

**IBM WebSphere Business Integration Server and
Server Express**



Adapter for Lotus Domino ユーザーズ・ガイド

アダプター・バージョン 1.3.x

**IBM WebSphere Business Integration Server and
Server Express**



Adapter for Lotus Domino ユーザーズ・ガイド

アダプター・バージョン 1.3.x

お願い

本書および本書で紹介する製品をご使用になる前に、91 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM WebSphere Business Integration Adapter for Lotus Domino (製品番号 5724-H20) バージョン 1.3.x に適用されます。

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： WebSphere Business Integration Server and Server Express
Adapter for Lotus Domino User Guide
Adapter Version 1.3.x

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.8

この文書では、平成明朝体™W3、平成明朝体™W7、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™W7、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2004, 2005. All rights reserved.

© Copyright IBM Japan 2005

目次

本書について	v
本書の内容	v
対象読者	v
関連文書	v
表記上の規則	v
本リリースの新機能	vii
リリース 1.3.x の新機能	vii
第 1 章 アダプターの概要	1
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
第 2 章 アダプターのインストールと構成	7
インストール・プロセスの概要	7
アダプター環境	8
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
コネクターの始動	21
コネクターの停止	22
複数のコネクター・インスタンスの作成	23
第 3 章 Lotus Domino アダプター・ビジネス・オブジェクト	27
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
第 4 章 トラブルシューティングおよびエラー処理	33
イベント状況	33
エラー・メッセージ	33
トレース	39
ビジネス・オブジェクト・サイズの制限	40
付録 A. コネクターの標準構成プロパティ	41
新規プロパティ	41
標準コネクター・プロパティの概要	41
標準プロパティの早見表	43
標準プロパティ	49
付録 B. Connector Configurator Express	65
Connector Configurator Express の概要	65
Connector Configurator Express の始動	66
System Manager からの Configurator の実行	67
コネクター固有のプロパティ・テンプレートの作成	67
新規構成ファイルの作成	70
既存ファイルの使用	72

構成ファイルの完成	73
構成ファイル・プロパティの設定	74
構成ファイルの保管	82
構成ファイルの変更	82
構成の完了	83
グローバル化環境における Connector Configurator Express の使用	83
Lotus Domino ビジネス・オブジェクト属性	85
DominoDocument ビジネス・オブジェクト属性	85
DominoItem ビジネス・オブジェクト属性	86
DominoItemValue ビジネス・オブジェクト属性	87
索引	89
特記事項	91
プログラミング・インターフェース情報	92
商標	93

本書について

製品 IBM^(R)WebSphere Business Integration Server Express および IBM^(R) 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 Collaboration Foundation 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。

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 および IBM WebSphere Business Integration 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。

本書の内容

本書では、IBM WebSphere Business Integration Adapter for *NAME* のインストーラー、コネクタ・プロパティ構成、ビジネス・オブジェクト開発、およびトラブルシューティングについて説明します。

対象読者

本書は、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 のどのインストーラー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。

表記上の規則

本書では、以下のような規則を使用しています。

courier フォント	コマンド名、ファイル名、入力情報、システムが画面に出力した情報などのリテラル値を示します。
イタリック、イタリック	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	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・パラメーターです。
...	構文の記述行の場合、省略符号 ... は直前のパラメーターが繰り返されることを示します。例えば、option[,...] は、複数のオプションをコンマで区切って指定できることを意味します。
< >	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、<server_name><connector_name>tmp.log 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: IBM¥WebSphereServer i5/OS: /QIBM/ProdData/WBIServer44/product Linux: /home/\${username}/IBM/WebSphereServer
/, ¥	本書では、ディレクトリー・パスに円記号 (¥) を使用します。Linux および i5/OS システムの場合には、円記号をスラッシュ (/) に置き換えてください。すべての製品パス名は、ご使用のシステム上で Connector for Lotus Domino がインストールされているディレクトリーを基準とした相対パス名です。

本リリースの新機能

リリース 1.3.x の新機能

本書は、以下の項目のサポートを含むように更新されました。

- Adapter Framework バージョン 2.6
- ApplicationUserName および ApplicationPassword プロパティ用の双方向スク립ト・データ
- IBM JRE/JDK 1.4.2
- Linux:
 - RedHat Enterprise Linux WS/AS/ES 3.0 Update 2、Intel (IA32)
 - SuSE Linux ES 8.1 SP3、Intel (IA32)
 - SuSE Linux ES 9.0、Intel (IA32)
- Windows 2003 (Standard Edition または Enterprise Edition)
- IBM i5/OS V5R3 および IBM OS/400 V5R2

注: 明示されない限り、i5/OS は OS/400 および i5/OS を意味します。

第 1 章 アダプターの概要

本章では、IBM WebSphere Business Integration 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 は統合ブローカー 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 R5 (5.0.3 以上) および R6 (6.5 以前) を統合できます。ただしこれよりも前のバージョンの Lotus Domino との互換性はありません。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。

統合ブローカー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エンタープライズ・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エンタープライズ・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

統合ブローカーと 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 と統合ブローカー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す。
- 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 と統合ブローカー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統合ブローカーに送信します。また、このコンポーネントは統合ブローカー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、統合ブローカーとアプリケーション固有のコンポーネントの間の中継を行います。

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
- 統合ブローカー要求の処理
- 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。統合ブローカー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統合ブローカー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、統合ブローカー (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・オブジェクトを統合ブローカーに送信します。

着信ビジネス・オブジェクト (統合ブローカー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、統合ブローカーからビジネス・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統合ブローカー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、次のように統合ブローカーへ渡されます。

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・オブジェクトを統合ブローカーに送信します。
7. 他のアプリケーションが統合ブローカーの Lotus Domino イベント情報を要求する場合には、統合ブローカー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。(この処理は、使用する統合ブローカーによって異なります。詳細については、ご使用の統合ブローカーの資料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。

