode 文字セットの基本ローマ字ブロックとも呼ばれます。

ISO-8859-1

ISO ローマ字アルファベット No. 1。ISO-LATIN-1 と呼ばれます。

UTF-8 8 ビット Unicode 変換フォーマット。

日付形式

日付形式は、以下の規則に従って表記されています。

- yyyy** 4 桁の形式の年。例えば、2003 です。
- MM** 2 桁の形式の月。例えば、10 月は 10 です。
- dd** 2 桁の形式の日付。例えば、18 日は 18 です。
- hh** 24 時間制の時間。例えば、午後 11:00 は 23 です。
- mm** 分。
- ss** 秒。
- SSSS** 1/1000 秒。

詳細については、Java 1.3.1 SimpleDateFormat の仕様を参照してください。

System Monitor の始動

System Monitor を始動してサーバーに接続します。各コネクターが表示されます。IndusConnector を探します。黄色のアイコンは、開始の準備ができていることを示します。

IndusConnector を右マウス・ボタンでクリックしてコネクターを開始します。正常に開始すると、アイコンが緑色に変化します。

コネクターの始動

Windows

```
start_Indus.bat Indus name_of_ICServer
```

UNIX /opt/IBM/WebSphereAdapters/bin/connector_manager -start Indus

WMQIB または WBIMB と共にアダプターを使用する方法

WebSphere MQ Integrator Broker (WMQIB) または WebSphere Business Integration Message Broker (WBIMB) のいずれかを統合ブローカーとして使用する場合、アダプターに変更を加える必要があります。この変更を加えることにより、デフォルトの wbieuAsset ビジネス・オブジェクト (デフォルトで再帰的に定義されるオブジェクト) が統合ブローカーで正常に処理されるようになります。ここで説明する変更は、wbieuAsset ビジネス・オブジェクトと、wbieuAsset ビジネス・オブジェクトを使用する XSL 変換ファイルに対して加える必要があります。

ビジネス・オブジェクト定義の変更

以下のステップを実行して、wbieuAsset ビジネス・オブジェクトに変更を加えます。

1. Business Object Designer を開きます。Business Object Designer の起動方法と使用方法については、「*IBM WebSphere Business Integration Adapters* ビジネス・オブジェクト開発ガイド」を参照してください。
2. wbieuAsset ビジネス・オブジェクトを開きます。
3. wbieuAsset ビジネス・オブジェクトのコピーを、subAsset などの別の名前で保管します。
4. 元の wbieuAsset ビジネス・オブジェクトで、subAsset 属性のタイプを変更します。新しいタイプは、ステップ 3 で作成したタイプにしなければなりません。
5. wbieuAsset ビジネス・オブジェクトに依存しているビジネス・オブジェクト (デフォルトでは wbieuWorkOrder ビジネス・オブジェクトと wbieuWorkTask ビジネス・オブジェクト) をすべてリフレッシュします。
6. 3 種類のビジネス・オブジェクトをすべて再配置します。

ここで説明したビジネス・オブジェクトの変更が完了したら、XSL 変換ファイルの変更を行う必要があります。

XSL 変換ファイルの変更

このセクションでは、wbieuAsset ビジネス・オブジェクトを使用する XSL 変換ファイルに変更を加える方法について説明します。

1. translations ディレクトリー内の以下のディレクトリーにある、convert.xml (XSL 変換ファイル) を開きます。

注: 作業を開始する前に、バックアップ・コピーを作成して、別のフォルダーに保管するか、ファイル拡張子を変更して保管してください。

```
xslt/Indus/cw/Equipment
xslt/Indus/cw/Schedule
xslt/Indus/cw/UniqueTrackedCmmdty
xslt/Indus/cw/WorkOrder
```

2. 各変換ファイルで、テキスト「subAsset」を検索します。このテキストは、**Attribute** 要素に含まれており、この XML 要素の **name** 属性として現れます。
3. subAsset 属性が検出された箇所のすべてで、**class** という名前の XML 属性を、元の wbieuAsset から、22 ページの『ビジネス・オブジェクト定義の変更』のステップ 3 (22 ページ) で作成した新しいビジネス・オブジェクトの名前に変更します。
4. 変更を加えた変換ファイルをすべて保管します。

以下に、変更を加える前のビジネス・オブジェクト定義と、変更を加えた後のビジネス・オブジェクト定義の例を示します。この例の変更は、Business Object Designer で subAsset 属性のタイプが wbieuAsset から newAsset に変更された場合の変更です。

変更前:

```
<xsl:element name="Attribute">
  <xsl:attribute name="class">wbieuAsset</xsl:attribute>
  <xsl:attribute name="name">subAsset</xsl:attribute>
  <xsl:attribute name="type">SINGLECARDSTRING</xsl:attribute>
  <xsl:element name="BusinessObject">
    <xsl:attribute name="class">wbieuAsset</xsl:attribute>
    <xsl:attribute name="type">SINGLECARDSTRING</xsl:attribute>
  </xsl:element name="Attribute">
```

