

**WebSphere Business Integration Server  
Express and Express Plus**



## **Adapter for Lotus Domino ユーザーズ・ガイド**

*Adapter* バージョン 4.3.1



**WebSphere Business Integration Server  
Express and Express Plus**



## **Adapter for Lotus Domino ユーザーズ・ガイド**

*Adapter* バージョン 4.3.1

**お願い**

本書および本書で紹介する製品をご使用になる前に、特記事項に記載されている情報をお読みください。

本書は、IBM WebSphere Business Integration Server Express バージョン 4.3.1 および IBM WebSphere Business Integration Server Express Plus バージョン 4.3.1 に適用されます。

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

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

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

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

(URL は、変更になる場合があります)

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

**原 典：** WebSphere Business Integration Server  
Express and Express Plus  
Adapter for Lotus Domino User Guide  
Adapter Version 4.3.1

**発 行：** 日本アイ・ピー・エム株式会社

**担 当：** ナショナル・ランゲージ・サポート

第1刷 2004.8

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

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

© Copyright International Business Machines Corporation 2003, 2004. All rights reserved.

© Copyright IBM Japan 2004

---

# 目次

本書について	v
対象読者	v
関連文書	v
表記上の規則	vi
<b>本リリースの新機能</b>	<b>vii</b>
リリース 4.3.1 の新機能	vii
リリース 4.3 の新機能	vii
<b>第 1 章 アダプターの概要</b>	<b>1</b>
使用例	1
アダプターのアーキテクチャー	1
コネクターの動作方法	3
<b>第 2 章 アダプターのインストールと構成</b>	<b>7</b>
アダプター環境	7
インストール・プロセスの概要	9
アダプターおよび関連ファイルのインストール	9
インストール済みファイルの構造	9
Lotus Domino Server の変更	11
コネクターの構成	17
複数のコネクター・インスタンスの作成	20
コネクターの開始	22
コネクターの停止	24
<b>第 3 章 Lotus Domino アダプター・ビジネス・オブジェクト</b>	<b>27</b>
メタデータの定義	27
ビジネス・オブジェクト構造の概要	27
Lotus Domino ビジネス・オブジェクト定義	28
ビジネス・オブジェクトの生成	31
<b>第 4 章 トラブルシューティングおよびエラー処理</b>	<b>33</b>
イベント状況	33
エラー・メッセージ	33
トレース	39
<b>付録 A. コネクターの標準構成プロパティー</b>	<b>41</b>
標準コネクター・プロパティーの構成	41
標準プロパティーの要約	42
標準構成プロパティー	45
<b>付録 B. Connector Configurator Express</b>	<b>57</b>
Connector Configurator Express の概要	57
Connector Configurator Express の始動	58
System Manager からの Configurator Express の実行	58
コネクター固有のプロパティー・テンプレートの作成	59
新規構成ファイルの作成	61
既存ファイルの使用	62
構成ファイルの完成	64
構成ファイル・プロパティーの設定	64

構成ファイルの保管 . . . . .	70
構成の完了 . . . . .	71
グローバル化環境における Connector Configurator Express の使用 . . . . .	71
<b>付録 C. Lotus Domino ビジネス・オブジェクト属性 . . . . .</b>	<b>73</b>
DominoDocument ビジネス・オブジェクト属性 . . . . .	73
DominoItem ビジネス・オブジェクト属性 . . . . .	74
DominoItemValue ビジネス・オブジェクト属性 . . . . .	74
<b>索引 . . . . .</b>	<b>77</b>
<b>特記事項 . . . . .</b>	<b>79</b>

---

## 本書について

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

本書では、WebSphere Business Integration Server Express and Express Plus for Lotus Domino のインストール、構成、ビジネス・オブジェクト定義、およびトラブルシューティングについて説明します。

特に明記されていない限り、本書の情報は、いずれも、IBM WebSphere Business Integration Server Express と IBM WebSphere Business Integration Server Express Plus の両方に当てはまります。WebSphere Business Integration Server Express という用語と、これを言い換えた用語は、これらの 2 つの製品の両方を指します。

---

## 対象読者

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

---

## 関連文書

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

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

**注:** 本書の発行後に公開されたテクニカル・サポートの技術情報や速報に、本書の対象製品に関する重要な情報が記載されている場合があります。これらの技術情報や速報は、WebSphere Business Integration のサポート Web サイト (<http://www.ibm.com/software/integration/websphere/support/>) で参照できます。適切なコンポーネント領域を選択し、「Technotes (技術情報)」セクションと「Flashes (速報)」セクションを参照してください。

## 表記上の規則

本書では、以下のような規則を使用しています。

---

<code>courier font</code>	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、イタリック 青のアウトライン	初出語、変数名、および相互参照を示します。 青のアウトラインは、マニュアルをオンラインで表示するときのみ見られるもので、相互参照用のハイパーリンクを示します。アウトラインの内側をクリックすることにより、参照先オブジェクトにジャンプできます。
{ }	構文の記述行の場合、中括弧 {} で囲まれた部分は、選択対象のオプションです。1 つのオプションのみを選択する必要があります。
	構文の記述行の場合、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>&lt;server_name&gt;&lt;connector_name&gt;tmp.log</code> のように使用します。
<i>ProductDir</i>	製品のインストール先ディレクトリーを表します。各プラットフォームのデフォルトは、以下のとおりです。 Windows: <code>IBM¥WebSphereServer</code> OS/400: <code>/QIBM/ProdData/WBIServer43/product</code> Linux: <code>/home/\${username}/IBM/WebSphereServer</code>
/、¥	本書では、Windows のディレクトリー・パスの表記規則として円記号 (¥) を使用します。UNIX、OS/400、および Linux では、ディレクトリー・パスにスラッシュ (/) を使用します。すべての WebSphere Business Integration Server Express システム製品のパス名は、ご使用のシステムでの製品インストール・ディレクトリーからの相対パスです。

---



---

## 本リリースの新機能

---

### リリース 4.3.1 の新機能

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

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

---

### リリース 4.3 の新機能

本書の最初のリリースです。



---

## 第 1 章 アダプターの概要

本章では、WebSphere Business Integration Server Express and Server Express Plus Adapter for Lotus Domino および関連するシステム・アーキテクチャーについて説明します。

Adapter for Lotus Domino は、アダプターとアダプター・フレームワークからなります。アダプター・フレームワークは、IBM が提供するアダプター構成、実行のためのソフトウェアです。アダプターおよびアダプター・フレームワークを一緒に使用すると、Lotus Domino Server は InterChange Server Express と情報を交換できます。

Adapter for Lotus Domino を介して Lotus Domino Server バージョン R6 (6.0.2 以上) を統合できます。各 Lotus Domino Server には、それぞれに専用の Lotus Domino アダプター・インスタンスが必要です。ただし、1 つのアダプター・インスタンスで同一サーバー上の複数のデータベースに対応できます。

InterChange Server Express とアダプターの関係の詳細については、「システム管理ガイド」を参照してください。

本章の内容は、次のとおりです。

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

---

### 使用例

Lotus Domino アダプターを使用して、Lotus Domino Server で作成された Lotus Domino 文書に関する情報を他のアプリケーションへ送信できます。例えば、購入注文書が Lotus Domino データベースで作成されると、この作成処理が検出され、関連するデータが該当するアプリケーションへアダプターを介して配信され、このアプリケーションで処理されます。

他のエンタープライズ・アプリケーションで生成または変更されたデータに関する情報を、Lotus Domino データベースに送信するときにも、このアダプターが使用されます。例えば、社内従業員データベースで、ある従業員に関するデータが更新された場合に、アダプターは Lotus Notes を通じて従業員に Lotus Domino 文書による通知を送信します。また、他のアプリケーションで作成されたレポートを Lotus Domino 文書として再作成することもできます。

---

### アダプターのアーキテクチャー

InterChange Server Express と Lotus Domino Server 間の通信を可能にするアダプター・コンポーネントを以下に示します。

- コネクタ。コネクタにより、Lotus Domino Server と InterChange Server Express の間の両方向データ交換が可能になります。コネクタには、アプリケーション固有のコンポーネントおよびコネクタ・フレームワークが含まれていません。
- イベント表。Lotus Domino Server で発生したイベントに関する情報が格納されます。
- イベント・リスナー。イベント・リスナーにより、Lotus Domino Server で実行された機能（作成、削除、更新など）が検出され、イベント表にこれらの機能が記録されます。

Lotus Domino アダプターには、上記のコンポーネントを含むすべてのアダプター・コンポーネントが装備されています。アダプターをセットアップするには、アダプターをインストールして構成するだけです。

## コネクタ・アーキテクチャー

コネクタは、Lotus Domino と InterChange Server Express 間の通信を可能にする以下のコンポーネントで構成されています。

- アプリケーション固有コンポーネント。このコンポーネントは、イベント表をポーリングし、必要なイベントが検出されると、ビジネス・オブジェクトを作成して InterChange Server Express に送信します。また、このコンポーネントは InterChange Server Express からの着信ビジネス・オブジェクトを受け入れます。
- コネクタ・フレームワーク。コネクタ・フレームワークは、InterChange Server Express とアプリケーション固有のコンポーネントの間の中継を行います。

### アプリケーション固有コネクタ・コンポーネント

コネクタのアプリケーション固有コンポーネントは、以下のタスクを実行します。

- アプリケーション・イベントの検出
- アプリケーション・イベントのアーカイブ
- アプリケーション固有のビジネス・オブジェクトの作成
- InterChange Server Express 要求の処理
- エラー・メッセージとトレース・メッセージのログへの記録

ビジネス・オブジェクト・ハンドラーは、アプリケーション固有のコンポーネントに属しています。InterChange Server Express からアダプターに送信されたビジネス・オブジェクトを処理します。

### コネクタ・フレームワーク

コネクタ・フレームワークは、コネクタのアプリケーション固有のコンポーネントと InterChange Server Express の間の対話を管理します。ビジネス・オブジェクトを送受信し、始動メッセージや管理メッセージの交換を管理します。また、コネクタが必要とするメタデータをリポジトリから検索します。

## イベント表

イベント表は、構成ビュー、イベント・ビュー、およびアーカイブ・ビューという 3 つのデータ・セット (ビュー) からなります。構成ビューには、アダプターの構成時にイベント・リスナーが検出するイベントとして選択した Lotus Domino Server イベントのタイプに関する情報が格納されます。イベント・ビューには、実際のサーバー・イベントの情報が格納されます。アーカイブ・ビューには、処理されたイベントの状況情報が格納されます。

## イベント・リスナー

イベント・リスナーは、Lotus Domino Extension Manager 実行可能ライブラリーです。イベント・リスナーは Lotus Domino Server の拡張機能としてインストールします。Lotus Domino Server で発生したイベントを検出し、指定のイベントに関する情報をイベント表に保管します。アダプターのセットアップ時に、情報を格納するイベント、つまり、InterChange Server Express (および最終的には別のアプリケーション) へ渡すイベントのタイプを構成します。これにより、指定されたタイプの Lotus Domino Server イベントをリスナーが検出し、イベント表に記録します。イベント・リスナーの名前をサーバーの NOTES.INI ファイルに追加する必要があります。これにより、サーバーの始動時にイベント・リスナーも始動されます。(NOTES.INI ファイルの変更方法の詳細については、7 ページの『第 2 章 アダプターのインストールと構成』を参照してください。)

---

## コネクターの動作方法

コネクターのアプリケーション固有コンポーネントは、イベントについてイベント表のイベント・ビューをポーリングします。該当するイベント (subscribed-to イベントと呼ばれる場合もある) を検出すると、Lotus Domino Server からそのイベントに関する情報を検索し、アプリケーション固有のビジネス・オブジェクトを作成します。コネクターは、Lotus Domino Server への API 呼び出しによって取得したデータをビジネス・オブジェクトに取り込みます。次に、コネクター・フレームワークを通じてこのビジネス・オブジェクトを InterChange Server Express に送信します。

着信ビジネス・オブジェクト (InterChange Server Express からアダプターに送信されたビジネス・オブジェクト) について、コネクターは、InterChange Server Express からビジネス・オブジェクトを受信し、アクティブな動詞に従ってビジネス・オブジェクトを処理してから、Lotus Domino にオペレーション要求を送信します。

以降のセクションでは、Lotus Domino Server で発生したイベントが InterChange Server Express に渡される仕組みと、別のアプリケーションからの要求が Lotus Domino Server に転送される方法について説明します。

## イベント処理

Lotus Domino Server で発生したイベントは、次のように InterChange Server Express へ渡されます。

1. イベント・リスナーは始動時に、イベント表の構成ビューを読み取り、データベース名を取得し、モニターするイベントを判別します。

2. イベント・リスナーは、データベースからイベントを検出し、イベント表の構成ビューの情報に従ってこれらのイベントをフィルターに掛けます。イベントのタイプがイベント表の構成ビューに指定されているタイプである場合、このイベントに関する情報が維持されます。
3. イベント・リスナーは、このイベントに関する情報 (キー・データ) をイベント表のイベント・ビューに格納します。
4. コネクター (アプリケーション固有のコンポーネント) が、イベント表のイベント・ビューをポーリングし、イベントのキー・データを取得します。
5. コネクターのアプリケーション固有コンポーネントが、イベント・キー・データを使用して Lotus Domino Server から該当するイベント関連データをすべて検索します。
6. コネクターのアプリケーション固有のコンポーネントがビジネス・オブジェクトを作成して、コネクター・フレームワークに送信します。コネクター・フレームワークは、そのビジネス・オブジェクトを InterChange Server Express に送信します。
7. 他のアプリケーションが InterChange Server Express の Lotus Domino イベント情報を要求する場合には、InterChange Server Express がこのイベント情報をそのアプリケーションへ渡します。(この処理の詳細は異なります。詳細については、「WebSphere Business Integration Server Express インストール・ガイド」を参照してください。)