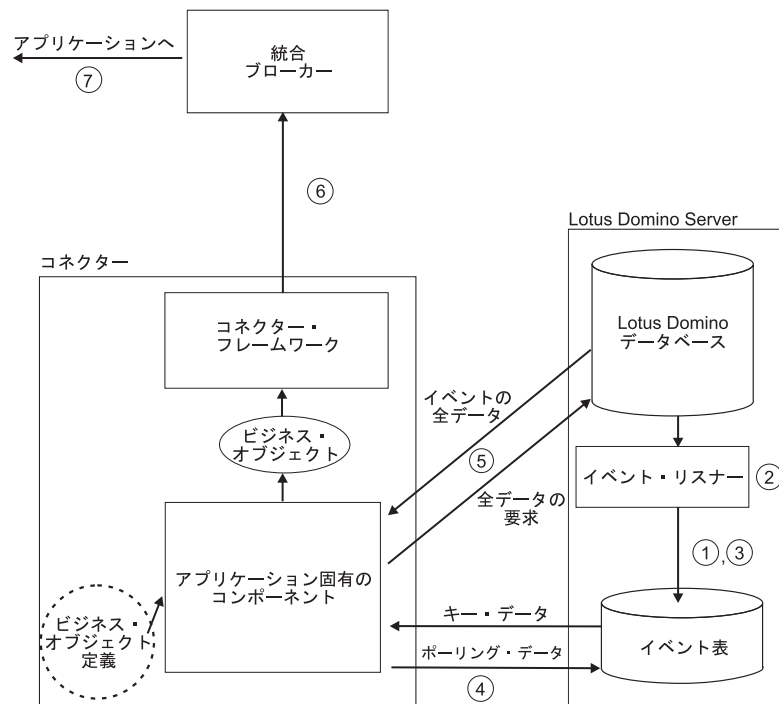


図 1. Lotus Domino アダプターのイベント処理

要求の処理

統合ブローカー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. 統合ブローカーはビジネス・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. 統合ブローカーにより 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、Lotus 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。)

- 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) を統合ブローカーに送信します。次に統合ブローカーは、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。

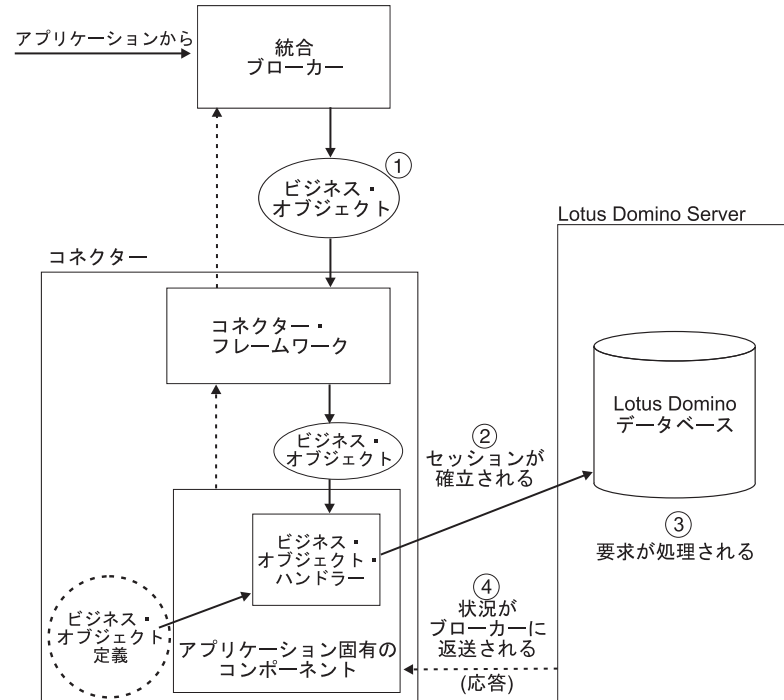


図 2. Lotus Domino アダプターの要求処理

第 2 章 アダプターのインストールと構成

本章では、IBM WebSphere Business Integration Adapter 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。

- 7 ページの『インストール・プロセスの概要』
- 8 ページの『アダプター環境』
- 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』
- 21 ページの『コネクターの始動』
- 22 ページの『コネクターの停止』
- 23 ページの『複数のコネクター・インスタンスの作成』

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. 統合ブローカーをインストールおよび構成します。統合ブローカーは、Lotus Domino とアダプターをインストールする同一のコンピューター、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。

注: Lotus Domino は、Lotus Domino Adapter をインストールする前に i5/OS にインストールしてください。

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。

アダプター環境

アダプターをインストール、構成、および使用する前に、以下のセクションで説明するような環境の要件を把握しておく必要があります。

- 8 ページの『ソフトウェア前提条件』
- 9 ページの『ロケール依存データの処理』

ソフトウェア前提条件

Lotus Domino アダプターをインストールする前に、アダプター・フレームワークに加え、以下のソフトウェアをインストールする必要があります。

オペレーティング・システム

アダプターをインストールするシステム上には、以下のオペレーティング・システムのいずれかがインストールされていなければなりません。

注: すべてのオペレーティング・システム環境には、カスタム・アダプターのコンパイルのために、Java コンパイラー (Windows 2000 の場合は IBM JDK 1.4.2) が必要です。

- Windows 2003 (Standard Edition または Enterprise Edition)
- Linux:
 - RedHat Enterprise Linux WS/AS/ES 3.0 Update 2、Intel (IA32)
 - SuSE Linux ES 8.1 SP3、Intel (IA32)
 - SuSE Linux ES 9.0、Intel (IA32)
- IBM i5/OS V5R3 および IBM OS/400 V5R2

アプリケーション・ソフトウェア

Lotus Domino Server バージョン R5 (5.0.3 以降) および R6 (6.5 以前) がシステムにインストールされている必要があります。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。

注: アダプターのイベント・リスナー・コンポーネントが、Lotus Domino Server にインストールされている必要があります。本リリースでは、イベント・リスナーは Linux Domino Server でサポートしていません。イベント・リスナーでは、オペレーティング・システムとして AIX (5.1 および 5.2) Solaris 9.0 のどちらもサポートされており、Lotus Domino Server マシンにインストールする必要があります。

その他のソフトウェア

- IBM JRE、バージョン 1.4.2
- JDK、バージョン 1.4.2
- Java 2 Runtime Environment (JRE) Standard Edition バージョン 1.3.1 以降