変更後:

```
<xsl:element name="Attribute">
  <xsl:attribute name="class">newAsset</xsl:attribute>
  <xsl:attribute name="name">subAsset</xsl:attribute>
  <xsl:attribute name="type">SINGLECARDSTRING</xsl:attribute>
  <xsl:element name="BusinessObject">
    <xsl:attribute name="class">newAsset</xsl:attribute>
    <xsl:attribute name="type">SINGLECARDSTRING</xsl:attribute>
  </xsl:element name="Attribute">
```

Latin 1 および Unicode 言語以外のメッセージ・ファイルの変換

WBIX アダプター・フレームワークの拡張機能には、英語のメッセージおよびプロパティ・ファイルが含まれています。これらのメッセージを Latin 1 および Unicode 以外の文字を使用する他の言語に翻訳するには、以下の手順を実行します。

1. 該当するメッセージ・ファイル、例えば `WBIXResource.properties` ファイルをターゲット言語に翻訳します。このファイルをシステム・デフォルトのエンコード方式で保管します。
2. Java Development Kit を使用して、`native2ascii` ツールを翻訳済みのプロパティ・ファイルに対して実行し、文字を Unicode でエンコードされた文字に変換します。`native2ascii` ツールとそのコマンド行構文については、Javadoc の資料を参照してください。
3. Java Development Kit を使用して、`jar` ツールを Unicode でエンコードされたプロパティ・ファイルに対して実行し、これを `WBIX.jar` ファイルに入れます。翻訳済みの Unicode でエンコードされたプロパティ・ファイルを元の WBIX プロパティ・ファイルと同じパス・ロケーションに入れます。`jar` ツールとその構文については、Javadoc の資料を参照してください。

WBIX アダプター・フレームワークの拡張機能により、ターゲット言語および Unicode 文字でメッセージが表示されるようになります。

第 3 章 ビジネス・オブジェクトについて

この章では、IndusConnect Framework ビジネス・オブジェクトおよびそのマッピングについて説明します。以下のトピックについて説明します。

- 『ビジネス・オブジェクト』
- 26 ページの『ビジネス・オブジェクト構造の概要』
- 27 ページの『IndusConnect Framework ビジネス・オブジェクトのマッピング』

ビジネス・オブジェクト

コラボレーションおよびコネクタは、InterChange Server 経由でビジネス・オブジェクトを送受信することによって対話します。

ビジネス・オブジェクトは、データ・エンティティを反映した、動作可能なユニットと見なすことができるデータのコレクションです。例えば、データ・エンティティは (フォームのフィールドすべてを含め) フォームと等価にすることができます。通常、フォームはアプリケーションまたは Web 経由で使用し、顧客、従業員、送り状に関するビジネス情報を含みます。

ビジネス・オブジェクトは、アクセスを高速化するためにコラボレーション実行中にメモリーにキャッシュされるほか、永続トランザクション状態ストアにも格納され、障害後にサーバーが再始動したときのコラボレーションの堅固なりカバリー、ロールバック、および再実行を提供します。

WBI システムは、エンティティに含まれる情報を反映するビジネス・オブジェクトを作成します。本書では、データ・エンティティを、その中に含まれているビジネス情報のコンテキストの種類で言及することがあります (例: 従業員エンティティ、顧客エンティティ)。

ビジネス・オブジェクトの役割

ビジネス・オブジェクトは、イベント、要求、または応答として機能します。

IndusConnect Framework ビジネス・オブジェクトの概要については、27 ページの表 7 を参照してください。

イベント

ビジネス・オブジェクトは、アプリケーション・イベント (アプリケーションでデータ・エンティティに影響を及ぼした操作) の発生を報告できます。アプリケーション・イベントは、データのコレクションの作成、削除、または値の変更です。コネクタがアプリケーション・イベントを検出し、対象とするコラボレーションにビジネス・オブジェクトを送信するときには、ビジネス・オブジェクトがイベントを表現する役割を持つため、WBI システムのイベントと呼ばれます。例えば、コネクタがコラボレーションの代わりにアプリケーションに新規従業員エンティティをポーリングするとします。アプリケーションが新規従業員エンティティを作成すると、コネクタはコラボレーションにイベント・ビジネス・オブジェクトを送信します。

要求

通常、要求は以下の 2 つのうちいずれかの方法で生成されます。

- コラボレーションは、コネクターに対する要求としてビジネス・オブジェクトを送信し、アプリケーションのデータの挿入、変更、削除、または検索をコネクターに要求できます。
- サーバー・アクセス・インターフェースは、コラボレーションに対する要求としてビジネス・オブジェクトを送信できます (コラボレーションがトリガーとして Retrieve 動詞を受け入れるように設計またはカスタマイズされている場合)。

応答

通常、コネクターは要求の処理が完了すると応答を戻します。例えば、コネクターは、アプリケーションから従業員データを検索する要求を受信すると、従業員データを含むビジネス・オブジェクトを送信します。

ビジネス・オブジェクト構造の概要

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

- タイプ名
- サポートされる動詞
- 属性

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

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

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

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

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

IndusConnect Framework ビジネス・オブジェクトのマッピング

IndusConnect Framework オブジェクトは、WBI に送信可能なビジネス・オブジェクトです。WBI ビジネス・オブジェクトは、WBI から IndusConnect Framework に送信されるビジネス・オブジェクトです。