図 1 に、Lotus Domino アダプターによるイベントの処理方法を示します。

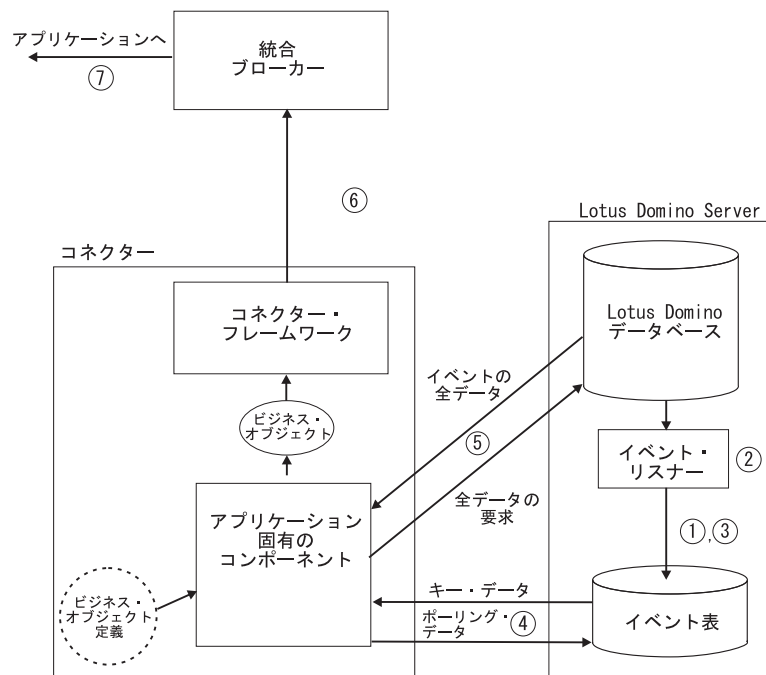


図 1. Lotus Domino アダプターのイベント処理

## 要求の処理

InterChange Server Express から発生するイベントの場合、次のように処理されます。

1. InterChange Server Express はビジネス・オブジェクトをコネクターに送信します。
2. コネクターが Lotus Domino Server とのセッションを確立します。
3. InterChange Server Express により Lotus Domino Server に渡された要求に対応するアクションが、このサーバーで実行されます。(例えば、Notes 文書のフィールド更新が要求された場合には、Lotus Domino データベースで値が更新されます。)
4. アダプターが、要求されたアクションの状況 (正常に完了したか、または問題が発生したかどうか) を InterChange Server Express に送信します。次に InterChange Server Express は、要求を発行したアプリケーションにこの情報を戻します。

図 2 に、Lotus Domino アダプターによる要求の処理方法を示します。InterChange Server Express は統合ブローカーです。

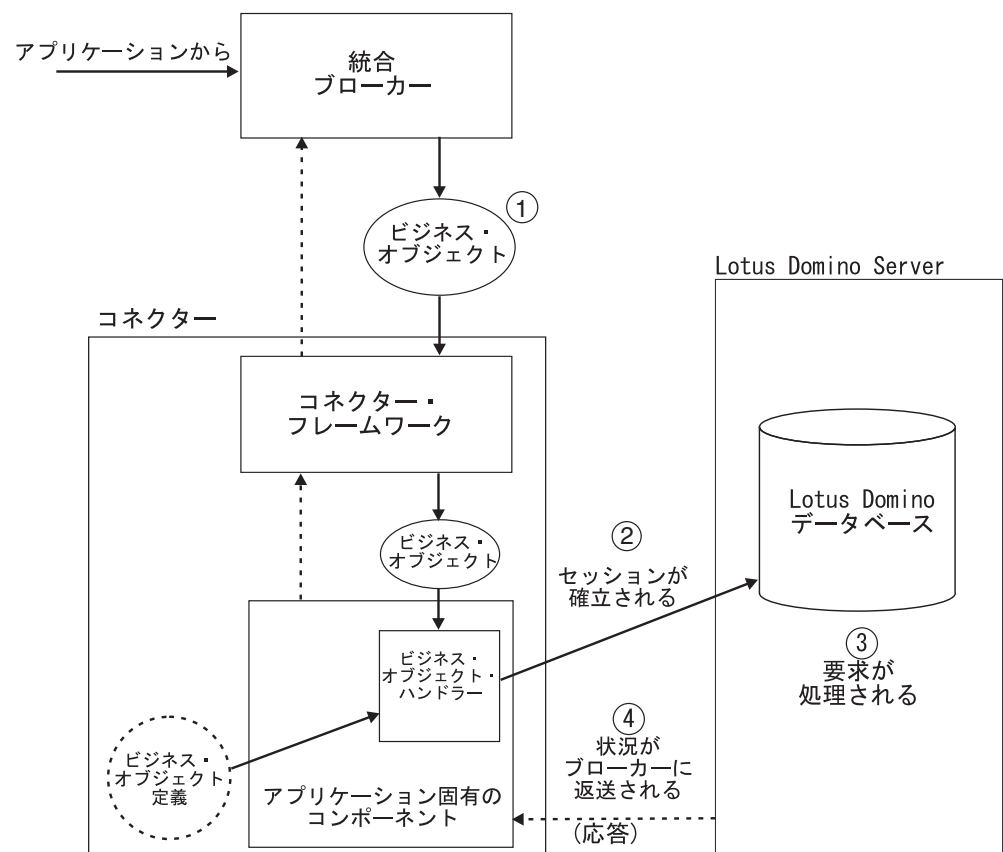


図 2. Lotus Domino アダプターの要求処理





---

## 第 2 章 アダプターのインストールと構成

本章では、WebSphere Business Integration Server Express for Lotus Domino のインストールおよび構成の方法と、このアダプターを使用して動作するアプリケーションの構成方法について説明します。本章は次のセクションから構成されます。

- 『アダプター環境』
- 9 ページの『インストール・プロセスの概要』
- 9 ページの『アダプターおよび関連ファイルのインストール』
- 9 ページの『インストール済みファイルの構造』
- 11 ページの『Lotus Domino Server の変更』
- 17 ページの『コネクターの構成』
- 20 ページの『複数のコネクター・インスタンスの作成』
- 22 ページの『コネクターの開始』
- 24 ページの『コネクターの停止』

---

### アダプター環境

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

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

### 互換性

使用するアダプター・フレームワークは、アダプターが通信する統合ブローカーのバージョンと互換性のある必要があります。Adapter for Lotus Domino バージョン 4.3.1 は、以下のアダプター・フレームワークおよび統合ブローカーでサポートされます。

アダプター・フレームワーク: Adapter Framework バージョン 4.3.1

統合ブローカー: InterChange Server Express バージョン 4.3.1

例外については、「リリース情報」を参照してください。

### アダプターのプラットフォーム

このアダプターは、次のソフトウェアでサポートされています。

オペレーティング・システム:

- Microsoft Windows 2000 (Service Pack 2, 3, またはそれ以降)

**注:** アダプター・インストーラーは、Windows XP 上にインストール可能であり、Windows XP ツールのサポートをインストールします。ただし、アダプターは Windows XP ホスト上では実行できません。

- Microsoft Windows 2003 (Lotus Domino R6 でのみサポート)
- IBM OS/400 V5R2、V5R3 (Lotus Domino R6 でのみサポート)
- Red Hat Enterprise Linux AS 3.0 (Update 1 を適用) (Lotus Domino R6 でのみサポート)
- SuSE Linux Enterprise Server 8.1 (SP3 を適用) (Lotus Domino R6 でのみサポート)

**データベース:** Lotus Domino

**サード・パーティー・ソフトウェア:** Lotus Domino Server バージョン R6 (6.0.2 以上)。

## アダプターの依存関係

Adapter for Lotus Domino は、Java 2 Runtime Environment (JRE) Standard Edition バージョン 1.3.1 以降に対する依存関係を持っています。

**注:** InterChange Server Express として、このブローカーと同じコンピューターにアダプターをインストールする場合には、インストール時に JRE が既にインストールされています。詳細については、「*WebSphere Business Integration Server Express インストール・ガイド*」を参照してください。

InterChange Server Express のインストール手順およびその前提条件については、「*WebSphere Business Integration Server Express インストール・ガイド*」を参照してください。

Lotus Domino Server のインストール方法については、Lotus Domino ソフトウェアに付属のインストール情報を参照してください。

## ロケール依存データ

アダプターは国際化され、2 バイト文字セットをサポートし、特定の言語でメッセージ・テキストを配信できるようになっています。アダプターは、ある文字コード・セットを使用する場所から別のコード・セットを使用する場所にデータを転送するとき、データの意味を保存するように文字変換を実行します。

Java 仮想マシン (JVM) 内部の Java ランタイム環境では、Unicode 文字コード・セットでデータを表現します。Unicode には、ほとんどの既知の文字コード・セット (1 バイト系とマルチバイト系を含む) の文字に対応できるエンコード方式が組み込まれています。WebSphere Business Integration システムのほとんどのコンポーネントは Java で記述されています。したがって、ほとんどの WebSphere Business Integration Server Express システム・コンポーネント間のデータ転送では、文字変換の必要はありません。

---

## インストール・プロセスの概要

Adapter for Lotus Domino を使用可能にするには、以下のステップを実行します。各ステップの詳細については、この章で後述します。

1. InterChange Server Express をインストールおよび構成します。InterChange Server Express は、Lotus Domino とアダプターのインストール先コンピューターと同じコンピューター、または別のコンピューターにインストールできます。
2. Lotus Domino がインストールされていない場合は、インストールおよび構成します。
3. アダプターをインストールするシステムに、Java 2 Runtime Environment をインストールします。
4. Adapter for Lotus Domino をインストールします。
5. Lotus Domino Server を変更します。この変更処理では、ファイルのコピー、一部の構成の変更、NOTES.INI ファイルの変更、およびイベント表の構成を行います。
6. アダプターが NCSO.jar ファイルへアクセスできるように設定します。
7. アダプターを構成します。
8. アダプターを始動します。

---

## アダプターおよび関連ファイルのインストール

アダプターのインストール方法の詳細については、以下の WebSphere Business Integration Server Express Infocenter のサイトにある Windows 版、OS/400 版、または Linux 版の「WebSphere Business Integration Server Express インストール・ガイド」を参照してください。

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

---

## インストール済みファイルの構造

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

10 ページの表 1 ではコネクターが使用する Windows ファイル構造を説明し、10 ページの表 2 では OS/400 ファイル構造を、11 ページの表 3 では Linux ファイル構造をそれぞれ説明します。

コネクター・コンポーネントのインストール方法の詳細については、「*IBM WebSphere Business Integration Server Express and Express Plus* インストール・ガイド」を参照してください。

## インストール済みファイルの構造 (Windows の場合)

表 1. コネクタ用としてインストールされた Windows ファイル構造

%ProductDir% のサブディレクトリー	説明
connectors¥Domino	始動スクリプト start_Domino.bat および start_Domino_service.bat が格納されています。また、BIADominoConnector.jar ファイルも格納されています。
connectors¥Domino¥dependencies	ファイル BIALD_EventTable.nsf、BIALD_emptytrash.gif、BIALD_movetotrash.gif、BIALD_newconfig.gif、および nbiald.dll が同梱されています。
connectors¥messages	BIADominoConnector.txt ファイルが格納されています。
repository	BIACN_Domino.txt および BIADominoDocument.txt ファイルが格納されています。

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

## インストール済みファイルの構造 (OS/400 の場合)

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

%ProductDir% のサブディレクトリー	説明
connectors/Domino	BIADominoConnector.jar ファイルと start_Domino.sh ファイル、およびサブディレクトリー dependencies が格納されています。
connectors/Domino/dependencies	ファイル BIALD_EventTable.nsf、BIALD_emptytrash.gif、BIALD_movetotrash.gif、BIALD_newconfig.gif、および libbiald.so が同梱されています。
connectors/messages	BIADominoConnector.txt ファイルが格納されています。
repository/Domino	BIACN_Domino.txt および BIADominoDocument.txt ファイルが格納されています。
bin/Data/App	BIA_Domino.xsd ファイルが格納されています。

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

## インストール済みファイルの構造 (Linux の場合)

表 3. コネクタ用としてインストールされた Linux ファイル構造

%ProductDir% のサブディレクトリー	説明
connectors/Domino	コネクタ BIADominoConnector.jar が格納されています。
connectors/Domino/dependencies	ファイル BIALD_EventTable.nsf、 BIALD_emptytrash.gif、 BIALD_movetotrash.gif、 BIALD_newconfig.gif、 および libbiald.so が同梱されています。
connectors/messages	BIADominoConnector.txt ファイルが格納されています。
repository	BIACN_Domino.txt および BIADominoDocument.txt ファイルが格納されています。

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

---

## Lotus Domino Server の変更

アダプターを実行する前に、Lotus Domino Server を次のように変更する必要があります。

1. Lotus Domino Internet Interoperability Protocol (IIOP) および HTTP サーバー (Web アプリケーション用サーバー) を稼動するようにサーバーを構成し、アプリケーションからのリモート・コールを許可するようにサーバーを構成します。これらの変更は、Lotus Domino 管理クライアントを用いて行います。
2. Lotus Domino Server の NOTES.INI ファイルを変更します。
3. イベント表および関連ファイルと、イベント・リスナーを Lotus Domino Server の該当するディレクトリーにコピーします。
4. イベント表を構成します。
5. アダプターが NCSO.jar ファイルへアクセスできるように設定します。

## Lotus Domino 管理クライアントを用いた構成変更

アダプターを作動させるため、Lotus Domino Server を、アプリケーションからのリモート・コールを受け入れるように構成する必要があります。構成は、Lotus Domino 管理クライアントを用いて行います (詳細については、Lotus Domino の管理マニュアルを参照してください)。以下のセクションで、構成を行う必要のある項目を説明します。変更内容を有効にするためには、変更後にサーバーを再始動してください。

### HTTP

Lotus Domino Server で、HTTP サーバー (Web アプリケーション用サーバー) を使用可能にします。セットアップ方法の詳細については、Lotus Domino の管理マニュアルを参照してください。

## IIOIP

Lotus Domino Internet Interoperability Protocol (IIOIP、DIIOP とも呼ばれる) を使用可能にします。IIOIP 用に、TCP/IP ポート番号が構成され、使用可能になっていることを確認してください。

## Java/COM

制限付き Java エージェントおよび制限なし Java エージェントを実行 (R6 の場合はメソッドも実行) するためには、アダプター用に構成されたユーザー名を構成する必要があります。また該当するものがあれば、着信ビジネス・オブジェクト内で指定されるユーザー名も構成します。

**注:** アダプター用に構成されたユーザー名は、Connector Configurator Express で構成された ApplicationUserName の値です。詳しくは、17 ページの『アプリケーション固有の構成プロパティ』を参照してください。

次のようにして構成します。

### R5 の場合:

1. 「セキュリティ」タブを選択します。
2. 「Java/COM の制限」セクションの、「制限なし Java/Javascript/COM の実行」および「制限付き Java/Javascript/COM の実行」の両方の下の「可能ユーザー」カラムに、Connector Configurator Express ファイル内で構成されている ApplicationUserName を入力します。また該当するものがあれば、着信ビジネス・オブジェクト内で指定されるユーザー名も入力します。