アダプターは、Lotus Domino Server、バージョン R5 (5.0.3 以降) および R6 (6.5 以前) との統合をサポートします。詳細については、Windows 版、Linux 版、または i5/OS 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。

ロケール依存データの処理

コネクタは国際化されています。2 バイト文字セット (DBCS) を、同じく 2 バイト文字セットをサポートするインターフェースに送信することができ、また指定された言語のメッセージ・テキストの送信をサポートします。コネクタは、ある文字コードを使用する場所から異なるコード・セットを使用する場所へデータを転送するときに、文字変換を実行してデータの意味を保持します。

Java 仮想マシン (JVM) 内の Java ランタイム環境は、Unicode 文字コード・セットでデータを表現します。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 システム内のほとんどのコンポーネントは、Java で書かれています。そのため、統合コンポーネント間でデータを転送する際に、ほとんどの場合文字変換は必要ありません。

このアダプターは、アラビア語、ヘブライ語、ウルドゥー語、ペルシア語、およびイディッシュ語などの、双方向スクリプト・データの処理をサポートします。双方向機能を使用するには、双方向標準プロパティを構成する必要があります。詳細については、『付録 A. コネクタの標準構成プロパティ』を参照してください。

アダプターおよび関連ファイルのインストール

WebSphere Business Integration Server Express アダプター製品のインストールの詳細については、「WebSphere Business Integration Server Express インストール・ガイド」(Windows 版、Linux 版、OS/400 および i5/OS 版) を参照してください。このガイドは、WebSphere Business Integration Server Express Adapters Infocenter (<http://www.ibm.com/websphere/wbiserverexpress/infocenter>) にあります。

インストール済みファイルの構造

以下のセクションでは、インストール後の製品のパスおよびファイル名について説明します。製品パス名はすべて、ご使用のシステム上に製品がインストールされているディレクトリーからの相対パスです。デフォルト・パスは、使用するオペレーティング・システムと統合ブローカーによって決まります。

ユーティリティーにより、コネクタ・エージェントが `ProductDir¥connectors¥Domino` ディレクトリーにインストールされ、コネクタ・エージェントのショートカットが「スタート」メニューに追加されます。

以下の表に、Windows、i5/OS、および Linux にインストール済みのファイル構造を示します。インストーラーを使用してコネクタのインストールを選択すると、ファイル構造は自動的にインストールされます。実例のインストール、構成、および実行の説明については、65 ページの『付録 B. Connector Configurator Express』を参照してください。

ファイルがシステムにロードされたら、13 ページの『Lotus Domino Server へのファイルのコピー』の説明に従って、いくつかのファイルを Lotus Domino Server にコピーする必要があります。

インストール済みファイルの構造 (Windows の場合)

表 1. インストール済みの Windows ファイル構造

%ProductDir% のサブディレクトリー	説明
connectors¥Domino	次の始動スクリプトが入っています。 コネクタ・サービスの始動スクリプト start_Domino.bat および start_DOMINO_service.bat。 BIADominoConnector.jar ファイルも格納されています。
connectors¥Domino¥dependencies	次のファイルが送られます。 BIALD_EventTable.nsf BIALD_emptytrash.gif BIALD_movetotrash.gif BIALD_newconfig.gif、nbiald.dll
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、「プログラム」>「IBM WebSphere Business Integration Server Express」>「アダプター」>「コネクタ」を選択して表示可能なコネクタ・ファイルのアイコンが、インストーラーによって追加されます。コネクタを迅速に始動するには、デスクトップからこのファイルへのショートカットを作成します。

ファイルがシステムにロードされたら、13 ページの『Lotus Domino Server へのファイルのコピー』の説明に従って、いくつかのファイルを Lotus Domino Server にコピーする必要があります。

インストール済みファイルの構造 (i5/OS の場合)

表 2. コネクタ用としてインストールされた i5/OS ファイル構造

{ProductDir} のサブディレクトリー	説明
connectors/Domino	BIADominoConnect.jar および startDomino.sh ファイルと、サブディレクトリー依存関係が格納されています。
connectors/Domino/dependencies	次のファイルが送られます。 BIALD_EventTable.nsf BIALD_emptytrash.gif BIALD_movetotrash.gif BIALD_newconfig.gif および libbiald.so
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。

表 2. コネクタ用としてインストールされた i5/OS ファイル構造 (続き)

{ProductDir} のサブディレクトリー	説明
bin/Data/App	BIA_Domino.xsd ファイルが格納されています。

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	BIADominoConnect.jar ファイルが格納されています。
connectors/Domino/dependencies	次のファイルが送られます。 BIALD_EventTable.nsf BIALD_emptytrash.gif BIALD_movetotrash.gif BIALD_newconfig.gif および libbiald.so
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。

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 Administrator 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 NCS0.jar ファイルへアクセスできるように設定します。

Lotus Domino Administrator 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 Administrator を用いて行います (詳細については、Lotus Domino の管理者

用資料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。詳細については、18 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 (Programmability Restrictions)」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。また該当するものがあれば、以下の項目について、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。
 - 制限付き Java/Javascript/COM を実行

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

i5/OS の場合: EXTMGR_ADDINS=biald

WRKDOMSVR を実行するようにアダプターを構成するには、biald を EXTMGR_ADDINS 設定に追加します。

注: 追加の設定値はコンマで分離され、例えば「EXTMGR_ADDINS=decsext,biald」のようになります。

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す。

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)
次のいずれか 1 つをコピーします。		
Lotus Domino Server ルート	libbiald.so	イベント・リスナー (i5/OS システム用)。このファイルは、i5/OS 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 (Linux システム用)。このファイルは、Linux で Lotus Domino Server を実行している場合にのみコピーします。

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

コピー先ディレクトリー	ファイル	説明
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。(Windows では、Lotus Domino データ・ディレクトリーのデフォルトは `c:\Lotus\Domino\Data` です。Linux ではデフォルト値がないので、状況に応じて異なるディレクトリー名が使用されます。)