表 7. IndusConnect Framework ビジネス・オブジェクトのマッピング

IndusConnect Framework オブジェクト	WBI オブジェクト	IndusConnect Framework オブジェクトから WBI オブジェクトへの動詞	WBI オブジェクトから IndusConnect Framework オブジェクトへの動詞
Schedule	wbieuWorkSchedule	update	create、update、delete
SchedRefData	Crew ¹ および CrewMember ¹	update	-
WorkOrder	wbieuWorkOrder	replace	create、update、delete、retrieve
ServiceRequest	wbieuWorkRequest	-	create、update、delete
Equipment	wbieuAsset ²	insert、update、delete、replace	create、update、retrieve
UniqueTrackedCmmnty	wbieuAsset ²	insert、update、delete、replace	create、update、retrieve
Catalog	indusCatalog	insert、update、delete、replace	create、update、delete
Manufacturer	indusManufacturer	insert、update、delete	create、update、delete
MaterialRequest	wbieuMaterialRequest	insert、update、delete、replace	create、update、delete
Vendor	Vendor	insert、update、delete	create、update、delete
EntityAddress	wbieuPartner	insert、update、delete	-
ClientData	wbieuConsumer	-	create、update、delete

注:

1. 複数の Crew/CrewMember オブジェクトが 1 つの ScheduleRef オブジェクトから生成される場合があります
2. wbieuAsset は、utc 属性が存在するか serialNumber 属性が存在するかに基づいて Equipment オブジェクトまたは UniqueTrackedCmmnty オブジェクトにマップされます

第 4 章 トラブルシューティングおよびエラー処理

この章には、問題が発生した際の問題の分離に役立つ情報が含まれています。以下のトピックについて説明します。

- 『トレース・メッセージ』
- 30 ページの『プロファイルからアダプターを組み立てるときの問題』
- 31 ページの『トラブルシューティングのヒント』
- 32 ページの『エラー・ログを使用した問題の分離』
- 33 ページの『エラー処理』

トレース・メッセージ

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

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

表 8 に、IndusConnect Framework アダプターが各トレース・レベルで出力するトレース・メッセージのタイプを説明します。すべてのトレース・メッセージがコネクター・プロパティ `TraceFileName` によって指定されたファイルに表示されます。これらのメッセージは、IBM WebSphere Business Integration Adapter アーキテクチャーによって出力されるトレース・メッセージに追加されます。

表 8. トレース・メッセージ

トレース・レベル	トレース・メッセージ
レベル 0	コネクターのバージョンを識別するメッセージ。このレベルでは他のトレースは行われません。このメッセージは、常時表示されます。
レベル 1	<code>pollForEvents</code> メソッドが実行される度に配信されるメッセージ。
レベル 2	<ul style="list-style-type: none">• <code>gotApplEvent</code> から統合ブローカーにビジネス・オブジェクトが通知される度にログに記録されるメッセージ。• ビジネス・オブジェクト要求を受信する度に表示されるメッセージ。
レベル 3	メッセージからビジネス・オブジェクトへの変換、およびビジネス・オブジェクトからメッセージへの変換に関する情報を示すメッセージ。
レベル 4	<ul style="list-style-type: none">• アプリケーション固有の通知メッセージ。例えば、ビジネス・オブジェクトのアプリケーション固有の通知フィールドを構文解析する関数によって戻される値を示すメッセージ。• コネクターが、コネクターのプロセス・フローのトレースに役立つ関数を開始または終了したかどうかを識別するメッセージ。

表 8. トレース・メッセージ (続き)

トレース・レベル	トレース・メッセージ
レベル 5	<ul style="list-style-type: none"> コネクタの初期化を示すメッセージ。例えば、統合ブローカーから検索された各構成プロパティの値を示すメッセージ。 ビジネス・オブジェクト・ダンプから構成されるメッセージ。このトレース・レベルでは、コネクタは、(コネクタがコラボレーションから受信するオブジェクトを示す) オブジェクトの処理を開始する前、および (コネクタがコラボレーションに戻すオブジェクトを示す) オブジェクトの処理を終了した後に、ビジネス・オブジェクトのテキスト表現を出力します。

プロファイルからアダプターを組み立てるときの問題

ロードのエラー

プロファイルからのアダプターを構築している間に、コアは致命的なエラーと致命的でないエラーの両方を報告します。致命的エラーは、アダプターを構成できなかった場合や、プロファイルに致命的な不整合が存在した場合に発生します。

致命的エラー

致命的エラーは以下のとおりです。

クラスをロードできませんでした (Could not load class)

クラスが見つからないか、ロードしようとしたときにエラーが発生しました。

クラスを構成できませんでした (Could not configure class)

ロード中にロード対象クラスが例外をスローしました。原因として、正しいパラメーターを渡していないことが考えられます。

クラスを開始できませんでした (Could not start class)

クラスが動作を正しく開始できませんでした。アプリケーションに接続できないことが原因であると考えられます。

ルーティング・パスが競合しています (Routing paths conflict)

複数のルーティング・パスに同じ起点およびオブジェクトが含まれています。

指定のルーティング宛先は存在しません (Routing destination specified does not exist)