### R6 の場合:

1. 「セキュリティ」タブを選択します。
2. 「Programmability の制限」セクションの「可能ユーザー」列に、Connector Configurator Express ファイル内で構成されている ApplicationUserName を入力します。また、「制限付き Java/Javascript/COM の実行」の項目について、該当する場合は、着信ビジネス・オブジェクト内で指定されるユーザー名 (1 つまたは複数) も入力します。

## NOTES.INI の変更

使用しているオペレーティング・システムに応じて、Lotus Domino Server の NOTES.INI ファイルに次の行を追加します。

**Windows の場合:** EXTMGR\_ADDINS=nbiald.dll

**OS/400 の場合:** EXTMGR\_ADDINS=libbiald.so

**Linux の場合:** EXTMGR\_ADDINS=libbiald.so

これにより、Lotus Domino Server の始動時にイベント・リスナーが呼び出されません。

**注:** イベント・リスナーは Lotus Domino Extension Manager ライブラリーです。したがって、NOTES.INI ファイルで他の Lotus Domino Extension Manager ライブラリーと同じ行に記述する必要があります。記述するときには、コンマを使用してほかのエントリーと区切ります。

## Lotus Domino Server へのファイルのコピー

アダプター・ホスト・コンピューターの

%ProductDir%/connectors/Domino/dependencies ディレクトリーから、イベント表ファイルとイベント・リスナーを Lotus Domino Server の該当するディレクトリーにコピーする必要があります。次の表に、コピーする必要のあるファイルとコピー先ディレクトリーを示します。

表 4. Lotus Domino Server のディレクトリーにコピーするファイル

コピー先ディレクトリー	ファイル	説明
Lotus Domino データ・ディレクトリー	BIALD_EventTable.nsf	イベント表
Lotus Domino データ・ディレクトリー	BIALD_emptytrash.gif BIALD_movetotrash.gif BIALD_newconfig.gif	イベント表ビットマップ (Lotus Notes Client を介してイベント表を参照するときに表示される)
次のものをコピーします。		
Lotus Domino Server ルート	nbiald.dll	イベント・リスナー (Windows システム用)。このファイルは、Windows で Lotus Domino Server を実行している場合にのみコピーします。
Lotus Domino Server ルート	libbiald.so	イベント・リスナー (OS/400 および Linux 用) <b>注:</b> これは、OS/400 または Linux で Lotus Domino Server を実行している場合にのみコピーします。

### イベント表ファイルのコピー

インストール・ディレクトリーから Lotus Domino データ・ディレクトリーに次のファイルをコピーします。

BIALD\_EventTable.nsf

BIALD\_emptytrash.gif

BIALD\_movetotrash.gif

BIALD\_newconfig.gif

**Windows システムの場合:**

データ・ディレクトリーのデフォルトは `c:¥Lotus¥Domino¥Data` です。

#### **OS/400 システムの場合:**

データ・ディレクトリーのデフォルトは `/Domino/dominoServerName/data` です。ここで、`dominoServerName` は Lotus Domino Server の名前です。

#### **Linux システムの場合:**

このデータ・ディレクトリーのデフォルトは `/local/notesdata` です。

### **イベント・リスナーのコピー**

インストール・ディレクトリー (`%ProductDir%/connectors/Domino/dependencies`) から Lotus Domino Server のルート・ディレクトリーにイベント・リスナーをコピーします。

Lotus Domino Server を実行するオペレーティング・システムに応じて、次のファイルのいずれかをコピーします。

#### **Windows システムの場合: nbiald.dll**

インストール先ルート・ディレクトリーは通常 `c:¥lotus¥domino` になります。

#### **OS/400 システムの場合:**

次の CL コマンドを使用してシンボリック・リンクを作成する必要があります:  
`ADDLNK OBJ('/QSYS.LIB/QWBISVR43.LIB/LIBBIALD.SRVPGM')`  
`NEWLNK('/QIBM/UserData/LOTUS/NOTES/LIBBIALD.SRVPGM')`

#### **Linux システムの場合: libbiald.so**

インストール後、このファイルは `%CROSSWORLDS%/connectors/Domino/dependencies` にあります。

実行のために構成後、このファイルを `/opt/lotus/notes/latest/linux` にコピーします。

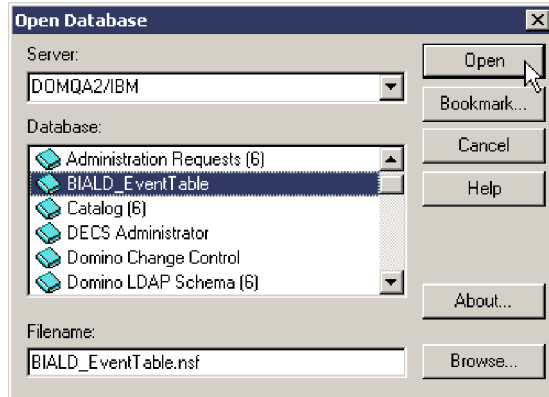
## **イベント表の構成**

アダプターのイベント・リスナーが、Lotus Domino Server で発生したイベントをモニターできるようにするには、アダプターのイベント表を構成する必要があります。イベント表は、Lotus Notes Client からオープンできる Lotus Domino データベース・ファイルです。

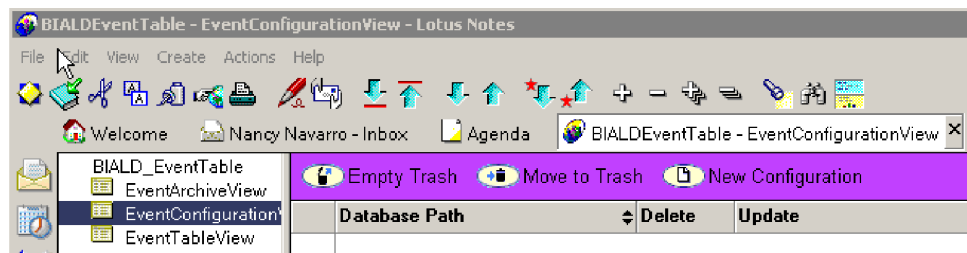
イベント表を構成するには、以下のステップを実行します。

1. Lotus Notes Client で「ファイル」>「データベース」>「開く」を選択し、このアダプターでサポートされている Lotus Domino Server の名前を選択します。
2. BIALD\_EventTable データベースを選択します。リストに表示されない場合には、「ファイル名」フィールドに `BIALD_EventTable.nsf` と入力するか、またはこのファイルを検索します。次のようなウィンドウが表示されます。





3. 「開く」をクリックします。次のウィンドウが表示されます。



4. 左側のパネルのナビゲーション・ビューで「EventConfigurationView」を選択します。
5. メニュー・バーで「新規サーバー設定」を選択します。次のテーブルが表示されます。

### Adapter for Lotus Domino ConfigurationTable:

Event Info	Value
Database Path:	/
Event Type:	<input type="checkbox"/> Create <input type="checkbox"/> Delete <input type="checkbox"/> Update

WebSphere Business Integration Adapter for Lotus Domino 構成テーブルが表示され、左側には「イベント情報 (Event Info)」列、右側には「値」列が表示されます。「イベント情報 (Event Info)」の下には、「データベース・パス (Database Path)」行とその下の「イベント・タイプ」行という、2つの行があります。

1 行目の右側の列にデータベース・パスを入力します。2 行目のチェック・ボックスを使用して、イベントに対する操作 (Create、Delete、または Update (この順序で表示される)) を選択します。

6. 「データベース・パス (Database Path)」フィールドに、アダプターがモニターするデータベースのパスを入力します。このパスは、*Domino server root directory/Data* を基準にした相対パスです。例えば、上の図を使用して、デー

データベース名が Buyer、パスが *Domino server root/Data/BusApp/Buyer.nsf* の場合には、「データベース・パス (Database Path)」フィールドに *BusApp/Buyer.nsf* と入力します。

- アダプターが特定のイベント・タイプをモニターできるようにするため、「イベント・タイプ」行で動詞のタイプまたは動詞のタイプの組み合わせを次のように選択します。

アダプターのモニター対象イベント	選択する動詞
Create イベント	Create と Update
Delete イベント	Delete
Update イベント	Update

**注:** アダプターは、Update イベントをモニターせずに、Create イベントのみをモニターすることはできません。Create イベントをモニターする場合は、チェック・ボックスを使用して、Create および Update を選択します。

- 「ファイル」>「保存」を選択して構成を保管し、イベント表データベースを閉じます。

## NCSO.jar ファイルをアクセス可能にする

NCSO.jar ファイルは、Lotus Domino Toolkit for Java/CORBA に含まれています。このツールキットは、Lotus Domino Server に付属しています。(ファイルは Lotus Domino Server および Notes Client で提供されています。) NCSO.jar には、Lotus Domino API (Domino Objects for Java として知られる) が含まれています。このファイルを指し示すようにアダプター始動スクリプトを更新する必要があります。

### NCSO.jar の格納先

NCSO.jar は一般に、Lotus Domino Server の次のディレクトリーに格納されています。

*DirectoryName*¥Data¥domino¥java

ここで、*DirectoryName* は、Lotus Domino Server のディレクトリーまたは Notes Client のディレクトリーの名前です。

### 始動スクリプトの変更

アダプター始動スクリプトで以下に示す行を変更し、NCSO.jar を指し示すようにします。

**Windows:** 次の行

```
set DOMINOCCLASSES=C:¥Notes¥Data¥domino¥java¥NCSO.jar
```

を次のように変更します。

```
set DOMINOCCLASSES=directoryname¥NCSO.jar
```

ここで、

*directoryname* は NCSO.jar が格納されているディレクトリーの名前です。

注: NCSO.jar ファイルが正しい場所を指すように変更する必要があるのは、Windows のみです。

---

## コネクターの構成

コネクターの構成プロパティには、標準構成プロパティとアダプター固有の構成プロパティという 2 つのタイプがあります。Connector Configurator Express (WebSphere Business Integration Server Express に付属のツール) を使用してアダプターを実行を実行する前に、これらのプロパティの値を設定する必要があります。コネクター構成プロパティの値の指定が完了すると、それらの値は Connector Configurator Express によりリポジトリに保管されます。これにより、アダプターの始動時にこれらのプロパティ値が使用可能になります。

Connector Configurator Express を使用してコネクター・プロパティを構成します。

- Connector Configurator Express の説明と手順については、57 ページの『付録 B. Connector Configurator Express』を参照してください。
- 標準コネクター・プロパティについては、『標準構成プロパティ』と 41 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。
- コネクター固有プロパティについては、『アプリケーション固有の構成プロパティ』を参照してください。

### 標準構成プロパティ

標準構成プロパティにより、すべてのアダプターによって使用される情報が提供されます。これらのプロパティの資料については 41 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。

コネクターは、始動時に構成値を取得します。実行時セッション中に、1 つまたは複数のコネクター・プロパティの値を変更することがあります。

- AgentTraceLevel などの一部のコネクター構成プロパティの変更は動的であり、変更内容が即時に有効になります。
- 他のコネクター・プロパティの変更は静的であるので、変更後にコンポーネントまたはシステムを再始動する必要があります。

動的プロパティであるか静的プロパティであるかを判別するには、Connector Configurator Express の更新メソッド列を参照してください。

ロギング・パラメーターとトレース・パラメーターは標準構成プロパティである点に注意してください。ロギングとトレースのセットアップ方法の詳細については、41 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。

### アプリケーション固有の構成プロパティ

アプリケーション固有のコネクター構成プロパティは、コネクターが実行時に必要とする情報を提供します。これらのプロパティを使用すれば、再コーディングや再ビルドせずに、コネクター内の静的情報やロジックを変更できます。

これらのプロパティーを表示および構成するには、Connector Configurator Express の「**アプリケーション構成プロパティー (Application Config Properties)**」タブを選択し、構成プロパティーを追加または変更します。詳しくは、57 ページの『付録 B. Connector Configurator Express』を参照してください。

表 5 に、コネクターのアプリケーション固有構成プロパティーと、各プロパティーの説明および指定可能な値を示します。

表 5. Lotus Domino のアプリケーション固有の構成プロパティー

プロパティー	説明	型	デフォルト値
DominoServerName	Lotus Domino Server のマシン名	String	なし
ApplicationUserName	Lotus Domino ユーザー名	String	なし
ApplicationPassword	Lotus Domino ユーザー・パスワード	String	なし
EventDBName	Lotus Domino イベント・データベース名	String	BIALD_EventTable.nsf
DocumentBOName	Lotus Domino のビジネス・オブジェクト名	String	DominoDocument
ForceSave	アダプターが Create 要求または Update 要求を処理するときに ForceSave が true であると、アダプターによる要求の処理時に文書が他のユーザーまたはコンポーネントにより編集、保管されている場合でも、この文書は強制的に保管され、前のバージョンの文書は破棄されます。ForceSave が false であり、アダプターによる要求の処理時に他のユーザーまたはコンポーネントがこの文書を編集している場合の処理結果は、MakeResponse プロパティーによって決まります。	Boolean	false
MakeResponse	アダプターが Create 要求または Update 要求を処理するときに、ForceSave が false であり MakeResponse が true の場合には、現行の文書が元の文書への返答になります。MakeResponse が false の場合には、保管操作が取り消されます。ForceSave が true の場合には、MakeResponse プロパティーの設定値は無効です。	Boolean	true
ForceDelete	アダプターが Delete 要求を処理するときに ForceDelete が true であると、アダプターにより開かれた文書を別のユーザーが変更している場合でも、この文書が削除されます。ForceDelete が false であり、アダプターにより開かれた文書を別のユーザーが変更している場合には、この文書は削除されません。	Boolean	false
InDoubtEvents	InProgress イベントのリカバリー・オプションです。指定できる値は Reprocess、FailOnStartup、LogError、または Ignore です。これらのオプションの詳細については、表 6 を参照してください。	String	Reprocess

表 5. Lotus Domino のアプリケーション固有の構成プロパティ (続き)

プロパティ	説明	型	デフォルト値
DateFormats	着信ビジネス・オブジェクトの Created および LastModified 日付のフォーマット。InterChange Server Express からビジネス・オブジェクトを受信すると、Lotus Domino アダプターはそのビジネス・オブジェクトの Created および LastModified 日付を Lotus Domino Server が解釈できる Java 形式に変換する必要があります。セミコロン (;) で区切ることで、複数のフォーマットを指定できます。アダプターは着信日付をリスト順に、指定されたフォーマットと比較します。フォーマットが指定されていない場合は、アダプターのホスト・システムの日付フォーマットが使用されます。日付フォーマットは、Java 2 Platform, Standard Edition (J2SE) v1.3.1 の java.text.SimpleDateFormat API で必要な日付フォーマットに準拠している必要があります。	String	MM/dd/yy hh:mm aaa; MM/dd/yy