BIALD_EventTable.nsf

BIALD_emptytrash.gif

BIALD_movetotrash.gif

BIALD_newconfig.gif

i5/OS 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 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。

i5/OS システムの場合: libbiald.so

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:
`:ADDLINK OBJ('/QSYS.LIB/QWBISVR44.LIB/LIBBIALD.SVRPGM')`
`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。実行のために構成された後、このファイルを `/opt/lotus/notes/latest/linux` にコピーします。

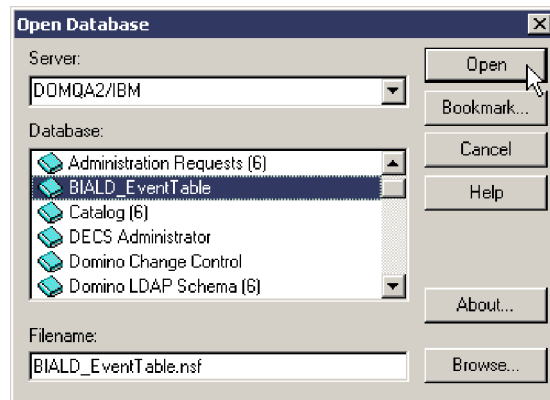
注: イベント・リスナーは Linux Domino Server ではサポートされません。

イベント表の構成

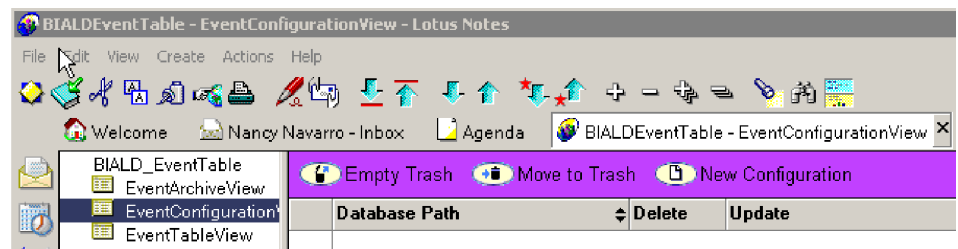
アダプターのイベント・リスナーが、Lotus Domino Server で発生したイベントをモニターできるようにするには、アダプターのイベント表を構成する必要があります。イベント表は、Lotus Notes Client からオープンできる Lotus Domino データベース・ファイルです。

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。

WBI Adapter for Lotus Domino ConfigurationTable:

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

- 「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、*Lotus Domino server root directory/Data* を基準にした相対パスです。例えば、データベース名が Buyer、パスが *Lotus 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 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

例えば次のように選択すると、アダプターが create イベント、update イベント、および delete イベントをモニターするようになります。

WBI Adapter for Lotus Domino ConfigurationTable:

Event Info	Value
Database Path:	/BusApp/Buyer
Event Type:	<input checked="" type="checkbox"/> Create <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> 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には、次の例に示すように、Create だけでなく Update も選択する必要があります。

WBI Adapter for Lotus Domino ConfigurationTable:

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

- 「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 Lotus Notes Client で提供されています。) NCSO.jar には、Lotus Domino API (Lotus 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 Lotus Notes Client のディレクトリーの名前です。

注: Linux の場合は、上記のパス名の中の円記号をスラッシュ (/) に置き換えてください。

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。

Linux: 次の行

```
DOMINOJAVACLASSES=/server1/domino/java  
CON_SPEC_JAR_ONE=${DOMINOJAVACLASSES}/NCSO.jar
```

を次のように変更します。

```
DOMINOJAVACLASSES=directoryname  
CON_SPEC_JAR_ONE=${DOMINOJAVACLASSES}/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。

Windows: 次の行

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

を次のように変更します。

```
set DOMINOCLASSES=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。

i5/OS の場合:

デフォルト設定は、DOMINOCLASSES=/QIBM/ProdData/lotus/notes/data/
/domino/JAVA/NCSO.jar です。

コネクターの構成

コネクターの構成プロパティーには、標準構成プロパティーとアダプター固有の構成プロパティーという 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する前に、これらのプロパティー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・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、65 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・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. コネクタの標準構成プロパティ』を参照してください。
- コネクタ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. コネクタの標準構成プロパティ』を参照してください。

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 つまたは複数のコネクタ・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・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。

プロパティが動的であるか静的であるかを判別するには、Connector Configurator Express の「更新メソッド」欄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構成プロパティは、コネクタ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 (Application Config Properties)」タブを選択し、構成プロパティを追加または変更します。詳細については、65 ページの『付録 B. Connector Configurator Express』を参照してください。

表 5 に、コネクタのアプリケーション固有構成プロパティと、各プロパティ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

プロパティ	説明	型	デフォルト値
DominoServerName	Lotus Domino Server のマシン名	String	なし。
ApplicationUserName	Lotus Domino ユーザーの名前このプロパティは、 双方向言語に構成できます。	String	なし。このプロパティは、 双方向言語に構成で きます。
ApplicationPassword	Lotus Domino ユーザーのパスワード。このプロパティは、 双方向言語に構成できます。	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	<p>着信するビジネス・オブジェクトの Created と LastModified の有効な日付形式です。</p> <p>統合ブローカーからビジネス・オブジェクトを受信する場合には、Lotus Domino アダプターによりこのオブジェクトの Created と LastModified の日付が、Lotus Domino Server が認識できる Java 形式に変換されます。</p> <p>複数の形式を指定できます。</p> <p>複数の形式を指定するときには、セミコロン (;) で各形式を区切ります。</p> <p>アダプターにより、指定の形式の着信日付がリストの順序に従って比較されます。</p> <p>日付形式が指定されていない場合には、アダプターのホスト・システムの日付形式が使用されます。</p> <p>日付形式は、Java 2 Platform, Standard Edition (J2SE) v1.3.1 の java.text.SimpleDateFormat API で要求される日付形式と一致していなければなりません。</p>	String	<p>MM/dd/yy hh:mm aaa;</p> <p>MM/dd/yy</p>

DominoServerName、ApplicationUserName、および ApplicationPassword を構成する必要があります。統合ブローカーから受信されるビジネス・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に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	<p>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。</p> <p>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。</p>
FailOnStartup	<p>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。</p>
LogError	<p>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。</p>
Ignore	<p>イベント表の進行中イベント・レコードを無視します。</p>