いずれかのルーティング・パスに、プロファイルの送信側にマップされていない宛先が指定されています。

指定のルーティング起点は存在しません (Routing origin specified does not exist)

いずれかのルーティング・パスに、プロファイルの受信側にマップされていない起点が指定されています。

指定のルーティング論理モジュールは存在しません (Routing logic module specified does not exist)

いずれかのルーティング・パスに、プロファイルに存在しない論理モジュールが指定されています。

致命的でないエラー

ロード中の致命的でないエラーには、正しく構成されたが不注意によるエラーを引き起こす可能性があるアダプターに関する警告が含まれます。致命的でないエラーの例を以下に示します。

CW コントローラー側からのサポートされないオブジェクト (Objects not supported from CW controller side)

WBI がブローカーの場合は、ICS のコネクタ・コントローラーが、アダプターが特定のビジネス・オブジェクトをサポートすることを要求する場合があります。この警告は、ICS からアダプターにオブジェクトを送信できない場合に作成されます。

特定経路が汎用経路をオーバーライドしています (Specific route overrides a Generic Route)

汎用経路をオーバーライドする特定経路がプロファイルに存在する場合は、警告が出されます。特定の受信側からのオブジェクトは特定経路をとり、汎用経路をとらないことがユーザーに警告されます。

致命的でないエラーはアダプターのパフォーマンスに影響を及ぼさないため、アダプターの動作を停止しません。しかし、ユーザーがアダプターを誤って構成した場合のために、警告メッセージが送信されます。

ルーティングのエラー

実行時には、アダプター・コアはすべてのエラーから回復しようとします。ただし、オブジェクトでルーティングのエラーが発生した場合は、コアはトランスポート受信側エラーを報告します。各トランスポート受信側は `handleError()` 関数を実装し、失敗したルーティングからの回復を試行する必要があります。ルーティング中に発生する可能性があるエラーの一部を以下に示します。

オブジェクトの経路が見つかりません (Route not found for object)

このエラーにはいくつかの理由が考えられます。まず、`MessageEvent` がルーティングを要求した受信側を正しく指定していない可能性があります。イベントをルーティングする前に、各トランスポート受信側は `MessageEvent` に名前を設定する必要があります。指定した名前の形式が誤っていた場合にもエラーが発生する場合があります。最後に、ルーティング・パスがプロファイルに存在しないため、適用する経路がコアに存在しない可能性もあります。

論理モジュールの処理が失敗しました (Logic Module processing failure)

処理中に、パスにあるいずれかの論理モジュールが例外をスローしました。

トラブルシューティングのヒント

トラブルシューティングの問題が発生した場合は、以下のヒントに従ってください。

- コネクタが実行しているかどうかをチェックしてください。
- サーバーが実行しているかどうかをチェックしてください。
- ビジネス・オブジェクトの構造がオペレーションと矛盾していないことを確認してください。

エラー・ログを使用した問題の分離

エラーが発生すると、アダプターはエラー・ログにエラー・レコードを書き込みます。エラーを調べるには、エラー・ログを検討して、特定のログ・メッセージを突き止めることにより、処理が失敗したかどうかおよび失敗時のイベント状況を判別します。イベントの内容を調べるには、ファイル・イベント・ストア内のレコードを検索する必要があります。障害が発生したことを示すエラー・レコードのほとんどは、状況が **FAIL** の状態でアーカイブに配置されます。つまり、ファイル名の最後の文字が **_3** で終わるファイルが存在します。中には、エラー状況 (最後が負の数値になっているファイル) のままイベント・ストア・ディレクトリーに存在するレコードもあります。ファイル内容を調べるには、イベント ID とイベント・レコード・ファイル名を関連付けるログ・メッセージを検索します。

```
(wbix_transport_msg0023=Translating eventID {0} to record name {1}).
```

次に、レコード・ファイルを調べて、エラーを引き起こしたイベントの内容を確認します。

失敗したイベントに対して講じる処置は、イベント・タイプおよびカスタマーの状態によって異なります。ただし、ほとんどの場合、失敗したイベントを回復して、ハブに再実行依頼することができます。

イベント・ストア内のトランザクション・データを使用して、障害を把握します。手動でデータをアダプターに再サブミットすることはお勧めしません。アダプターの **InDoubtEvents** 標準コネクター・プロパティーを設定するか、またはイベントを引き起こしたターゲット・アプリケーションを使用して、イベントを回復する必要があります。

失敗したイベントは、IndusConnect Framework から 2 通りの方法で再実行依頼することができます。推奨するメソッドは、Indus PassPort アプリケーションのオブジェクトを再編集することにより、IndusConnect Framework がオブジェクトをリパブリッシュする方法です。もう一方のメソッドは、外部イベント・ストア・テーブル内のイベント状況を **PrePollStatus** に再設定することにより、アダプターが次のポーリングでそのイベントを処理する方法です。