DominoServerName、ApplicationUserName、および ApplicationPassword を構成する必要があります。InterChange Server Express から受信されるオブジェクトでサーバー名、ユーザー名、およびパスワードが定義されている場合には、Lotus Domino Server とアダプターのセッションを確立するときこれらの情報が使用されます。ビジネス・オブジェクトにこれらの情報が含まれていない場合には、アダプターでは構成ファイルでこれらの項目に定義されている値が使用されます。

InDoubtEvents に指定できる値については、表 6 を参照してください。未確定イベントとは、イベントの送信が成功または失敗したかどうかを示すイベント状況を更新する前にアダプターが終了した場合に、アダプターの終了時に処理中であったイベントです。アダプターを再始動すると、アダプターは処理中だったイベントをイベント表で確認し、InDoubtEvents プロパティに設定されているオプションに基づいて処理を行います。

表 6. InDoubtEvents プロパティに指定できる値

InDoubtEvents の値	説明
Reprocess	後続のポーリング呼び出しでイベントが再実行依頼されます。 注: この値を指定すると、イベントが重複して作成される可能性があります。イベントの重複を防ぐには、別のオプションを使用します。
FailOnStartup	致命的エラーをログに記録してアダプターをシャットダウンします。
LogError	エラーをログに記録しますが、アダプターをシャットダウンしません。
Ignore	イベント表の進行中イベント・レコードを無視します。

## 複数のコネクタ・インスタンスの作成

**注:** このアダプター (あるいは WebSphere Business Integration Server Express または Express Plus に付属の任意のアダプター) の追加インスタンスを作成すると、そのアダプター・インスタンスは、配置できるアダプターの総数を制限するライセンス機能によって、別のアダプターとしてカウントされます。

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

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

### 新規ディレクトリの作成

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

- Windows プラットフォームの場合:

*%ProductDir%\%connectors%\connectorInstance*

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

*%ProductDir%\%repository%\connectorInstance*

startup.bat のパラメーターとして ICS サーバー名を指定できます。例えば、「start\_Domino.bat connName serverName」と指定します。

- OS/400 プラットフォームの場合:

*/QIBM/UserData/WBIServer43/WebSphereICSName/connectors/connectorInstance*

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

コネクタに、コネクタ固有のメタオブジェクトがある場合、コネクタ・インスタンス用のメタオブジェクトを作成する必要があります。メタオブジェクトをファイルとして保管する場合は、次のディレクトリを作成して、ファイルを */QIBM/UserData/WBIServer43/WebSphereICSName/repository/connectorInstance* に格納します。

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

- Linux プラットフォームの場合:

*%ProductDir%/connectors/connectorInstance*

ここで connectorInstance は、コネクタ・インスタンスを一意的に示します。コネクタに、コネクタ固有のメタオブジェクトがある場合、コネクタ・インスタンス用のメタオブジェクトを作成する必要があります。メタオブジェクトをファイルとして保管する場合は、次のディレクトリを作成して、ファイルを *%ProductDir%/repository/connectorInstance* に格納します。

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

## ビジネス・オブジェクト定義の作成

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

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

- Windows:

*¥%ProductDir%¥repository¥initialConnectorInstance*

作成した追加ファイルは、¥%ProductDir%¥repository の適切なコネクタ・インスタンス・サブディレクトリに置かれていなければなりません。

- OS/400: /QIBM/UserData/WBIServer43/WebSphereICSName/repository /initialConnectorInstance。ここで、WebSphereICSName はコネクタの実行に使用する InterChange Server Express サーバー・インスタンスの名前です。

作成した追加ファイルは、  
/QIBM/UserData/WBIServer43/WebSphereICSName/repository の適切な connectorInstance サブディレクトリ内に存在する必要があります。

- Linux:

*/%ProductDir%/repository/initialConnectorInstance*

作成した追加ファイルは、/%ProductDir%/repository の適切な connectorInstance サブディレクトリに置かれていなければなりません。

## コネクタ定義の作成

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

1. 初期コネクタの構成ファイル (コネクタ定義) をコピーし、名前変更します。

2. 各コネクタ・インスタンスが、サポートされるビジネス・オブジェクト (および関連メタオブジェクト) を正しくリストしていることを確認します。
3. 必要に応じて、コネクタ・プロパティをカスタマイズします。

## 始動スクリプトの作成

始動スクリプトは以下のように作成します。

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

dirname

(Linux の場合のみ) 始動スクリプト CONJAR を  
 CONJAR=\${CONDIR}/CW\${CONNAME}.jar から  
 「CONJAR=\${CONDIR}/BIADominoConnector.jar」に変更する必要があります。

2. この始動スクリプトを、20 ページの『新規ディレクトリーの作成』で作成したコネクタ・ディレクトリーに格納します。
3. (Windows の場合のみ) 始動スクリプトのショートカットを作成します。
4. (Windows の場合のみ) 初期コネクタのショートカット・テキストをコピーし、新規コネクタ・インスタンスの名前に一致するように (コマンド行で) 初期コネクタの名前を変更します。
5. (OS/400 の場合のみ) 次の情報を使用してコネクタのジョブ記述を作成します。CRTDUPOBJ OBJ(QWBIDOMC) FROMLIB(QWBISVR43) OBJTYPE(\*JOB) TOLIB(QWBISVR43) NEWOBJ(newdominoname)

ここで、*newdominoname* は新規 Lotus Domino コネクタのジョブ記述に使用する 10 文字の名前です。

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

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

---

## コネクタの開始

コネクタは、**コネクタ始動スクリプト**を使用して明示的に始動する必要があります。始動スクリプトは、次に示すようなコネクタのランタイム・ディレクトリーに存在していなければなりません。例えば、Windows の場合は以下を使用します。

```
%ProductDir%\connectors\%connName
```

ここで、*connName* はコネクタを示します。始動スクリプトの名前は、表 7 に示すように、オペレーティング・システム・プラットフォームによって異なります。

表 7. コネクタの始動スクリプト

オペレーティング・システム	始動スクリプト
Windows	start_ <i>connName</i> .bat
OS/400	start_ <i>connName</i> .sh



表7. コネクターの始動スクリプト (続き)

オペレーティング・システム	始動スクリプト
Linux	「connector_manager -start connName [-cConfigFile]」により、環境変数が設定され、start_connName.sh 始動スクリプトが自動的に開始されます。始動スクリプトを手動で実行する必要はありません。

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

## 始動スクリプトの起動 (Windows の場合)

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

- System Monitor から

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

- 「スタート」メニューから

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

デフォルトでは、プログラム名は「IBM WebSphere Business Integration Express」となっています。ただし、これはカスタマイズすることができます。あるいは、ご使用のコネクターへのデスクトップ・ショートカットを作成することもできます。

Windows システムでは、Windows サービスとして始動するようにコネクターを構成することができます。この場合、Windows システムがブートしたとき (自動サービスの場合)、または Windows サービス・ウィンドウを通じてサービスを始動したとき (手動サービスの場合) に、コネクターが始動します。

- コマンド行から。

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

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

## 始動スクリプトの起動 (OS/400 の場合)

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

- Windows から

WebSphere Business Integration Server Express Console がインストールされているマシンから、「プログラム」>「IBM WebSphere Business Integration Console」>「コンソール」を選択します。次に、OS/400 システム名または IP アドレスと、

\*JOBCTL 特殊権限を持つユーザー・プロファイルおよびパスワードを指定します。アダプターのリストから `connName` アダプターを選択し、「アダプターを始動」ボタンを選択します。

- OS/400 コマンド行から

**バッチ・モード:** CL コマンド QSH を実行し、QSHELL 環境から以下のコマンドを実行します。

```
/QIBM/ProdData/WBIServer43/bin/submit_adapter.sh connName WebSphereICSName  
pathToConnNameStartScript jobDescriptionName
```

ここで、`connName` はコネクター名であり、`WebSphereICSName` は InterChange Server Express サーバー名 (デフォルトは QWBIDFT)、`pathToConnNameStartScript` はコネクターの始動スクリプトの絶対パス、`jobDescriptionName` は QWBISVR43 ライブラリーで使用するジョブ記述の名前です。

**対話モード:** CL コマンド QSH を実行し、QSHELL 環境から以下のコマンドを実行します。

```
/QIBM/UserData/WBIServer43/WebSphereICSName/connectors  
/connName/start_connName.sh connName WebSphereICSName [-cConfigFile]
```

ここで、`connName` はご使用のコネクターの名前であり、`WebSphereICSName` は Interchange Server Express インスタンスの名前です。

**注:** TCP/IP サーバーを使用して始動するには、コマンド

```
/QIBM/ProdData/WBIServer43/bin/add_autostart_adapter.sh を使用します。このとき、上記で説明したように、パラメーター connName、WebSphereICSName、pathToConnNameStartScript、および jobDescriptionName をこの順序で指定します。
```

## 始動スクリプトの起動 (Linux の場合)

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

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

ここで、`connName` はコネクターの名前であり、`WebSphereICSName` は `WebSphereICS` です。

---

## コネクターの停止

コネクターを停止する方法は、コネクターが始動された方法によって異なります。

## コネクターの停止 (Windows から)

Windows プラットフォームでは、以下の方法でコネクターを停止できます。

- 「コネクター」ウィンドウをアクティブにして、「q」と入力して Enter を押しします。

- コネクタが Windows のサービスとして始動された場合は、コントロール・パネル (「コントロール パネル」 > 「管理ツール」 > 「サービス」 > 「CWConnectorWBIDominoAdapter」) を使用してコネクタを停止できます。

## コネクタの停止 (OS/400 から)

OS/400 プラットフォームでは、以下の方法でコネクタを停止できます。

- コンソールまたはコマンド行から

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

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

- QSHELL から `start_connName.sh` スクリプトを使用してアダプタを始動した場合は、F3 を押してコネクタを終了します。

## コネクタの停止 (Linux から)

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

```
connector_manager connName -stop serverName
```

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



---

## 第 3 章 Lotus Domino アダプター・ビジネス・オブジェクト

本章では Lotus Domino ビジネス・オブジェクトについて説明します。 Lotus Domino アダプターには、すぐに使用できる構成済みビジネス・オブジェクトがあります。このビジネス・オブジェクトを変更する必要はありません。また 7 ページの『第 2 章 アダプターのインストールと構成』で説明するように、アダプターのインストール時に必要なビジネス・オブジェクト定義がインストールされます。

本章では、背景情報としてビジネス・オブジェクトの構造と Lotus Domino ビジネス・オブジェクトについて説明し、アダプターによるビジネス・オブジェクトの処理方法を解説します。

本章は、以下のセクションから構成されています。

- 27 ページの『メタデータの定義』
- 27 ページの『ビジネス・オブジェクト構造の概要』
- 28 ページの『Lotus Domino ビジネス・オブジェクト定義』
- 31 ページの『ビジネス・オブジェクトの生成』

---

### メタデータの定義

Adapter for Lotus Domino はメタデータ主導型です。WebSphere Business Integration システムでは、メタデータは、Lotus Domino によりエクスポートされたアプリケーション固有情報として定義されています。このメタデータには、データ構造が記述されています。メタデータを使用してビジネス・オブジェクト定義が構成されます。このビジネス・オブジェクト定義は、実行時にコネクターがビジネス・オブジェクトを作成するときに使用されます。

メタデータ主導型アダプターは、サポートする各ビジネス・オブジェクトを、ビジネス・オブジェクト定義にエンコードされているメタデータに基づいて処理します。したがって、コードを変更せずに、アダプターが新しいビジネス・オブジェクト定義や変更されたビジネス・オブジェクト定義を処理できます。

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

---

### ビジネス・オブジェクト構造の概要

WebSphere Business Integration システムのビジネス・オブジェクト定義は、次の要素で構成されます。

- 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す。このオブジェクトには、特定のアプリケーションのデータ構造と属性プロパティーが反映されます。

一部の属性はデータを含んでおらず、代わりにデータが含まれている子ビジネス・オブジェクトまたは子ビジネス・オブジェクト配列を示します。親レコードと子レコードのデータは、キーによって関連付けられています。

アダプターのビジネス・オブジェクトは、フラットなものと同層のものがあります。フラットなビジネス・オブジェクトの属性はすべて単純です。つまり、子ビジネス・オブジェクトを指し示すのではなく、単一の値 (ストリングなど) を表します。階層ビジネス・オブジェクトには、単純属性と、属性値が含まれている子ビジネス・オブジェクトまたはその配列の両方が含まれています。

カーディナリティー 1 のコンテナ・オブジェクト、すなわち単一カーディナリティー関係は、親ビジネス・オブジェクト内の属性に単一の子ビジネス・オブジェクトが含まれているときに発生します。この場合、子ビジネス・オブジェクトは 1 つのレコードのみを持つ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・オブジェクトのタイプと同じです。

カーディナリティー  $n$  のコンテナ・オブジェクト、すなわち複数カーディナリティー関係は、親ビジネス・オブジェクト内の属性に子ビジネス・オブジェクト配列が含まれているときに発生します。この場合、子ビジネス・オブジェクトは複数のレコードを持つ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タイプと同じです。

---

## Lotus Domino ビジネス・オブジェクト定義

Lotus Domino アダプターは、メイン・ビジネス・オブジェクト (DominoDocument ビジネス・オブジェクト) を 1 つ使用します。DominoDocument には、このリリースで使用される子ビジネス・オブジェクト (DominoItem ビジネス・オブジェクト) が 1 つ含まれています。DominoItem ビジネス・オブジェクトには、DominoItemValue ビジネス・オブジェクトが子オブジェクトとして含まれています。

**注:** DominoDocument にはもう 1 つの子ビジネス・オブジェクト、DominoEmbeddedObject ビジネス・オブジェクトも定義されていますが、このリリースのアダプターでは使用されません。

DominoDocument ビジネス・オブジェクトは Lotus Domino 文書を表します。文書とは、メール、予定表、タスクなどの Lotus Domino データ・ノートです。DominoItem ビジネス・オブジェクトは、Lotus Domino 文書のコンポーネントである項目を表します。

**注:** アダプターは、今回のリリースではテキスト、日時、および数といった単純な Lotus Domino 項目タイプ (AUTHORS、DATETIMES、NAME、NUMBERS、READERS、および TEXT) のみをサポートします。アダプターは、RICHTEXT Lotus Domino 項目タイプをサポートしません。