コネクターの始動

コネクターは、**コネクター始動スクリプト**を使用して明示的に始動する必要があります。Windows システムでは、始動スクリプトはコネクターのランタイム・ディレクトリー `ProductDir¥connectors¥connName` に存在していなければなりません。ここで、`connName` はコネクターを示します。

Linux システムでは、始動スクリプトは `ProductDir/bin` ディレクトリーに存在していなければなりません。

i5/OS システムでは、始動スクリプトは、コネクターの実行に使用する `/QIBM/UserData/WBIServer44/<instance>/connectors/<ConnInstance/` に存在していなければなりません。

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. コネクターの始動スクリプト

オペレーティング・システム	始動スクリプト
Linux	connector_manager
i5/OS	start_connName.sh
Windows	start_connName.bat

始動スクリプトが実行されると、デフォルトで構成ファイルは `Productdir` にあることが要求されます (下記のコマンドを参照)。ここに構成ファイルを格納します。

注: アダプターが JMS トランSPORTを使用している場合は、ローカル構成ファイルが必要です。

• Windows システムでのコネクターの開始

- 「スタート」メニューから、「プログラム」>「IBM WebSphere Business Integration Server Express」>「アダプター」>「コネクター」を選択します。デフォルトでは、プログラム名は「IBM WebSphere Business Integration Server 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 コマンド行から、次を入力します: `start_connName connName brokerName {-cconfigFile}`
- 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。

• Linux システムでのコネクターの開始:

- コマンド行から、以下を入力します。
`connector_manager -start connName WebSphereICSName [-cconfigFile]`

ここで、`connName` はコネクターの名前であり、`brokerName` はご使用の統合ブローカーを表します。

- InterChange Server Express の場合は、*brokername* に InterChange Server Express インスタンスの名前を指定します。
- **i5/OS システムでのコネクターの開始**
 - WebSphere Business Integrations Server Express Console がインストールされている Windows システムから、「**IBM WebSphere Business Integration Server Express**」>「**Toolset Express**」>「**管理**」>「**コンソール**」を選択します。次に、OS/400 または i5/OS 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。コネクターのリストからコネクターを選択して、「開始」をクリックします。
 - コンソールを使用してアダプターを自動的に開始するには、`submit_adapter.sh` スクリプトを使用します。これが、サーバーの自動開始ジョブ・エントリ内のサブシステムを使用してアダプターが開始する唯一の方法です。
 - バッチ・モードでは、i5/OS コマンド行から CL コマンド QSH を実行し、QSHHELL 環境から `/QIBM/ProdData/WBIServer44/bin/submit_adapter.sh connName WebSphereICSName pathToConnNameStartScript jobDescriptionName` を実行する必要があります。ここで、*connName* はコネクター名、*WebSphereICSName* は Interchange Server Express サーバー名 (デフォルトは QWBIDFT44)、*pathToConnNameStartScript* はコネクター始動スクリプトの絶対パス、*jobDescriptionName* は QWBISVR44 ライブラリーで使用するジョブ記述の名前です。
 - 対話モードでは、CL コマンド QSH を実行し、QSHHELL 環境から `/QIBM/UserData/WBIServer44/WebSphereICSName/connectors/connName/start_connName.sh connNameWebSphereICSName [-cConfigFile]` を実行する必要があります。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。

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。

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。

- **Windows:**
 - コネクター用の別個の「コンソール」ウィンドウを作成する始動スクリプトを起動できます。このウィンドウで、「q」と入力して Enter キーを押すと、コネクターが停止します。
 - コネクターを Windows のサービスとして始動するように構成できます。この場合、Windows システムのシャットダウン時に、コネクターは停止します。
- **i5/OS:**
 - コンソールを使用して、または QSHHELL で「`submit_adapter.sh`」スクリプトを使用してコネクターを始動した場合は、次の 2 つの方法のうちの 1 つを使用してコネクターを停止できます。
 - WebSphere Business Integration Server Express Console がインストールされている Windows システムから、「**IBM WebSphere Business Integration Express**」>「**Toolset Express**」>「**管理**」>「**コンソール**」を選択します。次に、OS/400 または i5/OS 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。リストから Lotus Domino アダプターを選択して、「停止」ボタンを選択します。CL コマンド WRKACTJOB SBS (QWBISVR44) を使用して Server Express 製品に対するジョブを表示します。リストをスクロールして、コネクターのジョブ記述に一致するジョブ名を持つジョブを探します。例えば、Lotus Domino コネクターの場合のジョブ名は QWBIDOMC です。「コントロール パネル」このジョブに対してオプション 4 を選択し、F4 を押して ENDJOB コマンドのプロンプトを取得します。次に、オプション・パラメーターとして *IMMED を指定し、Enter を押します。

注: QWBISVR44 サブシステムが終了すると、コネクターは終了します。

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

/QIBM/ProdData/WBIServer44/bin ディレクトリーにあるスクリプト stop_adapter.sh を使用して、エージェントを停止することもできます。

- **Linux:**

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

```
connector_manager -stop connName
```

ここで、connName はコネクターの名前です。

複数のコネクター・インスタンスの作成

コネクターの複数のインスタンスを作成する作業は、いろいろな意味で、カスタム・コネクターの作成と同じです。以下に示すステップを実行することによって、コネクターの複数のインスタンスを作成して実行するように、ご使用のシステムを設定することができます。次のようにする必要があります。

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

新規ディレクトリーの作成

- **Windows** プラットフォームの場合:

```
ProductDir¥connectors¥connectorInstance
```

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

```
ProductDir¥repository¥connectorInstance
```

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

InterChange Server Express サーバー名は、例えば start_DOMINO.bat connName serverName のように、startup.bat のパラメーターとして指定できます。

- **i5/OS プラットフォームの場合:**

```
/QIBM/UserData/WBIServer44/WebSphereICSName  
/connectors/connectorInstance
```

ここで、connectorInstance はコネクタ・インスタンスを一意的に示し、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、次のディレクトリーを作成して、ファイルをそこに格納します。

```
/QIBM/UserData/WBIServer44/WebSphereICSName/repository/connectorInstance。こ  
こで WebSphereICSName はコネクタの実行に使用する Interchange Server Express  
インスタンスの名前です。
```