アーカイブには、すべてのイベント、つまり成功したイベントと失敗したイベントの両方に関する情報が含まれます。重複した ID を持つイベントはこの規則の例外です。特定の状況を持つ最初のイベントのみがアーカイブされます。そのため、イベントが ID 123 のアダプターを実行し、かつ、最初のイベントがブローカーに送信される前に、ID 123 の他のイベントが送信された場合、アーカイブには 2 つのレコード、つまり、成功したイベントのレコードと、重複のため失敗したレコードが格納されます。同じ ID を持つイベントがさらに送信されても、アーカイブに格納されるのはこれら 2 つのイベントのみです。ただし、一連のイベントに関するエラー・メッセージは、引き続きログに書き込まれます。

エラー・レコードの形式は以下のとおりです。

```
[timestamp][system][component][thread label][error message: error exception:
errorexception message]
```

以下のサンプルは、前の形式にマップするというエラーを示したものです。

```
[Time: 2003/06/25 15:37:05.715] [System: ConnectorAgent] [SS: IndusConnector]
[Thread: Indus.IndusSenderReceiver (#6276108)] Unable to extract
events from HTTP store: com.ibm.wbix.adapter.transports.http.HTTPTransportException:
XML parsing failed: org.xml.sax.SAXParseException:
The element type "MRAccounting" must be terminated by the matching end-tag "
</MRAccounting>"
```

エラー処理

アダプターで発生する可能性があるエラーは、以下の 3 種類に大別されます。

• アダプター構成エラー

始動時、またはビジネス・オブジェクトが処理されようとしたときに、致命的エラーまたは致命的でないエラーが発生することがあります。

- 始動時に致命的な構成エラーが発生すると、アダプターは始動できなくなります。この場合、即時にエラー・メッセージがログに記録されます。
- 同様に、オブジェクトが処理されるときに構成エラーまたはルーティングのエラーが発生した場合も、即時にログに記録されます。pollForEvents() の呼び出しの途中でこのようなエラーが発生した場合、戻されるエラーは、このメソッドの記述に合致するエラー・コードのみになります。doVerbFor() の途中でこのようなエラーが発生した場合、エラー・メッセージ、エラー戻りコード、およびメッセージの説明が、応答オブジェクトに含まれて、このメソッドを呼び出したコラボレーションに戻されます。

• ロジック処理エラー

エラーは、ロジック処理 (オブジェクトの変換) の際にも発生する可能性があります。ただし、この場合、通常は構成エラーや適切に記述されていない XSL 変換文書がエラーの発生原因です。このようなエラーは、pollForEvents() の呼び出しの途中では発生しません。doVerbFor() の途中でこのようなエラーが発生した場合、エラー・メッセージ、エラー戻りコード、およびメッセージの説明が、応答オブジェクトに含まれて、このメソッドを呼び出したコラボレーションに戻されます。

• トランスポート送信/受信エラー

エラーの大部分は、宛先アプリケーションとの間でオブジェクトが送受信されるときに発生します。

送信エラーは、doVerbFor() メソッドの途中で発生します。

- 通信エラーがアダプターにとって致命的なエラーとなるのは、入出力エラーが発生した場合です。この場合、APPRESPONSETIMEOUT 戻りコードが戻され、アダプターはシャットダウンします。
- 処理エラーが発生した場合や宛先アプリケーションでエラーが発生した場合には、コラボレーション例外オブジェクトに、戻りコードと必要なエラー・メッセージが含まれます。これらのメッセージは、アプリケーションから戻される文書に組み込まれており、アダプターの受信コンポーネントで処理されます。

受信エラーは、アプリケーションに新しいイベントの有無をポーリングしたとき、または送信されたオブジェクトへの応答を受信するときに発生する可能性があります。

- 致命的なエラーは、HTTP を使用しているアプリケーションにアダプターが正常に接続できないという入出力エラーです。このようなエラーはログに記録されます。次回に `pollForEvents()` メソッドや `doVerbFor()` メソッドが呼び出されたときには、`APPRESPONSETIMEOUT` 戻りコードが送信されます。
- 個々のメッセージの処理時に、致命的でないエラーが発生することがあります。通常、これらのエラーは、不適切な XML 構造が原因で発生します。このようなエラーは、`pollForEvents()` メソッドの途中でログに記録されます。該当のメソッドは、イベント・ストア内で保持され、再処理されなければなりません。
- `doVerbFor()` メソッドの途中で致命的でないエラーが発生した場合、アダプターは、宛先アプリケーションから送られたエラー・メッセージを、すべてログに記録します。`generateErrorObject` プロパティが `true` に設定されている場合、アダプターは、宛先アプリケーションからのエラー・メッセージや警告メッセージを含むエラー・オブジェクトを別途作成して、サーバーに送信しようとしません。処理エラーとそのメッセージは、アダプターによってログに記録されます。オブジェクトが追加で作成されることはありません。これらのエラー・メッセージは、いずれも、コラボレーションに戻されるコラボレーション例外オブジェクトに含まれます。

第 5 章 アダプターの拡張およびカスタマイズ

この章では、アダプターが使用する XML 文法について説明します。

XML 文法

XML 文法は、ビジネス・オブジェクトの属性 = 値の組を含む公開された定義です。XML 文法は、組み込みビジネス・オブジェクトおよびビジネス・オブジェクト・コンテナーの概念をサポートします。