アダプターは DominoDocument ビジネス・オブジェクトを介してインバウンド要求で動詞 Create、Update、Delete、Retrieve、および Exists をサポートします。つまり、別のアプリケーションが Lotus Domino Server での新規 Lotus Domino 文書の作成、既存の文書の更新または削除を要求できます。また、アプリケーションへの既存の Lotus Domino 文書の送信を要求し、特定の文書が Lotus Domino Server に存在しているかどうかを確認することもできます。

アダプターは、アウトバウンド・イベントの動詞 Create、Update、および Delete もサポートします。Lotus Domino Server で文書が作成、更新、または削除されると、イベント通知が InterChange Server Express へ送信されます。

図 3 に、Business Object Designer Express に表示される DominoDocument ビジネス・オブジェクトを示します。

Pos	Name	Type	Key	Foreign	Required	Card	Maximum Length	Default	App Spec Info
1	NotedID	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
2	DatabaseName	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3	UNID	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
4	ServerName	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
5	UserName	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
6	Password	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
7	Created	Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
8	FTSearchScore	Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
9	Key	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
10	LastModified	Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
11	NameOfProfile	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
12	ParentDocumentUNID	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
13	Signer	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
14	Size	Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
15	URL	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
16	Verifier	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
17	HasEmbedded	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
18	IsDeleted	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
19	IsProfile	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
20	IsResponse	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
21	IsSentByAgent	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
22	IsSigned	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
23	IsValid	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
24	EmbeddedObjects	DominoEmbeddedObject	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N			
25	Items	DominoItem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N			
26	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
27			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		

図 3. DominoDocument ビジネス・オブジェクト

図 4 に、Business Object Designer Express に表示される子ビジネス・オブジェクト DominoItem を示します。

	Pos	Name	Type	Key	Foreign	Required	Card	Maximum Length	Default
1	1	LastModified	Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
2	2	Name	String	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	
3	3	Type	String	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	
4	4	Values	DominoItemValue	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N		
5	5	IsAuthors	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
6	6	ValueLength	Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
7	7	IsEncrypted	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
8	8	IsNames	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
9	9	IsProtected	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
10	10	IsReaders	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
11	11	IsSigned	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
12	12	IsSummary	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
13	13	DummyKey	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	
14	14	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
15	15			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	

図 4. DominoItem 子ビジネス・オブジェクト

図 5 に、Business Object Designer Express に表示される DominoItemValue ビジネス・オブジェクトを示します。

	Pos	Name	Type	Key	Foreign	Required	Card	Maximum Length	Default	App Spec Info
1	1	Value	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
2	2	DummyKey	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3	3	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
4	4			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		

図 5. DominoItemValue 子ビジネス・オブジェクト

子ビジネス・オブジェクトを除き、すべての DominoDocument ビジネス・オブジェクトの属性にはカーディナリティー 1 の関係があります。子ビジネス・オブジェクト DominoItem には、 $n$  カーディナリティーの関係があります。つまり、アダプターは任意の数の従属文書項目が含まれている Lotus Domino 文書を処理します。

DominoItem ビジネス・オブジェクト内の子ビジネス・オブジェクト

DominoItemValue も  $n$  カーディナリティー関係を持ちます。つまり、1 つの項目に複数の値を関連付けることができます。(例えば、E メールに複数の受信者を指定することがあります。)

Lotus Domino ビジネス・オブジェクト属性の定義については、本書の『付録 C. Lotus Domino ビジネス・オブジェクト属性』を参照してください。



---

## ビジネス・オブジェクトの生成

実行時には、Lotus Domino Server で該当するイベントが発生するたびに (つまり、Lotus Domino 文書の作成、更新、削除が行われるたびに)、コネクターが Lotus Domino からイベント・データを取得します。(3 ページの『コネクターの動作方法』で、このプロセスを詳しく説明します。) 次にコネクターは、ビジネス・オブジェクト定義にデータをマップし、特定のビジネス・オブジェクト・インスタンス (単純に「ビジネス・オブジェクト」と呼ばれます) を作成します。処理のため、このビジネス・オブジェクトを InterChange Server Express に送信します。これにより、別のアプリケーションに対しイベント情報が使用可能になります。また、コネクターは InterChange Server Express からビジネス・オブジェクトを受信します。InterChange Server Express から受信する別のアプリケーションのビジネス・オブジェクトは、Lotus Domino による Lotus Domino Server での文書の作成、更新、または削除、文書の存在の確認、あるいは Lotus Domino 文書全体を InterChange Server Express を介して要求元のアプリケーションへ戻すことを要求します。



---

## 第 4 章 トラブルシューティングおよびエラー処理

本章には、Adapter for Lotus Domino によるエラー処理方法について説明します。アダプターは、イベントの状況について報告し、エラー・メッセージまたはトレース・メッセージを生成します。本章では、これらのメッセージとトラブルシューティングのヒントを収録しています。本章は、以下のセクションから構成されています。

- 『イベント状況』
- 33 ページの『エラー・メッセージ』
- 39 ページの『トレース』

---

### イベント状況

イベント・ビューおよびイベント表のアーカイブ・ビューで、イベントの状況を調べることができます。これはイベント処理で発生した問題を判別する際に役立ちます。イベント・ビューでは、イベント状況は「Ready for Poll」または「In-progress」のいずれかとなっています。アーカイブ・ビューでは、イベント状況は「Success」、「Unsubscribed」、「Error Processing Event」、「Error Posting Event」、または「Error Object Not Found」です。

エラーを解決した後またはトラブルシューティングの目的で、以下のようしてイベントの状況を変更することができます。

1. Lotus Notes Client からイベント表アーカイブ・ビュー (EventArchiveView) または イベント・ビュー (EventTableView) に移動します。
2. 状況を変更したいイベントを選択します。
3. 右マウス・ボタンをクリックし、「編集」を選択します。
4. ドロップダウン・ボックスから、変更後の状況を選択します。
5. 「ファイル」>「保存」を選択します。

イベントの状況が変更され、該当するイベント表ビューに表示されます。

---

### エラー・メッセージ

エラー・メッセージは STDOUT または指定したファイルに記録されます。ログ・ファイルの指定方法は、使用する InterChange Server Express によって異なります。メッセージの記録先を変更する方法については、「システム・インプリメンテーション・ガイド」の資料を参照してください。

アダプターにより生成される一般的なエラーと推奨される解決方法を 34 ページの表 8 に示します。この表の「xxx」と「yyy」は、実際に生成されるメッセージのストリングを表します。

表 8. 一般的なアダプター・エラーと推奨される解決方法

エラー番号	エラー・メッセージ	説明/推奨される解決方法
4001	例外が xxx でキャッチされました。: yyy (Exception caught in xxx: yyy)	例外がメソッド xxx でキャッチされました。例外のテキストは yyy に示され、メッセージ ID と NotesException クラスから抽出したテキストを含んでいます。
4006	構成プロパティー xxx の値がありません。 デフォルト値 yyy が使用されます。(No value for configuration property xxx. Using default value : yyy)	必要に応じて、Connector Configurator Express を使用してプロパティーに適切な値を設定します。
4008	構成プロパティー xxx の値がありません。 (No value for configuration property xxx.)	必要に応じて、Connector Configurator Express を使用してプロパティーに適切な値を設定します。
4034	文書 BO を InterChange Server Express へ配信できませんでした。イベント番号 xxx、状況 yyy (Failed to deliver the Document BO to InterChange Server Express event # xxx, status yyy)	この状況は、障害の原因を示します。問題を訂正すると、状況を「ready for poll」に変更することができます。
4036	試行された BO 配信のサブスクリプションが見つかりません (イベント番号 xxx、状況 yyy)。 (No Subscription found for the attempted BO delivery, event # xxx, status yyy.)	コラボレーションとコネクターが開始していることを確認します。
4049	NoteID xxx の文書の検索に失敗しました (イベント番号 yyy)。(Failed to Retrieve Document with NoteID : xxx, event # yyy)	指定のイベントで指定された文書の検索時に問題が発生しました。指定されているイベントの発生後に文書が削除されている可能性があります。
4050	イベント・レコード (番号: xxx) の検索に失敗しました。(Failed to Retrieve Event Record # : xxx)	BIALD_EventTable.nsf データベースが使用可能であることと、コネクター構成に指定されているユーザーにデータベースの文書を読み取る権限が付与されていることを確認します。
4051	ポーリングできるイベントが EventTable にありません。(No more events to poll from the EventTable.)	イベント表にイベントがリストされていません。処置は不要です。
4052	イベント・ビュー xxx の検索に失敗しました。(Failed to retrieve the Event View : xxx)	BIALD_EventTable.nsf データベースのイベント表ビューの読み取り時に問題が発生しました。 BIALD_EventTable.nsf データベースが使用可能であることと、Connector Configurator Express に指定されているユーザーにデータベースの文書を読み取る権限が付与されていることを確認します。
4059	認識できない項目タイプです。(Item Type is not recognized.)	アダプターにより処理されていた文書に、サポートされていない項目タイプがありました。この項目は DominoDocument ビジネス・オブジェクトには現れません。サポートされるアイテム・タイプについては、『Lotus Domino ビジネス・オブジェクト定義』を参照してください。

表 8. 一般的なアダプター・エラーと推奨される解決方法 (続き)

エラー番号	エラー・メッセージ	説明/推奨される解決方法
4064	Lotus Domino イベントに対しデータベース xxx をオープンできませんでした (文書 NoteID: yyy、イベント番号 zzz)。 (Failed to open Database : xxx for Domino event, Document NoteID= yyy, event # zzz)	データベースが使用可能であることと、コネクタ構成に指定されているユーザーにデータベースの文書を読み取る権限が付与されていることを確認します。
4072	IN_PROGRESS イベントが検出されました (イベント番号 xxx)。エラーをログに記録するようにコネクタが構成されています。 (Encountered an IN_PROGRESS Event, event number # xxx. Connector has been configured to Log Error.)	コネクタ構成に基づいて、In Progress イベントがエラーとしてログに記録されます。構成されているアクションが正しくない場合には、Connector Configurator Express を使用して更新してください。
4073	IN_PROGRESS イベントが検出されました (イベント番号 xxx)。致命的エラーをログに記録するようにコネクタが構成されています。 (Encountered an IN_PROGRESS Event, event number # xxx. Connector has been configured to Log Fatal Error.)	コネクタ構成に基づいて、In Progress イベントが致命的エラーとしてログに記録されます。構成されているアクションが正しくない場合には、Connector Configurator Express を使用して更新してください
4074	IN_PROGRESS イベントが検出されました。このイベントを無視するようにコネクタが構成されています。 (Encountered an IN_PROGRESS Event. Connector has been configured to Ignore Event.)	コネクタ構成に基づき、In-Progress イベントは無視されます。構成されているアクションが正しくない場合には、Connector Configurator Express を使用して更新してください。
4075	IN_PROGRESS イベントが検出されました。このイベントを再処理するようにコネクタが構成されています。 (Encountered an IN_PROGRESS Event. Connector has been configured to Reprocess Event.)	コネクタ構成に基づいて、In Progress イベントが再処理されます。構成されているアクションが正しくない場合には、Connector Configurator Express を使用して更新してください。
4076	メソッド xxx でセッション検証に失敗しました。 (Failed to validate session in method xxx.)	Lotus Domino Server が開始していることを確認します。また、コネクタ構成でサーバー、ユーザー名、およびパスワードが正しく構成されていることを確認します。
4077	メソッド xxx でイベント表データベース/ビューへの接続確立に失敗しました。 (Failed to establish connection to event table database/view in method xxx.)	Lotus Domino Server が開始していることを確認します。また、コネクタ構成でサーバー、ユーザー名、およびパスワードが正しく構成されていることを確認します。
4081	イベント番号 xxx の文書データベース・パスが無効です。 (Invalid document database path for event # xxx)	このメッセージは、構成されているパスにデータベースが存在しないことを示します。イベント表へのイベント送信後、アダプターがそのイベントの処理を試みる前に、データベースが削除されていた可能性があります。
4084	Lotus Domino セッションが無効です。 (Domino Session is no longer valid.)	Lotus Domino Server が開始していることを確認します。また、コネクタ構成またはビジネス・オブジェクトでサーバー、ユーザー名、およびパスワードが正しく構成されていることを確認します。

表 8. 一般的なアダプター・エラーと推奨される解決方法 (続き)

エラー番号	エラー・メッセージ	説明/推奨される解決方法
5001	Lotus Domino Server へのアクセス中にエラーが発生しました (サーバー: xxx, ユーザー名: yyy, エラー・テキスト: zzz)。 (Error accessing Domino server Server: xxx Username: yyy Error text: zzz)	Lotus Domino Server へのアクセス中にエラーが発生しました。サーバーが開始していることを確認します。
5002	データベース xxx へのアクセス中にエラーが発生しました。エラー・テキスト: yyy (Error accessing database: xxx. Error text: yyy)	構成されているユーザーに、データベースへのアクセス権限が付与されていることを確認します。
5003	データベース xxx が見つかりません。 (Database xxx not found.)	ビジネス・オブジェクトに指定されているデータベースが見つかりませんでした。ビジネス・オブジェクトに指定されているデータベースのパスと名前を確認してください。
5005	データベース xxx のオープン中にエラーが発生しました。エラー・テキスト: yyy (Error opening database: xxx Error text: yyy)	詳細については、エラー・テキストを参照してください。構成されているユーザーに、データベースへのアクセス権限が付与されていることを確認します。
5006	動詞: xxx がサポートされていません。 (Unsupported verb: xxx)	コネクターが、着信ビジネス・オブジェクトで指定されている動詞をサポートしていません。サポートされる動詞は、Create、Retrieve、Update、Delete、および Exist です。
5007	文書のオープン中にエラーが発生しました (NoteID: xxx, エラー・テキスト: yyy)。 (Error opening document. NoteID=xxx. Error text: yyy)	ビジネス・オブジェクト・ハンドラーは、指定されたデータベースに存在しない NoteID を持つビジネス・オブジェクトを受信しました。この NoteID を持つ文書は削除された可能性があります。NoteID が正しくない場合は、それを訂正し、ビジネス・オブジェクトを再送します。
5008	文書のオープン中にエラーが発生しました (NoteID: xxx)。データベースに文書がありません。 (Error opening document. NoteID=xxx. Document not found in database.)	データベースで、指定の NoteID を持つ文書が見つかりませんでした。
5009	文書の作成中にエラーが発生しました。エラー・テキスト: xxx (Error creating document. Error text: xxx)	指定されたデータベース内で文書を作成するために必要な権限が、ユーザーに付与されていない可能性があります。権限レベルを変更するか、または異なるユーザー・ログイン情報を使用してください。
5010	文書の NoteID の検索中にエラーが発生しました。 (Error retrieving NoteID from Document.)	Create 動詞の処理で、データベース内に作成された文書の NoteID を検索中にエラーが発生しました。この NoteID はビジネス・オブジェクトには追加されません。NoteID が必要な場合には、Notes Client を使用して文書を参照してください。
5011	文書の削除中にエラーが発生しました (NoteID: xxx, エラー・テキスト: yyy)。 (Error deleting document. NoteID=xxx. Error text: yyy)	ユーザーに文書を削除する権限が付与されていない可能性があります。権限を変更するか、または異なるユーザー・ログイン情報を使用してください。