- **Linux プラットフォームの場合:**

ProductDir/connectors/connectorInstance。ここで connectorInstance は、コネクタ・インスタンスを一意的に示します。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、ディレクトリー ProductDir/repository/connectorInstance を作成し、ファイルをここに格納します。InterChange Server Express のサーバー名を connector_manager のパラメーターとして指定することができます。例: connector_manager -start connName WebSphereICSName [-cConfigFile]。

ビジネス・オブジェクト定義の作成

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

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

```
ProductDir¥repository¥initialConnectorInstance
```

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

コネクタ定義の作成

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

2. この始動スクリプトを、23 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. (i5/OS の場合のみ。) 以下の情報を使用して、コネクタのジョブ記述を作成します。
CRTDUPOBJ(QWBIDOMINO) FROMLIB(QWBISVR44)OBJTYPE(*JOB)TOLIB(QWBISVR44) NEWOBJ(newdominoname)。ここで、newdominoname は新規コネクタのジョブ記述で使用する 10 文字の名前です。
6. (i5/OS の場合のみ。) 新規コネクタを WebSphere Business Integration Server Express Console に追加します。WebSphere Business Integration Server Express Console 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。

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・オブジェクトには単純属性、つまり、1つの値 (ストリングなど) を表す属性のみが含まれます。子ビジネス・オブジェクトを指すことはありません。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・メッセージ、カレンダー・エントリー、予定表項目などの 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、イベント通知が統合ブローカーへ送信されます。

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	NotelID	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	LastModified	Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
2	Name	String	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	
3	Type	String	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	
4	Values	DominoItemValue	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N		
5	IsAuthors	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
6	ValueLength	Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
7	IsEncrypted	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
8	IsNames	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
9	IsProtected	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
10	IsReaders	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
11	IsSigned	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
12	IsSummary	Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
13	DummyKey	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	
14	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
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	Value	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
2	DummyKey	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
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 文書进行处理します。

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『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・オブジェクトを統合ブローカーに送信します。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、コネクターは統合ブローカーからビジネス・オブジェクトを受信します。統合ブローカー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 文書全体を統合ブローカー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。

注: 通常、アプリケーション固有のビジネス・オブジェクトは、可能であれば 1 MB 未満に制限し、5 MB を超えてはなりません。詳細については、40 ページの『ビジネス・オブジェクト・サイズの制限』を参照してください。

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、以下のように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、使用する統合ブローカー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」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。 (There is no value for configuration property xxx.) デフォルト値 yyy を使用します。 (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。 (There is 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	文書ビジネス・オブジェクトの統合ブローカーへの送信に失敗しました。(Failed to deliver the document business object to the integration broker.) イベント番号 xxx、状況 yyy。(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せん。
4036	送信を試みたビジネス・オブジェクトのサブスクリプションが見つかりませんでした。 (No subscription was found for the attempted business object delivery.) イベント番号 xxx、状況 yyy。(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、イベント番号 yyy の文書の検索に失敗しました。(Failed to retrieve the 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、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イベント表にありません。(There are no more events to poll from the event table.)	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、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。(The 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 イベントのデータベース xxx のオープンに失敗しました。(Failed to open the database: xxx for a Domino event.) 文書 noteID yyy。(Document noteID yyy.) イベント番号 zzz。(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)。(Encountered an IN_PROGRESS event, event number # xxx.) コネクタはエラーをログに記録するよう構成されています。(The connector has been configured to log an 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)。(Encountered an IN_PROGRESS event, event number # xxx.) コネクタは致命的エラーをログに記録するよう構成されています。(The connector has been configured to log a 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。(Encountered an IN_PROGRESS event.) コネクタはイベントを無視するよう構成されています。(The connector has been configured to ignore the 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。(Encountered an IN_PROGRESS event.) コネクタはイベントを再処理するよう構成されています。(The connector has been configured to reprocess the event.)	コネクタ構成に基づき、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・テーブル・データベースまたはビューへの接続の確立に失敗しました。(Failed to establish a connection to the event table database or 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	この Lotus Domino セッションは無効です。(This 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。(Error accessing Domino server.) サーバー: xxx、ユーザー名: yyy、 エラー・テキスト: zzz (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。詳細については、エラー・テキストを参照してください。
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。(Error accessing database: xxx.) エラー・テキスト: yyy (Error text: yyy)	指定されたデータベースへのアクセス中にエラーが発生しました。詳細については、エラー・テキストを参照してください。
5003	データベース xxx が見つかりません。 (Database xxx not found.)	指定されたデータベースがサーバー上で見つかりませんでした。
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。(Error opening database: xxx) データベースが存在しません。(The database does not exist.)	指定されたデータベースはサーバー上に存在しません。
5006	動詞: xxx がサポートされていません。 (Unsupported verb: xxx)	指定された動詞は、コネクタでサポートされていません。
5007	noteID xxx を持つ文書のオープン中にエラーが発生しました。(Error opening document with noteID xxx.) エラー・テキスト: yyy (Error text: yyy.)	指定された文書のオープン中にエラーが発生しました。詳細については、エラー・テキストを参照してください。
5008	noteID xxx を持つ文書のオープン中にエラーが発生しました。(Error opening document with noteID xxx.) データベース内に文書が見つかりませんでした。(The document was not found in the 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。 (Error creating document.) エラー・テキスト: xxx (Error text: xxx.)	データベースで新規文書の作成中にエラーが発生しました。詳細については、エラー・テキストを参照してください。
5010	文書から noteID の検索中にエラーが発生しました。(Error retrieving noteID from the document.)	データベースに文書を正常に作成した後、その文書から noteID を検索している時にエラーが発生しました。noteID はビジネス・オブジェクトに追加されません。
5011	noteID xxx を持つ文書の削除中にエラーが発生しました。(Error deleting the document with noteID xxx.) エラー・テキスト: yyy (Error text: yyy.)	指定された文書をデータベースから削除している時にエラーが発生しました。詳細については、エラー・テキストを参照してください。
5012	noteID xxx を持つ文書の検証中にエラーが発生しました。(Error validating document with noteID xxx.) エラー・テキスト: yyy (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。詳細については、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。 (Error saving document.) エラー・テキスト: xxx (Error text: xxx.)	指定された文書の保管中にエラーが発生しました。詳細については、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
5014	<p>文書は保管されませんでした。(The Document was not saved.)</p> <p>考えられる原因は以下のとおりです。(Possible causes are:)</p> <ol style="list-style-type: none"> 1. 文書が要求によって変更されなかった。(The document was unchanged by the request.) 2. ForceSave および MakeResponse プロパティが誤っている。(The ForceSave and MakeResponse properties are false.) 	<p>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こと、または ForceSave および MakeResponse コネクター・プロパティ値が原因です。</p>
5015	<p>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 the business object yyy.)</p>	<p>指定された属性は、指定されたビジネス・オブジェクトで見つかりませんでした。</p>
5016	<p>属性 xxx が、ビジネス・オブジェクト yyy の String 型ではありません。(Attribute xxx is not String type in the business object yyy.)</p>	<p>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。</p>
5017	<p>属性 xxx が、ビジネス・オブジェクト yyy の Boolean 型ではありません。(Attribute xxx is not Boolean type in the business object yyy.)</p>	<p>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。</p>
5018	<p>属性 xxx が、ビジネス・オブジェクト yyy のビジネス・オブジェクト・タイプではありません。(Attribute xxx is not a type in the business object yyy.)</p>	<p>指定された属性が、指定のビジネス・オブジェクトのビジネス・オブジェクト型属性ではありません。</p>
5019	<p>項目 xxx を文書に追加中にエラーが発生しました。(Error adding item xxx to the document.) エラー・テキスト: yyy (Error text: yyy.)</p>	<p>項目を文書に追加中にエラーが発生しました。詳細については、エラー・テキストを参照してください。</p>
5020	<p>文書の親データベースを検索中にエラーが発生しました。(Error retrieving the document's parent database.) エラー・テキスト: yyy (Error text: yyy.)</p>	<p>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。詳細については、エラー・テキストを参照してください。</p>
5021	<p>現在のセッションをデータベースから検索中にエラーが発生しました。(Error retrieving the current session from the database.) エラー・テキスト: xxx (Error text: xxx.)</p>	<p>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。詳細については、エラー・テキストを参照してください。</p>
5022	<p>ストリング xxx から DateTime オブジェクトを作成中にエラーが発生しました。(Error creating DateTime object from the string xxx.) エラー・テキスト: yyy (Error text: yyy.)</p>	<p>指定されたストリングから DateTime オブジェクトを作成中にエラーが発生しました。詳細については、エラー・テキストを参照してください。日付ストリングは、java.text.SimpleDateFormat API で指定されたパターンと同じにする必要があります。</p>
5023	<p>型 xxx に対して ItemHandler が定義されていません。(No ItemHandler defined for type: xxx.) 有効な型は、Double、String、Date です。(Valid types include Double, String, and Date)</p>	<p>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。</p>