このアダプターでは、ビジネス・オブジェクトを統合ブローカーへから送信するために XML 形式を使用します。この形式は、通常、Business Object Designer を介して統合ブローカー内に作成される新規ビジネス・オブジェクトをサポートするために定義します。このセクションでは、このフォーマットの内容 (XML スキーマ)、および XML インスタンスが統合ブローカーのビジネス・オブジェクトをどのように反映するかについて説明します。この形式を理解することにより、既存のビジネス・オブジェクトの変更だけでなく、サポートするアダプターに応じた新規オブジェクトの追加もできるようになります。

オブジェクト定義は `BusinessObject` タグで開始します。このタグの属性は以下のとおりです。

BusinessObject

class オブジェクトのビジネス・オブジェクト・タイプ。

type オブジェクトのカーディナリティー。値は常に `SINGLECARDSTRING` です。

verb このオブジェクトに関連した動詞。この属性は初期ドキュメント・タグにのみ必要です。

各 `BusinessObject` 要素の内容は `Attribute` です。属性は以下のとおりです。

Attribute

class ビジネス・オブジェクトのクラス。Attribute 値がビジネス・オブジェクトであるため、`BusinessObject` タグのようにそのオブジェクトのクラスを参照する場合にのみ使用します。

name オブジェクト定義内の属性名。

type 属性値のタイプ。この属性の値は、任意のビジネス・オブジェクト単純タイプにすることができます (例: `String`、`Date`)。この場合、タグの内容は属性値の `String` 表現です。

ビジネス・オブジェクトを属性として定義するには、タイプの値を `SINGLECARDSTRING` (単一カーディナリティー・オブジェクトの場合) または `MULTIPLECARDSTRING` (n カーディナリティー・オブジェクトの場合) とします。この場合、タグの内容は 1 つの `BusinessObject` 要素 (n カーディナリティーの場合は複数の要素) です。この文法により、任意の深さのビジネス・オブジェクトを表現できます。

次の例には、参考としてサンプル・オブジェクト定義とともにスキーマが含まれています。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="BusinessObject">
    <xsd:annotation>
      <xsd:documentation> Grammar definition for WBI Business Objects
    </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="1" ref="Attribute"/>
      </xsd:sequence>
      <xsd:attribute name="type" type="xsd:string" use="required"/>
      <xsd:attribute name="verb" type="xsd:string" use="required"/>
      <xsd:attribute name="class" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Attribute">
    <xsd:complexType mixed="true">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="BusinessObject"/>
      </xsd:choice>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="type" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