表 8. 一般的なアダプター・エラーと推奨される解決方法 (続き)

エラー番号	エラー・メッセージ	説明/推奨される解決方法
5012	文書の検証中にエラーが発生しました (NoteID: xxx, エラー・テキスト: yyy)。 (Error validating document. NoteID=xxx. Error text: yyy)	指定された文書の検証中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。
5013	文書の保管中にエラーが発生しました。エラー・テキスト: xxx (Error saving document. Error text: xxx)	説明: 指定された文書の保管中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。
5014	文書は保管されませんでした。(The Document was not saved.)	原因: 要求、または ForceSave プロパティーと MakeResponse プロパティーの設定により文書が変更されませんでした。文書が保管されませんでした。これは、文書が変更されなかったか、またはコネクター・プロパティーの設定に基づいています。コネクターのアプリケーション固有のプロパティー (ForceSave および MakeResponse) を必要に応じて設定します。詳しくは、18 ページの表 5 または Lotus Domino の資料を参照してください。
5015	ビジネス・オブジェクト yyy に属性 xxx がありません。(Attribute xxx not found in Business Object yyy.)	指定の必須属性がビジネス・オブジェクトで見つかりませんでした。使用されているビジネス・オブジェクトのバージョンが正しいことを確認してください。
5016	属性 xxx は、ビジネス・オブジェクト yyy の String 型属性ではありません。(Attribute xxx is not String type in Business Object yyy)	アダプターにより String 型属性が予期されていましたが、指定された属性が、指定のビジネス・オブジェクトの String 型属性ではありません。使用されているビジネス・オブジェクトのバージョンが正しいことを確認してください。
5017	属性 xxx は、ビジネス・オブジェクト yyy の Boolean 型属性ではありません。(Attribute xxx is not boolean type in Business Object yyy)	アダプターにより Boolean 型属性が予期されていましたが、指定された属性が、指定のビジネス・オブジェクトの Boolean 型属性ではありません。使用されているビジネス・オブジェクトのバージョンが正しいことを確認してください。
5018	属性 xxx は、ビジネス・オブジェクト yyy の Business Object 型属性ではありません。(Attribute xxx is not Business Object type in Business Object yyy)	アダプターにより Business Object 型属性が予期されていましたが、指定された属性が、指定のビジネス・オブジェクトの Business Object 型属性ではありません。使用されているビジネス・オブジェクトのバージョンが正しいことを確認してください。
5019	項目 xxx を文書へ追加中にエラーが発生しました。エラー・テキスト: yyy (Error adding Item xxx to document. Error text: yyy)	ユーザーに権限が付与されていない可能性があります。権限レベルを変更するか、または異なるユーザー・ログイン情報を使用してください。
5020	文書の親データベースの検索中にエラーが発生しました。エラー・テキスト: yyy (Error retrieving Document parent Database. Error text: yyy)	文書の親データベースの検索中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。
5021	データベースでの現行セッションの検索中にエラーが発生しました。エラー・テキスト: xxx (Error retrieving current Session from the Database. Error text: xxx)	データベースでの現行セッションの検索中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。

表 8. 一般的なアダプター・エラーと推奨される解決方法 (続き)

エラー番号	エラー・メッセージ	説明/推奨される解決方法
5022	ストリング xxx から DateTime オブジェクトの作成中にエラーが発生しました。エラー・テキスト: yyy (Error creating DateTime object from the String xxx. Error text: yyy)	説明: 指定されたストリングから DateTime オブジェクトの作成中にエラーが発生しました。日付ストリングは、java.text.SimpleDateFormat API で指定されているパターンに準拠する必要があります。詳細については、アプリケーション固有の構成プロパティを参照してください。
5023	型 xxx に対して ItemHandler が定義されていません。有効な型は「Double」、 「String」 および「Date」です。(No ItemHandler defined for type: xxx. Valid types include “Double”, “String,” and “Date.”)	DominoItem.type 属性に指定されている型に対して ItemHandler が定義されていません。DominoItem.type 属性にサポートされている型が設定されていることを確認してください。
5024	項目 yyy でフラグ xxx の設定中にエラーが発生しました。例外テキスト: zzz (Error setting flag xxx in Item yyy. Exception text: zzz)	指定の項目で指定されたフラグを設定中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。
5025	ストリング xxx を Double オブジェクトへ変換中にエラーが発生しました。(Error converting String xxx to a Double.)	指定されたストリングを Double オブジェクトへ変換中にエラーが発生しました。DominoItemValue.value が着信ビジネス・オブジェクトでの有効な Double 値であることを確認してください。
5026	Lotus Domino 文書をビジネス・オブジェクトへ変換中にエラーが発生しました (NoteID: xxx, エラー・テキスト: yyy)。 (Error converting Domino Document to a Business Object. NoteID=xxx Error text: yyy)	文書をビジネス・オブジェクトへ変換中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。
5029	文書からアイテム xxx の除去中にエラーが発生しました (NoteID: yyy, エラー・テキスト: zzz)。 (Error removing Item xxx from Document. NoteID=yyy Error text: zzz)	文書から指定の項目を除去中にエラーが発生しました。詳しくは、エラー・テキストを参照してください。
5030	ビジネス・オブジェクト属性の初期化および検証中にエラーが発生しました (ビジネス・オブジェクト名: xxx, エラー・テキスト: yyy)。 (Error initializing and validating Business Object attributes. BO Name: xxx Error text: yyy)	IsRequired が Yes に設定されている属性について、着信ビジネス・オブジェクト内で値が指定されていることを確認してください。
5031	サブ動詞 xxx は、DominoItem 名 yyy でサポートされていないサブ動詞です。(Unsupported subverb: xxx for DominoItem Name=yyy)	指定されているサブ動詞はコネクタによりサポートされていません。DominoItem ビジネス・オブジェクトに対し、サポートされているサブ動詞 (Create、Update、または Delete) が指定されていることを確認してください。



表 8. 一般的なアダプター・エラーと推奨される解決方法 (続き)

エラー番号	エラー・メッセージ	説明/推奨される解決方法
5032	ロケールのデフォルト形式を使用して日付ストリングを解析中にエラーが発生しました (日付: xxx、ロケール: yyy)。 (Error parsing Date String using default format for locale. Date: xxx Locale: yyy)	指定のロケールのシステム・デフォルト形式を使用して指定の日付ストリングを解析中にエラーが発生しました。日付ストリングとビジネス・オブジェクトのロケールのシステム・デフォルト形式が一致しているかどうかを確認してください。一致していない場合には、コネクターのアプリケーション固有プロパティ <code>DateFormats</code> に必要な日付形式を指定してください。詳しくは、『アプリケーション固有の構成プロパティ』を参照してください。
5033	構成されている形式を使用して日付ストリングを解析中にエラーが発生しました (日付: xxx、ロケール: yyy、形式: zzz)。 (Error parsing Date String using configured format. Date: xxx Locale:yyy Format: zzz.)	指定の形式とロケールを使用して指定の日付ストリングを解析中にエラーが発生しました。日付ストリングが、コネクターのアプリケーション固有プロパティ <code>DateFormats</code> に指定されている形式の 1 つに一致しているかどうかを確認してください。詳しくは、『アプリケーション固有の構成プロパティ』を参照してください。
5034	構成されている日付形式ストリングを使用して <code>Date</code> オブジェクトを生成できませんでした (日付: xxx、ロケール: yyy)。 (Unable to generate Date object using configured Date Format strings. Date: xxx Locale: yyy.)	エラーを記録する前にすべての日付形式を使用した生成操作が試行されます。 <code>Date</code> 生成操作がすべて失敗しているかどうかを確認してください。失敗している場合には、必要な日付形式をコネクターのアプリケーション固有プロパティ <code>DateFormats</code> に指定します。詳しくは、『アプリケーション固有の構成プロパティ』を参照してください。
5035	ビジネス・オブジェクトに <code>NoteID</code> が指定されていませんでした。 (The NoteID was not specified in the Business Object.)	着信ビジネス・オブジェクトに <code>NoteID</code> が常に指定されていることを確認します。
5038	<code>Name</code> 属性に指定されている値が無効です (Name=xxx、BO=yyy)。 (Invalid value specified for Name attribute. Name=xxx, BO=yyy.)	この属性に <code>CxIgnore</code> 値または <code>CxBlank</code> 値が指定されていないことを確認します。

## トレース

トレースは、コネクターの動作を詳細に追跡するために使用できるオプション・デバッグ機能です。トレース・メッセージは、デフォルトでは `STDOUT` に書き込まれます。トレース・プロパティは、標準構成プロパティである `AgentTraceLevel`、`TraceFileName`、および `ControllerTraceLevel` を用いて設定されます。トレース・メッセージの構成について詳しくは、41 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。

表 9 に、トレース・レベル別の受信する情報のタイプを示します。

表 9. トレース・メッセージの内容

レベル	説明
レベル 0	コネクターのバージョンを識別します。このレベルでは、他のトレースは実行されません。

表9. トレース・メッセージの内容 (続き)

レベル	説明
レベル 1	<ul style="list-style-type: none"> <li>• 状況情報を示します。</li> <li>• 処理された各ビジネス・オブジェクトの主要情報を示します。</li> <li>• ポーリングが発生するたびに記録します。</li> </ul>
レベル 2	<ul style="list-style-type: none"> <li>• コネクタにより処理される各オブジェクトに使用されるビジネス・オブジェクト・ハンドラーを示します。</li> <li>• ビジネス・オブジェクトが InterChange Server Express に通知されるたびに、ログに記録します。</li> <li>• 要求ビジネス・オブジェクトを受信するたびにオブジェクトの受信を示します。</li> </ul>
レベル 3	<ul style="list-style-type: none"> <li>• 必要に応じて、処理される外部キーを識別します。このレベルのメッセージは、コネクタがビジネス・オブジェクトで外部キーを検出した場合、またはコネクタがビジネス・オブジェクトに外部キーを設定した場合に表示されます。</li> <li>• ビジネス・オブジェクトの処理。例えば、ビジネス・オブジェクト間の一致の検出、子ビジネス・オブジェクト配列でのビジネス・オブジェクトの検出などがあります。</li> </ul>
レベル 4	<ul style="list-style-type: none"> <li>• アプリケーション固有情報を示します。この情報には、ビジネス・オブジェクトのアプリケーション固有情報フィールドを処理するメソッドにより戻される値などがあります。</li> <li>• コネクタが関数を開始または終了した時期を示します。このレベルのメッセージは、コネクタのプロセス・フローをトレースするときに役立ちます。</li> <li>• スレッド固有の処理をすべて記録します。例えば、コネクタが複数のスレッドを作成した場合、メッセージには新しいスレッドの作成がすべて記録されます。</li> </ul>
レベル 5	<ul style="list-style-type: none"> <li>• コネクタ初期化を示します。このレベルのメッセージには、InterChange Server Express から検出された Connector Configurator Express プロパティの値などを含めることができます。</li> <li>• コネクタが作成した各スレッドの、実行中の状態の詳細を示します。</li> <li>• アプリケーションで実行されるステートメントを示します。コネクタのログ・ファイルには、ターゲット・アプリケーションで実行されるすべてのステートメントと、必要に応じて置換される変数の値が記述されます。</li> <li>• ビジネス・オブジェクト・ダンプを記録します。コネクタは、オブジェクトの処理開始前のビジネス・オブジェクト (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) を示すテキストと、処理完了後のビジネス・オブジェクト (コネクタがコラボレーションに戻したオブジェクト) を示すテキストを作成します。</li> </ul>

---

## 付録 A. コネクタの標準構成プロパティ

この付録では、WebSphere InterChange Server Express で動作する、WebSphere Business Integration Server Express のアダプターに含まれるコネクタ・コンポーネントの標準構成プロパティについて説明します。

コネクタによっては、一部の標準プロパティが使用されないことがあります。Connector Configurator Express から統合ブローカーを選択すると、ご使用のアダプターに対して構成する必要がある標準プロパティのリストが表示されます。

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

---

### 標準コネクタ・プロパティの構成

アダプター・コネクタには 2 つのタイプの構成プロパティがあります。

- 標準構成プロパティ
- コネクタ固有のプロパティ

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

### Connector Configurator Express の使用

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

### プロパティ値の設定と更新

プロパティ・フィールドのデフォルトの長さは 255 文字です。

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

1. デフォルト
2. リポジトリ
3. ローカル構成ファイル
4. コマンド行

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

- **動的**  
変更を System Manager に保管すると、変更が即時に有効になります。
- **コンポーネント再始動**  
System Manager でコネクターを停止してから再始動しなければ、変更が有効になりません。アプリケーション固有コンポーネントまたは統合ブローカーを停止、再始動する必要はありません。
- **サーバー再始動**  
アプリケーション固有のコンポーネントおよび統合ブローカーを停止して再始動しなければ、変更が有効になりません。
- **エージェント再始動**  
アプリケーション固有のコンポーネントを停止して再始動しなければ、変更が有効になりません。

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

## 標準プロパティの要約

表 10 は、標準コネクター構成プロパティの早見表です。標準プロパティの依存関係は RepositoryDirectory に基づいているため、コネクターによっては使用されないプロパティがあり、使用する統合ブローカーによってプロパティの設定が異なる可能性があります。

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

表 10. 標準構成プロパティの要約

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
AdminInQueue	有効な JMS キュー名	CONNECTORNAME /ADMININQUEUE	コンポーネント再始動	Delivery Transport は JMS
AdminOutQueue	有効な JMS キュー名	CONNECTORNAME/ADMINOUTQUEUE	コンポーネント再始動	Delivery Transport は JMS
AgentConnections	1 から 4	1	コンポーネント再始動	Delivery Transport は IDL
AgentTraceLevel	0 から 5	0	動的	
ApplicationName	アプリケーション名	コネクター・アプリケーション名として指定された値	コンポーネント再始動	
BrokerType	ICS	ICS		
CharacterEncoding	ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437 注: これは、サポートされる値の一部です。	ascii7	コンポーネント再始動	
ConcurrentEventTriggeredFlows	1 から 32,767	1	コンポーネント再始動	Repository Directory は <REMOTE>