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
5024	項目 yyy でフラグ xxx の設定中にエラーが発生しました。(Error setting flag xxx in item yyy.) 例外テキスト: zzz。(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。詳細については、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 オブジェクトに変換中にエラーが発生しました。
5026	Lotus Domino 文書をビジネス・オブジェクトに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 the Domino document to a business object with noteID xxx Error text: yyy.)	文書をビジネス・オブジェクトに変換中にエラーが発生しました。詳細については、エラー・テキストを参照してください。
5029	項目 xxx を noteID yyy の文書から除去中にエラーが発生しました。(Error removing item xxx from the document noteID yyy.) エラー・テキスト: zzz (Error text: zzz.)	指定された項目を文書から除去中にエラーが発生しました。詳細については、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中および検証中にエラーが発生しました。(Error initializing and validating the business object attributes.) ビジネス・オブジェクト名: xxx (Business object name: xxx.) エラー・テキスト: yyy (Error text: yyy)	指定されたビジネス・オブジェクトの CWConnectorUtil.initAndValidateAttributes() への呼び出しで、エラーが発生しました。詳細については、エラー・テキストを参照してください。
5031	サブ動詞 xxx は、Lotus Domino 項目 yyy でサポートされていないサブ動詞です。(Unsupported subverb: xxx for Domino item yyy.)	指定されたサブ動詞はコネクターでサポートされていません。
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。(Error parsing the date using default format for locale.) 日付: xxx。(Date: xxx.) ロケール: yyy。(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。
5033	構成された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。(Error parsing the date string using the configured format.) 日付: xxx。(Date: xxx.) ロケール: yyy。(Locale: yyy.) 形式: zzz。(Format: zzz.)	指定の形式およびロケールを使用して指定された日付ストリングを解析中にエラーが発生しました。
5034	構成されている日付形式ストリングを使用して Date オブジェクトを生成できませんでした。(Unable to generate a object using configured date format strings.) 日付: xxx。(Date: xxx.) ロケール: yyy。(Locale: yyy.)	エラーが記録される前に、すべての日付形式が試行されます。Date を生成する試みがすべて失敗していることを確認してください。失敗している場合、DateFormats コネクター・アプリケーション固有のプロパティに必要な日付形式を指定します。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。

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
5035	ビジネス・オブジェクトに noteID 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 noteID が常に指定されていることを確認します。
5038	名前属性に無効な値が指定されました。(An invalid value was specified for the name attribute.) 名前 xxx。(Name xxx.) ビジネス・オブジェクト yyy。(Business object yyy.)	この属性に CxIgnore 値または CxBlank 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。
レベル 1	<ul style="list-style-type: none"> 状況情報を示します。 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。 ポーリングが発生するたびに記録します。
レベル 2	<ul style="list-style-type: none"> コネクター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。 ビジネス・オブジェクトが統合ブローカーに通知されるたびに、ログに記録します。 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。
レベル 3	<ul style="list-style-type: none"> 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、コネクターがビジネス・オブジェクトで外部キーを検出した場合、またはコネクターがビジネス・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。 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。
レベル 4	<ul style="list-style-type: none"> 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。 コネクター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、コネクターのプロセス・フローをトレースするときに役立ちます。 スレッド固有の処理をすべて記録します。例えば、コネクター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。