以下に単純な例を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<BusinessObject class="EMail" type="SINGLECARDSTRING" verb="Create">
  <Attribute class="Header" name="Header" type="SINGLECARDSTRING">
    <BusinessObject class="Header" type="SINGLECARDSTRING">
      <Attribute name="sendDate" type="Date">2003-1-18</Attribute>
      <Attribute name="subject" type="String">Greetings</Attribute>
    </BusinessObject>
    <Attribute class="Name" name="Recipients" type="MULTIPLECARDSTRING">
      <BusinessObject class="Name" type="SINGLECARDSTRING">
        <Attribute name="FirstName" type="String">John</Attribute>
        <Attribute name="LastName" type="String">Doe</Attribute>
      </BusinessObject>
      <BusinessObject class="Name" type="SINGLECARDSTRING">
        <Attribute name="FirstName" type="String">Jane</Attribute>
        <Attribute name="LastName" type="String">Doe</Attribute>
      </BusinessObject>
    </Attribute>
  </BusinessObject>
  <Attribute name="body" type="String">Hello All</Attribute>
</BusinessObject>
```

付録 A. コネクターの標準構成プロパティ

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

- WebSphere InterChange Server (ICS)
- WebSphere Message Brokers (WebSphere MQ Integrator (WMQI)、WebSphere MQ Integrator Broker (WMQIB)、および WebSphere Business Integration Message Broker (WBIMB))
- 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きには、そのブローカーで実行されるアダプターについて構成する必要のある標準プロパティのリストが表示されます。

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

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

新規プロパティと削除されたプロパティ

以下の標準プロパティは、本リリースで追加されました。

新規プロパティ

- XMLNamespaceFormat

削除されたプロパティ

- RestartCount
- 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 B『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 システム上で稼働している場合でも、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、コネクタ一用の 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。コネクタが System Manager から独立してスタンドアロン・モードで稼働している場合 (例えば、いずれかの WebSphere Message Brokers と連携している場合) 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9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、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9. 標準構成プロパティの要約

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

表9. 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9. 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	10000	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: <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9. 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	コンポーネント再始動	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	エージェント再始動	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

RepositoryDirectory が <REMOTE> 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 (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 crossworlds.queue.manager です。

jms.NumConcurrentRequests

コネクター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ー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

コネクター・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。

コネクター・メッセージ・ファイルが存在しない場合は、コネクターは `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: 特定のコネクターについて、コネクター独自のメッセージ・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

コネクター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。このプロパティーは、`DeliveryTransport` プロパティー値が `JMS` であり、かつ `DuplicateEventElimination` が `TRUE` に設定されている場合のみ使用されます。

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

OADAutoRestartAgent

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

コネクター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ーを再始動したり、System Monitor からリモート・コネクター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、このプロパティーを `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。このプロパティーを有効にするには、`OADAutoRestartAgent` プロパティーを `true` に設定する必要があります。

デフォルト値は `10000` 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・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、44 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。

ビジネス・オブジェクト定義の XML 形式でネーム・スペースを short と long のどちらにするかをユーザーが指定できるようにするための、強力なプロパティです。更新メソッドはコンポーネント再始動です。

デフォルト値は short です。

付録 B. Connector Configurator

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

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

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

注:

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

この付録では、次のトピックについて説明します。

- 『Connector Configurator の概要』
- 56 ページの『Connector Configurator の始動』
- 57 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』
- 60 ページの『新規構成ファイルの作成』
- 63 ページの『構成ファイル・プロパティの設定』
- 72 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 (WMQI)、WebSphere MQ Integrator Broker (WMQIB)、および WebSphere Business Integration Message Broker (WBIMB)。これらをまとめて MQ Message Brokers 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する。
- **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する。
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 (『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、57 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

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

Connector Configurator の始動

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

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

スタンドアロン・モードでの Configurator の実行

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

これを行うには、以下のステップを実行します。

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

- 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、WebSphere Message Brokers または 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こともできます (62 ページの『構成ファイルの完成』を参照)。

System Manager からの Configurator の実行

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

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

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

既存の構成ファイルを編集するには、以下のステップを実行します。

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

コネクタ固有のプロパティ・テンプレートの作成

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

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

- テンプレートの新規作成については、『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)

>= (より大が等しい)

<= (より小が等しい)

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

新規構成ファイルの作成

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

ブローカーを選択するには、以下のステップを実行します。

- Connector Configurator のホーム・メニューで、「ファイル」>「新規」>「コネクター構成」をクリックします。「新規コネクター」ダイアログ・ボックスが表示されます。
- 「Integration Broker」フィールドで、ICS 接続、WebSphere Message Brokers 接続、WAS 接続のいずれかを選択します。
- この章で後述する説明に従って「新規コネクター」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」>「プロジェクトに保管」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、WebSphere Message Brokers、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 65 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。

- 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・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。
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. 64 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 A 『コネクタの標準構成プロパティ』の 38 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。

サポートされているビジネス・オブジェクトを指定するには、指定するビジネス・オブジェクトとそのマップがシステム内に存在している必要があります。

- ビジネス・オブジェクト定義とマップ定義が System Manager プロジェクトに保管されている必要があります。
- 統合ブローカーとして WebSphere Message Broker を使用している場合、ビジネス・オブジェクト定義と MQ メッセージ・セット・ファイルが存在している必要があります。

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

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

MQ メッセージ・セット・ファイル (*.set ファイル) には、メッセージ・セット ID が記述されています。この ID は、コネクターがサポートするビジネス・オブジェクトを指定するときに、Connector Configurator で必要となります。MQ メッセージ・セット・ファイルの作成の詳細については、「*WebSphere MQ Integrator Broker* 用インプリメンテーション・ガイド」を参照してください。

ビジネス・オブジェクト定義をシステムに追加するごとに、Connector Configurator を使用して、コネクターがサポートするビジネス・オブジェクトを指定する必要があります。

サポートされるビジネス・オブジェクトを指定するには、以下のステップを実行します。

1. 「サポートされているビジネス・オブジェクト」タブを選択して、System Manager が実行中の場合は、ドロップ・リストを使用してビジネス・オブジェクト名を指定します。
2. System Manager が実行中でない場合は、「ビジネス・オブジェクト名」を入力します。
3. 「メッセージ・セット ID (オプション)」を入力します。

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、「サポートされるビジネス・オブジェクト」タブの「ビジネス・オブジェクト名」列の下にある空のボックスを選択できます。コンボ・ボックスが表示され、コネクタが属する統合コンポーネント・ライブラリー・プロジェクトから選択可能なビジネス・オブジェクトのリストが示されます。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。

関連付けられたマップ (ICS のみ)

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

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

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

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

「関連付けられたマップ」タブには以下のフィールドが表示されます。

- **ビジネス・オブジェクト名**

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

- **関連付けられたマップ**

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

- **明示的**

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

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

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

以下の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・エージェントが、コネクタ・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・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、コネクタによるキューの使用方法に影響します。

トレース/ログ・ファイル値の設定

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

ログとトレースの値を変更するには、以下の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『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が保管されます。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。

ICS の場合:

- System Manager から、ICS ユーザー・プロジェクトに `*.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。
- ファイルをローカル構成ファイルとして使用する場合には、スタンドアロン・モードで、`*.cfg` 拡張子が付いたファイルとしてディレクトリー・フォルダーに保管します。

WMQI の場合:

- 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。

- ファイルをローカル構成ファイルとして使用する場合には、スタンドアロン・モードで、*.cfg 拡張子が付いたファイルとしてディレクトリー・フォルダーに保管します。
- 指定したディレクトリーに保管します。

WAS の場合:

- System Manager から、WAS ユーザー・プロジェクトに *.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。

構成ファイルを作成し、そのプロパティーを設定した後に、ご使用のコネクターに対応した適切な場所にこの構成ファイルを配置する必要があります。

- 統合ブローカーとして ICS を使用している場合は、構成を System Manager プロジェクトに保管し、System Manager を使用してそのファイルを ICS に配置します。
- 統合ブローカーとして WebSphere Message Broker を使用している場合は、構成ファイルを適切な場所にコピーします。これは、ご使用のコネクターの始動ファイルに指定されている構成ファイルの場所とまったく同じ場所であればなりません。
- 統合ブローカーとして WAS を使用している場合には、ファイルを WAS ユーザー・プロジェクトに保管します。「ファイル」>「エクスポート」を選択し、.wsdl ファイルを作成します。作成された .wsdl ファイルは WSAD-IE ヘインポートできます。
また、構成ファイルを .jar ファイルとして指定のディレクトリーへエクスポートできます。

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

- ICS: 「WebSphere InterChange Server 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變更され、選択した新規ブローカーに適したタブとフィールドのみが表示されます。

構成の完了

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

これを行うには、コネクタ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>
```

```
        <DefaultValue>en_US</DefaultValue>
    </ValidValues>
</Property>
```