表 10. 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
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	IDL または JMS	IDL	コンポーネント再始動	
DuplicateEventElimination	true または false	false	コンポーネント再始動	JMS トランスポートのみ、Container Managed Events は <NONE> でなければならない
EnableOidForFlowMonitoring	true または false	false	コンポーネント再始動	
FaultQueue		CONNECTORNAME/FAULTQUEUE	コンポーネント再始動	JMS トランスポートのみ
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	JMS トランスポートのみ
jms.MessageBrokerName	crossworlds.queue.manager	crossworlds.queue.manager	コンポーネント再始動	JMS トランスポートのみ
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	JMS トランスポートのみ
jms.Password	任意の有効なパスワード		コンポーネント再始動	JMS トランスポートのみ
jms.UserName	任意の有効な名前		コンポーネント再始動	JMS トランスポートのみ
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	Repository Directory は <REMOTE>

表 10. 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
Locale	en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DEit_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	
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE>
MessageFileName	パスまたはファイル名	InterchangeSystem.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	CONNECTORNAME/MONITORQUEUE	コンポーネント再始動	JMS トランスポートのみ: DuplicateEvent Elimination は true でなければならない。
OADAutoRestartAgent	true または false	false	動的	Repository Directory は <REMOTE>
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE>
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE>
PollEndTime	HH:MM (HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒)  no (ポーリングを使用不可にする)  key (コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力された場合にのみポーリングする)	10000	動的	
PollQuantity	1 から 500	1	エージェント再始動	JMS トランスポートのみ: Container Managed Events を指定
PollStartTime	HH:MM(HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
RequestQueue	メタデータ・リポジトリの場所		エージェント再始動	<REMOTE> に設定する

表 10. 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 は JMS
RestartRetryCount	0 から 99	3	動的	
RestartRetryCount	適切な正数 (単位: 分): 1 から 2147483547	1	動的	
SourceQueue	有効な JMS キュー名	CONNECTORNAME/SOURCEQUEUE	エージェント再始動	Delivery Transport が JMS であり、かつ Container Managed Events が指定されている場合のみ
SynchronousRequestQueue	有効な JMS キュー名	CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
SynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	Delivery Transport は JMS
SynchronousResponseQueue	有効な JMS キュー名	CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
WireFormat	CwB0	CwB0	エージェント再始動	

## 標準構成プロパティ

このセクションでは、各標準コネクタ構成プロパティの定義を示します。

### AdminInQueue

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

デフォルト値は CONNECTORNAME/ADMININQUEUE です。

### AdminOutQueue

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

デフォルト値は CONNECTORNAME/ADMINOUTQUEUE です。

### AgentConnections

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

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

## AgentTraceLevel

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

## ApplicationName

コネクタ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を指定します。ICS を指定する必要があります。

## CharacterEncoding

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

**注:** Java ベースのコネクタでは、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、本書の `Connector Configurator Express` の使用方法に関する付録を参照してください。

## ConcurrentEventTriggeredFlows

コネクタ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、順次に行われる単一スレッド処理であるコネクタのポーリングでは無効です。

## ContainerManagedEvents

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

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

デフォルト値は `No value` です。

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

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

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

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

## ControllerStoreAndForwardMode

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

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

ます。アプリケーション固有のコンポーネントが作動可能になると、コネクター・コントローラーはアプリケーション固有のコンポーネントにその要求を転送します。

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

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

デフォルト値は `true` です。

## ControllerTraceLevel

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

## DeliveryQueue

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

コネクターから `WebSphere InterChange Server Express` 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、`IDL` (`CORBA IIOP`) または `JMS` (`Java Messaging Service`) です。デフォルトは `IDL` です。

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

### JMS

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

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

**重要:** `WebSphere InterChange Server Express` で動作しているコネクターで `JMS` トランスポート機構を使用すると、メモリー制限が発生することがあります。

この環境では、`WebSphere MQ` クライアント内でメモリーが使用されるため、(サーバー側の) コネクター・コントローラーと (クライアント側の) コネクターの両方を始動するのは困難な場合があります。

## DuplicateEventElimination

このプロパティを `true` に設定すると、JMS 対応コネクターによるデリバリー・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、コネクターに対し、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。これはコネクター開発時に設定されます。

このプロパティは、`false` に設定することもできます。

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

## EnableOidForFlowMonitoring

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

デフォルト値は `false` です。

## FaultQueue

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

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

## JvmMaxHeapSize

エージェントの最大ヒープ・サイズ (メガバイト単位)。

デフォルト値は `128M` です。

## JvmMaxNativeStackSize

エージェントの最大ネイティブ・スタック・サイズ (キロバイト単位)。

デフォルト値は `128K` です。

## JvmMinHeapSize

エージェントの最小ヒープ・サイズ (メガバイト単位)。

デフォルト値は `1M` です。

## jms.FactoryClassName

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

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

## jms.MessageBrokerName

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

デフォルト値は `crossworlds.queue.manager` です。

## jms.NumConcurrentRequests

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

デフォルト値は 10 です。

## jms.Password

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

デフォルトはありません。

## jms.UserName

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

デフォルトはありません。

## Locale

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

ロケール名は、次の書式で指定します。

`ll_TT.codeset`

ここで、以下のように説明されます。

<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、本書の `Connector Configurator Express` の使用方法に関する付録を参照してください。

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

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

## LogAtInterchangeEnd

統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。ブローカーのログ宛先にログを記録すると、E 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 E 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、E メール・メッセージが送信されます。デフォルト値は false です。

## MaxEventCapacity

コントローラー・バッファ内のイベントの最大数。このプロパティは、フロー制御で使用されます。

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

## MessageFileName

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

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

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

## MonitorQueue

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

デフォルト値は CONNECTORNAME/MONITORQUEUE です。

## OADAutoRestartAgent

コネクタが自動再始動およびリモート再始動機能を使用するかどうかを指定します。この機能では、MQ により起動される Object Activation Daemon (OAD) を使用

して、異常シャットダウン後にコネクタを再始動したり、System Monitor からリモート・コネクタを始動したりします。

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

デフォルト値は `false` です。

## OADMaxNumRetry

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

デフォルト値は 1000 です。

## OADRetryTimeInterval

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

デフォルト値は 10 です。

## PollEndTime

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

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

## PollFrequency

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

- ポーリング・アクション間のミリ秒数。
- ワード `key`。コネクタは、コネクタのコマンド・プロンプト・ウィンドウで文字 `p` が入力されたとき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

WebSphere InterChange Server Express からコネクタへビジネス・オブジェクトが送信されるときに使用されるキューです。

デフォルト値は `CONNECTOR/REQUESTQUEUE` です。

## RepositoryDirectory

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

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

## ResponseQueue

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

`JMS` 応答キューを指定します。`JMS` 応答キューは、応答メッセージをコネクタ・フレームワークから統合ブローカーへデリバリーします。WebSphere InterChange Server Express は、要求を送信した後、`JMS` 応答キューで応答メッセージを待機します。

## RestartRetryCount

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

デフォルト値は 3 です。

## RestartRetryInterval

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

デフォルト値は 1 です。

## SourceQueue

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

JMS イベント・ストアを使用する JMS 対応コネクタでの保証付きイベント・デリバリーをサポートするコネクタ・フレームワークに、JMS ソース・キューを指定します。詳細については、47 ページの『ContainerManagedEvents』を参照してください。

デフォルト値は CONNECTOR/SOURCEQUEUE です。

## SynchronousRequestQueue

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルトは CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE です。

## SynchronousResponseQueue

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルトは CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE です。

## SynchronousRequestTimeout

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルト値は 0 です。



## WireFormat

トランスポートのメッセージ・フォーマットです。設定値は CwB0 です。



---

## 付録 B. Connector Configurator Express

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

この付録では、次のトピックについて説明します。

- 57 ページの『Connector Configurator Express の概要』
- 58 ページの『Connector Configurator Express の始動』
- 59 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』
- 61 ページの『新規構成ファイルの作成』
- 64 ページの『構成ファイル・プロパティの設定』
- 71 ページの『グローバル化環境における Connector Configurator Express の使用』

---

### Connector Configurator Express の概要

Connector Configurator Express では、WebSphere InterChange Server Express で使用するアダプターのコネクタ・コンポーネントを構成できます。

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

- コネクタを構成するための**コネクタ固有のプロパティ・テンプレート**を作成する。
- **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、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。

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

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

標準プロパティの範囲は、ブローカーと構成によって異なる可能性があります。特定のプロパティに特定の値が設定されている場合にのみ使用できるプロパティがあります。Connector Configurator Express の「標準のプロパティ」ウィンドウには、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、59 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

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

---

## Connector Configurator Express の始動

以下の 2 種類のモードで Connector Configurator Express 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 Express の実行

どのブローカーを実行している場合にも、Connector Configurator Express 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 Business Integration Server Express」>「Toolset Express」>「開発」>「Connector Configurator Express」をクリックします。
- 「ファイル」>「新規」>「構成ファイル」を選択します。

Connector Configurator Express を個別に実行して構成ファイルを生成してから、System Manager に接続してこの構成ファイルを System Manager プロジェクトに保存することもできます (64 ページの『構成ファイルの完成』を参照)。

---

## System Manager からの Configurator Express の実行

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

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

1. System Manager を開きます。
2. 「System Manager」ウィンドウで、「統合コンポーネント・ライブラリー」アイコンを展開し、「コネクタ」を強調表示します。
3. System Manager メニュー・バーから、「ツール」>「Connector Configurator Express」をクリックします。「Connector Configurator Express」ウィンドウが開き、「新規コネクタ」ダイアログ・ボックスが表示されます。

既存の構成ファイルを編集するには、以下のステップを実行します。

1. 「System Manager」ウィンドウの「コネクタ」フォルダーでいずれかの構成ファイルを選択し、右クリックします。

2. 「標準のプロパティ」タブをクリックし、この構成ファイルに含まれているプロパティを確認します。

---

## コネクタ固有のプロパティ・テンプレートの作成

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

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

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

### 新規テンプレートの作成

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

テンプレートは以下のように作成します。

1. 「ファイル」>「新規」>「コネクタ固有プロパティ・テンプレート」をクリックします。
2. 以下のフィールドを含む「コネクタ固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。
  - 「テンプレート」、「名前」

このテンプレートが使用されるコネクタ（またはコネクタのタイプ）を表す固有の名前を入力します。テンプレートから新規構成ファイルを作成するためのダイアログ・ボックスを開くと、この名前が再度表示されます。

- 「旧テンプレート」、「変更する既存のテンプレートを選択してください」

「テンプレート名」表示に、現在使用可能なすべてのテンプレートの名前が表示されます。

- テンプレートに含まれているコネクタ固有のプロパティ定義を調べるには、「テンプレート名」表示でそのテンプレートの名前を選択します。そのテンプレートに含まれているプロパティ定義のリストが「テンプレートのプレビュー」表示に表示されます。テンプレートを作成するときには、ご使用のコネクタに必要なプロパティ定義に類似したプロパティ定義が含まれている既存のテンプレートを使用できます。
3. 「テンプレート名」表示からテンプレートを選択し、その名前を「名前の検索」フィールドに入力し（または「テンプレート名」で自分の選択項目を強調表示し）、「次へ」をクリックします。

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

## 一般特性の指定

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

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

プロパティの一般特性の選択を終えたら、「値」タブをクリックします。

## 値の指定

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

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

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

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

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

値が作成されて、グリッドに表示されると、そのテーブルの表示内から編集できるようになります。テーブルにある既存の値の変更を行うには、その行の行番号

をクリックして行全体を選択します。次に「値」フィールドを右マウス・ボタンでクリックし、「値の編集 (Edit Value)」をクリックします。

## 依存関係の設定

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

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

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

== (等しい)

!= (等しくない)

> (より大)

< (より小)

>= (より大か等しい)

<= (より小か等しい)

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

---

## 新規構成ファイルの作成

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

### コネクター固有のテンプレートからの構成ファイルの作成

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

1. 「ファイル」>「新規」>「コネクター構成」をクリックします。

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

- **名前**

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

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

- **システム接続**

デフォルトのブローカーは ICS です。この値は変更できません。

- **コネクタ固有プロパティ・テンプレートを選択**

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

使用するテンプレートを選択し、「**OK**」をクリックします。

- 構成しているコネクタの構成画面が表示されます。タイトル・バーに統合ブローカーとコネクタの名前が表示されます。ここですべてのフィールドに値を入力して定義を完了するか、ファイルを保管して後でフィールドに値を入力するかを選択できます。

- ファイルを保管するには、「**ファイル**」>「**保管**」>「**ファイルに**」をクリックするか、「**ファイル**」>「**保管**」>「**プロジェクトに**」をクリックします。プロジェクトに保管するには、System Manager が実行中でなければなりません。ファイルとして保管する場合は、「**ファイル・コネクタを保管**」ダイアログ・ボックスが表示されます。\*.cfg をファイル・タイプとして選択し、「**ファイル名**」フィールド内に名前が正しいスペル (大文字と小文字の区別を含む) で表示されていることを確認してから、ファイルを保管するディレクトリーにナビゲートし、「**保管**」をクリックします。Connector Configurator Express のメッセージ・パネルの状況表示に、構成ファイルが正常に作成されたことが示されます。

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

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

---

## 既存ファイルの使用

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

使用可能な既存ファイルは、以下の 1 つまたは複数の形式になります。



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

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

既存ファイルを使用してコネクタを構成するには、Connector Configurator Express 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、ディレクトリから `*.txt`、`*.cfg` または `*.in` ファイルを開きます。

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

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. ディレクトリ表示内で、適切なコネクタ定義ファイルへ移動し、ファイルを選択し、「開く」をクリックします。

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

1. System Manager を始動します。System Manager が開始されている場合にのみ、構成を System Manager から開いたり、System Manager に保管したりできます。
2. Connector Configurator Express を始動します。

3. 「ファイル」>「開く」>「プロジェクトから」をクリックします。

---

## 構成ファイルの完成

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

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

- 65 ページの『標準コネクター・プロパティの設定』
- 65 ページの『アプリケーション固有の構成プロパティの設定』
- 66 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』
- 68 ページの『関連付けられたマップ』
- 69 ページの『トレース/ログ・ファイル値の設定』

**注:** コネクターが JMS メッセージングを使用するものである場合、データをビジネス・オブジェクトに変換するデータ・ハンドラーを構成できるように、追加のカテゴリが表示されることがあります。詳細については、70 ページの『データ・ハンドラー』を参照してください。