付録 C. コネクタ構成の階層

以下の例は、アダプター構成の階層を示しています。ここで示す内容は、Connector Configurator でコネクタ・プロパティを参照したときに表示される構造と関連しています。

wbiadapter	Must start with wbiadapter tag
transports	Must have transports tag with >= two child transports
<i>Broker Transport Name</i>	Name of transport is arbitrary
senderclass	Transport must have senderclass defined
receiverclass	Transport must have receiverclass defined
eventstoreclass	Transport can have one eventstore defined
isbroker TRUE	WBI transport must have TRUE for isbroker
receivers	If a transport has receivers, must have receivers tag
<i>Receiver</i>	Name of <i>Receiver</i> is arbitrary
parameters	If parameters are provided, must have parameters tag
<i>name value</i>	Each child is a name-value association
<i>nnn nnn</i>	Can have N number of parameters
<i>nnn</i>	Can have N receivers, each with parameters
senders	If a transport has senders, must have senders tag
<i>Sender</i>	Name of <i>Sender</i> is arbitrary
parameters	If parameters are provided, must have parameters tag
<i>name value</i>	Each child is a name-value association
<i>nnn nnn</i>	Can have N number of parameters
<i>nnn</i>	Can have N senders, each with parameters
eventstore	Can only have one eventstore per transport, and is optional
parameters	If parameters are provided, must have parameters tag
<i>name value</i>	Each child is a name-value association
<i>nnn nnn</i>	Can have N number of parameters
<i>nnn</i>	Can have N number of transports
logicmodules	Must have logicmodules tag, but can be empty below
<i>Name of logic module</i>	Arbitrary name for logicmodule
parameters	If parameters are provided, must have parameters tag
<i>name value</i>	Each child is a name-value association
<i>nnn nnn</i>	Can have N number of parameters
<i>nnn</i>	Can have N number of logicmodules
routingpaths	Must have routingpaths tag with at least one working route
<i>routingpath1</i>	Arbitrary name for routingpath
origin	Each route must have an origin tag
transport	Each origin must have transport tag
receiver	Each origin must have receiver tag, but value can be empty
destination	Each route must have a destination tag
transport	Each destination must have transport tag
sender	Each destination must have sender tag, but value can be empty
logicmodulerefs	Each route must have a logicmodulerefs tag, but cannot have child
<i>one logic module</i>	Arbitrary name (" <i>one</i> " for order reminder); value must match logicmodule
<i>nnn nnn</i>	Can have N number of logicmodule references
includeobjects	Each route must have an includeobjects tag, but cannot have child
<i>MyBusObjectQuery</i>	List of objects
<i>nnn</i>	Can have N number of objects supported
<i>nnn</i>	Can have N number of routingpaths

付録 D. 特記事項

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

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

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

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

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

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

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

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

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

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
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。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。

索引

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

[ア行]

アーキテクチャー
アダプター 2
アダプター
アーキテクチャー 2
アダプター構成エラー 33
アダプター・サブレット 5
インストール
概要 10
インポート、ライブラリーの 12
エラー処理
アダプター構成エラー 33
トランスポート受信エラー 33
トランスポート送信エラー 33
ロジック処理エラー 33
応答 26

[カ行]

規則 vi

[サ行]

サブレット、アダプター 5
始動、System Monitor の 22

[タ行]

トラブルシューティング
ヒント 31
トラブルシューティングのヒント 31
トランスポート受信エラー 33
トランスポート送信エラー 33

[ハ行]

パラメーター
ポーリングの設定 20
ビジネス・オブジェクト、変更 22
ビジネス・オブジェクト・マッピング 26
表記上の規則 vi
変換ファイル、XSL の変更 23
ポーリング・パラメーター
設定 20

ポーリング・パラメーターの設定 20
本書について v
本書の対象読者 v

[マ行]

まえがき v
マッピング、ビジネス・オブジェクト 26

[ラ行]

ライブラリーのインポート 12
ロジック処理エラー 33

B

Business Object Designer 22

S

System Monitor の始動 22

W

WBIMB アダプターの変更 22
WMQIB アダプターの変更 22

X

XSL 変換ファイル、変更 23

[特殊文字]

.war ファイル 5



Printed in Japan