---

## 構成ファイル・プロパティの設定

新規のコネクター構成ファイルを作成して名前を付けるとき、または既存のコネクター構成ファイルを開くときには、Connector Configurator Express 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。

標準プロパティとコネクター固有プロパティの違いは、以下のとおりです。

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

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

- 背景がグレーのフィールドは、標準のプロパティを表します。値を変更することはできますが、名前の変更およびプロパティの除去はできません。

- 背景が白のフィールドは、アプリケーション固有のプロパティを表します。これらのプロパティは、アプリケーションまたはコネクタの特定のニーズによって異なります。値の変更も、これらのプロパティの除去も可能です。
- 「値」フィールドは構成可能です。
- 各プロパティごとに表示される「更新メソッド」は、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。

## 標準コネクタ・プロパティの設定

標準のプロパティの値を変更するには、以下の手順を実行します。

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

## アプリケーション固有の構成プロパティの設定

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

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

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

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

## コネクター・プロパティの暗号化

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

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

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

## 更新メソッド

付録 A 『コネクターの標準構成プロパティ』の 41 ページの『プロパティ値の設定と更新』にある更新メソッドの説明を参照してください。

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

## サポートされるビジネス・オブジェクト定義の指定

コネクターで使用するビジネス・オブジェクトを指定するには、Connector Configurator Express の「サポートされているビジネス・オブジェクト」タブを使用します。汎用ビジネス・オブジェクトと、アプリケーション固有のビジネス・オブジェクトの両方を指定する必要があり、またそれらのビジネス・オブジェクト間のマップの関連を指定することが必要です。

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

**注:** コネクターによっては、アプリケーションでイベント通知や (メタオブジェクトを使用した) 追加の構成を実行するために、特定のビジネス・オブジェクトをサポートされているものとして指定することが必要な場合もあります。詳細につ

いては、本書のビジネス・オブジェクトに関する章と、「ビジネス・オブジェクト開発ガイド」を参照してください。

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

## ビジネス・オブジェクト名

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

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

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

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

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

## エージェント・サポート

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

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

ビジネス・オブジェクトがコネクター・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 Express」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。

## 関連付けられたマップ

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

- **関連付けられたマップ**

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

- **明示的**

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

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

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

以下の手順を実行して、マップを明示的にバインドします。

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

## リソース

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

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

## トレース/ログ・ファイル値の設定

コネクター構成ファイルまたはコネクター定義ファイルを開くと、Connector Configurator Express は、そのファイル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 Express 内でこれらの値を変更できます。

ログとトレースの値を変更するには、以下の手順を実行します。

1. 「トレース/ログ・ファイル」タブをクリックします。

2. ログとトレースのどちらでも、以下のいずれかまたは両方へのメッセージの書き込みを選択できます。

- コンソールに (STDOUT):  
ログ・メッセージまたはトレース・メッセージを STDOUT ディスプレイに書き込みます。

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

- ファイルに:  
ログ・メッセージまたはトレース・メッセージを指定されたファイルに書き込みます。ファイルを指定するには、ディレクトリー・ボタン (省略符号) をクリックし、指定する格納場所へ移動し、ファイル名を指定し、「保管」をクリックします。(コネクターが、Connector Configurator Express をインストールした Windows プラットフォームで実行されていない場合は、最初に、システム上のファイルの格納場所にドライブをマップする必要があります。)ログ・メッセージまたはトレース・メッセージは、指定した場所の指定したファイルに書き込まれます。

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

## データ・ハンドラー

データ・ハンドラー・セクションの構成が使用可能となるのは、DeliveryTransport の値に JMS を、また ContainerManagedEvents の値に JMS を指定した場合のみです。このタブは、アダプターが保証付きイベント・デリバリーを利用するものである場合に使用可能になります。

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

---

## 構成ファイルの保管

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

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

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



System Manager でのプロジェクトの使用方法和、配置の詳細については、IBM WebSphere Business Integration Server Express の「ユーザース・ガイド」を参照してください。

---

## 構成の完了

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

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

---

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

Connector Configurator Express 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 Express では、ネイティブなエンコード方式を使用しています。構成ファイルに書き込む場合は UTF-8 エンコード方式を使用します。

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

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

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

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

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



## 付録 C. Lotus Domino ビジネス・オブジェクト属性

この付録では、DominoDocument ビジネス・オブジェクト、DominoItem 子ビジネス・オブジェクト、および DominoItemValue 子ビジネス・オブジェクトの属性の定義を示します。

### DominoDocument ビジネス・オブジェクト属性

属性名	型	カーディナリティ	説明
NoteID	String	1	文書の NoteID。
DatabaseName	String	1	文書が含まれているデータベースの名前。
UNID	String	1	文書のユニバーサル ID。
ServerName	String	1	Lotus Domino Server の名前。ブランクの場合、ServerName 構成プロパティの値が使用されます。
UserName	String	1	ユーザー名。アダプターが Lotus Domino Server とのセッションを確立するときに、このユーザー名が使用されます。ブランクの場合、UserName 構成プロパティの値が使用されます。
Password	String	1	ユーザー・パスワード。アダプターが Lotus Domino Server とのセッションを確立するときに、このパスワードが使用されます。ブランクの場合、Password 構成プロパティの値が使用されます。
Created	Date	1	文書作成日時。この日付は、DateFormats 構成プロパティに指定されている日付形式の 1 つに従っている必要があります。
FTSearchScore	Integer	1	全文検索で文書を検索した場合の関連性の値。
Key	String	1	プロファイル文書のプロファイル・キー。
LastModified	Date	1	文書の最終更新日時。この日付は、DateFormats 構成プロパティに指定されている日付形式の 1 つに従っている必要があります。
NameOfProfile	String	1	プロファイル文書のプロファイル名。
ParentDocumentUNID	String	1	返答文書の場合の親文書のユニバーサル ID。
Signer	String	1	署名付き文書の場合の署名者。
Size	Integer	1	添付を含む文書のサイズ (バイト単位)。
URL	String	1	文書の親オブジェクトの Lotus Domino URL。
Verifier	String	1	署名付き文書の場合の署名を検証する証明書の名前。
HasEmbedded	Boolean	1	文書に埋め込みオブジェクトがある場合は true。
IsDeleted	Boolean	1	文書が削除されている場合は true。
IsProfile	Boolean	1	文書がプロファイル文書の場合は true。
IsResponse	Boolean	1	文書が返答文書の場合は true。

属性名	型	カーディナリティー	説明
IsSentByAgent	Boolean	1	文書がプログラムによりメール送信された場合は true。
IsSigned	Boolean	1	署名付き文書の場合は true。
IsValid	Boolean	1	既存の文書の場合 (削除スタブではない場合) には true。
EmbeddedObjects	DominoEmbedded Object	n	このリリースでは使用されません。
Items	DominoItem	n	文書項目を表す子ビジネス・オブジェクト。
ObjectEventID	String	1	ビジネス・オブジェクト、要求、および応答をトラッキングするために Websphere Business Integration システムが生成した値。

## Dominoltem ビジネス・オブジェクト属性

Dominoltem は、DominoDocument ビジネス・オブジェクトの子です。

属性名	型	カーディナリティー	説明
LastModified	Date	1	文書の最終更新日時。
Name	String	1	項目の名前。必須。
Type	String	1	項目タイプ Item.AUTHORS、Item.DATETIMES、Item.NAMES、Item.NUMBERS、Item.READERS、または Item.TEXT のいずれか。必須。
Values	DominoItemValue	n	子ビジネス・オブジェクト。
ValueLength	Integer	1	項目の長さ (バイト単位)。
isAuthors	Boolean	1	項目のタイプが AUTHORS の場合は true。
isEncrypted	Boolean	1	項目が暗号化されている場合は true。
isNames	Boolean	1	項目のタイプが NAMES の場合は true。
isProtected	Boolean	1	項目を変更するためにエディターへのアクセス権が必要な場合は true。
isReaders	Boolean	1	項目のタイプが READERS の場合は true。
isSigned	Boolean	1	署名付き文書の場合は true。
isSummary	Boolean	1	項目値をビューに表示できる場合は true。
ObjectEventID	String	1	ビジネス・オブジェクト、要求、および応答をトラッキングするために Websphere Business Integration システムが生成した値。

## DominoltemValue ビジネス・オブジェクト属性

DominoltemValue は、DominoItem ビジネス・オブジェクトの子です。

属性名	型	カーディナリティ	説明
Value	String	1	項目の値。
ObjectEventID	String	1	ビジネス・オブジェクト、要求、および応答をトラッキングするために Websphere Business Integration システムが生成した値。



# 索引

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

## [ア行]

アーカイブ・ビュー 3  
アーキテクチャー, アダプター 1  
アダプター, 構成 17  
アダプターのアーキテクチャー 1  
アダプターのインストール 7  
アダプターの構成 7  
アダプターの使用方法 1  
アダプター・フレームワーク, 定義 1  
アプリケーション固有の構成プロパティ 17  
アプリケーション固有のコンポーネント 2  
イベント, モニター対象として選択 16  
イベント状況, イベント表での 33  
イベント状況, 検討 33  
イベント処理, ステップ 3  
イベント表 2, 3  
    イベント状況 33  
イベント表, 構成 14  
イベント表, コピー 13  
イベント表, Lotus Domino Server にコピー 13  
イベント表の構成 14  
イベント表ファイル 13  
イベント・タイプ 16  
イベント・ビュー 3  
イベント・リスナー 2, 3  
イベント・リスナー, Lotus Domino Server にコピー 13  
インストール, アダプターの 7  
インストール, 概要 9  
インストールのステップ 9  
エラー, 報告 33  
エラー処理 33  
エラー・メッセージ 33  
エラー・メッセージ, ロギング 33

## [カ行]

カーディナリティー, ビジネス・オブジェクト内の 28  
概要, アダプターの  
    紹介, アダプターの 1

関連文書 v  
規則, 表記上の vi  
構成 16  
    アダプター 17  
    イベント表 14  
    イベント・タイプ 16  
    コネクタ 17  
    モニターするデータベース 15  
DIIOP 12  
HTTP 11  
IIOP 12  
Lotus Domino Internet Interoperability Protocol 12  
Lotus Domino Server 上の  
    Java/Javascript/COM 12  
Lotus Domino Server の  
    ApplicationUserName 12  
    NOTES.INI 12  
構成, アダプターの 7  
構成ビュー 3  
構成プロパティ 17  
    ApplicationPassword 18  
    DocumentBOName 18  
    DominoServerName 18  
    EventDBName 18  
    ForceDelete 18  
    ForceSave 18  
    InDoubtEvents 18  
    MakeResponse 18  
構成プロパティ, アプリケーション固有の 17  
構成プロパティ, 標準 17  
コネクタ, 構成 17  
コネクタ, 動作方法  
    コネクタの動作方法 3  
コネクタ・コンポーネント 2  
コネクタ・アーキテクチャー 2  
コネクタ・フレームワーク 2  
    フレームワーク, コネクタ 2  
コピー, イベント表 13  
コンポーネント, アプリケーション固有の 2

## [サ行]

サポートされる動詞 29  
始動スクリプト, 変更 16  
状況情報  
    状況, イベント  
        イベント状況 3  
使用のシナリオ 1

使用例 1  
処理, イベント 3  
処理, 要求 5  
資料, 関連 v  
選択, データベース 15  
属性, ビジネス・オブジェクトの 27, 73

## [タ行]

データベース, 選択 15  
ディレクトリ, Lotus Domino Server 上の 13  
統合ブローカー, インストール 8  
動詞, サポートされる 29  
トラブルシューティング 33  
トレース 39  
トレース・メッセージ 39  
トレース・レベル, 定義 39

## [ハ行]

ビジネス・オブジェクト  
    属性 73  
        DominoDocument 28  
        DominoItem 28  
        DominoItemValue 28  
ビジネス・オブジェクト, 子 28  
ビジネス・オブジェクト, 処理 2  
ビジネス・オブジェクト, 生成 31  
ビジネス・オブジェクト, 説明 27  
ビジネス・オブジェクト, Lotus Domino  
    アダプター用 27  
ビジネス・オブジェクト構造 27  
ビジネス・オブジェクト・ハンドラー  
    ハンドラー, ビジネス・オブジェクト 2  
ビュー  
    アーカイブ, イベント表での 3  
    イベント, イベント表での 3  
    構成, イベント表での 3  
ビュー, イベント表 3  
ビュー, イベント表のデータの 3  
表記上の規則 vi  
ファイル, イベント表の 13  
ファイル, イベント表のためにコピー 13  
ファイル, Lotus Domino Server にコピーされる 13  
ファイル, Lotus Domino Server にコピーする 13  
ファイルのコピー, Lotus Domino Server への 13

ブローカー、統合  
インストール 8  
プロパティ、構成 17  
プロパティ、構成、アプリケーション固  
有の 17  
文書、関連 v

## [マ行]

メタデータ、定義 27  
メッセージ、エラー 33  
メッセージ、トレース 39

## [ヤ行]

要求の処理、ステップ 5

## A

ApplicationPassword 18  
ApplicationUserName、Lotus Domino  
Server で構成 12

## D

DocumentBOName 18  
DominoDocument ビジネス・オブジェクト  
28  
DominoItem 28  
DominoItemValue 28  
DominoServerName 18

## E

EventDBName 18

## F

ForceDelete 18  
ForceSave 18

## H

HTTP、構成 11

## I

IIOP、構成 12  
InDoubtEvents 18  
InDoubtEvents、値 19  
Internet Interoperability Protocol、構成  
12

## J

Java/Javascript/COM の構成 12

## L

Lotus Domino API 16  
Lotus Domino Objects for Java 16  
Lotus Domino Server、アダプターの構成  
11  
Lotus Domino Server、インストール 8  
Lotus Domino Server、コピーするファイ  
ル 13  
Lotus Domino Server、変更 11  
Lotus Domino 管理クライアント、変更を  
加える 11  
Lotus Domino 項目タイプ、サポートされ  
る 28  
Lotus Domino、バージョン 1

## M

MakeResponse 18

## N

NCSO.jar 16  
NCSO.jar、および始動スクリプト 16  
NOTES.INI の構成 12

## P

ProductDir、定義 vi

## W

WBIA、インストール 9  
WebSphere Business Integration Adapters、  
インストール 9



---

## 特記事項

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

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

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

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

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

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

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

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

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

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

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

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

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

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

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

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

#### 著作権使用許諾

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

#### プログラミング・インターフェース情報

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

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

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

**注:** 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

## 商標

以下は、IBM Corporation の商標です。

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

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

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

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

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

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

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



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





Printed in Japan