表9. トレース・メッセージの内容 (続き)

レベル	説明
レベル 5	<ul style="list-style-type: none"> コネクタ初期化を示します。このタイプのメッセージには、統合ブローカーから検索された各 Connector Configurator Express プロパティの値などを含めることができます。 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。 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。 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。

ビジネス・オブジェクト・サイズの制限

InterChange Server Express が稼働する Java 仮想マシンには、基本的な制限があります。

これらの制限は、以下の理由により推奨されます。

- ビジネス・オブジェクトが InterChange Server Express を介して伝搬するときに、参照がビジネス・オブジェクトに追加され、アプリケーション固有のビジネス・オブジェクトから GBO へ進むときに初期ビジネス・オブジェクトのサイズが拡張されます。
- トランザクション動作が必要な場合はさらに大きなオーバーヘッドが必要であり、ビジネス・オブジェクト・サイズも大きくなります。このオーバーヘッドは、InterChange Server Express 内のビジネス・オブジェクトに対して実行される多様なビジネス・ロジックの処理の永続性および柔軟性のために必要です。

注: JVM に対してオブジェクトが大きいと (20 MB より大)、過剰なヒープ・フラグメントを招いたり、十分なメモリーを割り振れなくなり、`java.lang.OutOfMemory` エラーが発生して InterChange Server Express がシャットダウンします。

付録 A. コネクターの標準構成プロパティ

この付録では、WebSphere Business Integration Server Express アダプターのコネクタ・コンポーネントの標準構成プロパティについて説明します。説明は、InterChange Server Express が対象となります。

このコネクタに固有のプロパティについては、本書の該当するセクションを参照してください。

新規プロパティ

以下の標準プロパティは、本リリースで追加されました。

- AdapterHelpName
- BiDi.Application
- BiDi.Broker
- BiDi.Metadata
- BiDi.Transformation
- ControllerEventSequencing
- jms.ListenerConcurrency
- jms.TransportOptimized
- TivoliTransactionMonitorPerformance

標準コネクタ・プロパティの概要

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。フレームワークが使用します。
- アプリケーション固有またはコネクタ固有の構成プロパティ。エージェントが使用します。

これらのプロパティは、アダプターのフレームワークおよびエージェントの実行時の振る舞いを決定します。

このセクションでは、Connector Configurator Express の始動方法について説明し、すべてのプロパティに共通する特性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ズ・ガイドを参照してください。

Connector Configurator Express の始動

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 の使用法の詳細については、本書の Connector Configurator Express に関する付録を参照してください。

Connector Configurator Express と System Manager は、Windows システム上でのみ動作します。コネクタを Linux システム上で稼動している場合でも、これらのツールがインストールされた Windows マシンが必要です。

Linux 上で動作するコネクタのコネクタ・プロパティを設定する場合は、Windows マシン上で System Manager を起動し、Linux の統合ブローカーに接続してから、コネクタ用の Connector Configurator Express を開く必要があります。

構成プロパティ値の概要

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。

1. デフォルト
2. InterChange Server Express 統合ブローカー用のリポジトリ
3. ローカル構成ファイル
4. コマンド行

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。STRING プロパティ・タイプの長さに制限はありません。INTEGER タイプの長さは、アダプターを実行しているサーバーによって決まります。

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。

プロパティの更新特性 (すなわちコネクタ・プロパティへの変更を有効にする方法とタイミング) は、プロパティの性質によって異なります。

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、新しい値が即時に有効になります。ただし、コネクタがスタンドアロン・モードの場合 (System Manager に依存しない) です。

• エージェント再始動 (InterChange Server Express のみ)

コネクタ・エージェントを停止して再始動しなければ、新規の値が有効になりません。

• コンポーネント再始動

System Manager でコネクタを停止してから再始動しなければ、新しい値が有効になりません。エージェントまたはサーバー・プロセスを停止して再始動する必要はありません。

• システム再始動

コネクタ・エージェントおよびサーバーを停止して再始動しなければ、新規の値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、43 ページの表 10 の「更新メソッド」列を参照してください。

標準プロパティが存在できる場所が 3 箇所あります。一部のプロパティは複数の場所にあってもかまいません。

- **ReposController**

このプロパティはコネクタ・コントローラ内にあり、その場所でのみ有効です。エージェント・サイドで値を変更した場合、コントローラには影響しません。

- **ReposAgent**

このプロパティはエージェント内にあり、その場所でのみ有効です。プロパティによっては、ローカル構成によってこの値をオーバーライドされることがあります。

- **LocalConfig**

このプロパティは、コネクタの構成ファイル内にあり、構成ファイルを通じてのみ機能することができます。コントローラはこのプロパティの値を変更することができず、システムが再配置されてコントローラが明示的に更新されなければ、構成ファイルに加えられた変更を認識しません。

標準プロパティの早見表

表 10 は、標準コネクタ構成プロパティの早見表です。すべてのコネクタでこれらのプロパティすべてを必要とするわけではなく、プロパティ設定は異なる場合があります。

各プロパティの説明については、表の次のセクションを参照してください。

注: 表 10 の注の欄で、「RepositoryDirectory が <REMOTE> に設定され」という句は、ブローカーが InterChange Server Express であることを示します。

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
AdapterHelpName	有効な <Regional Setting> ディレクトリーを含む <ProductDir>%bin%Data%App%Help 内の有効なサブディレクトリーのいずれか	テンプレート名 (有効な場合) またはブランク・フィールド	コンポーネント再始動	サポートされる地域設定。 chs_chn、cht_twn、 deu_deu、esn_esp、 fra_fra、ita_ita、 jpn_jpn、kor_kor、 ptb_bra、および enu_usa (デフォルト) を含む。
AdminInQueue	有効な JMS キュー名	<CONNECTORNAME> /ADMININQUEUE	コンポーネント再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
AdminOutQueue	有効な JMS キュー名	<CONNECTORNAME> /ADMINOUTQUEUE	コンポーネント再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
AgentConnections	1 から 4	1	コンポーネント再始動	このプロパティは、 DeliveryTransport の 値が MQ または IDL で、 RepositoryDirectory の値が <REMOTE> に設定され、 BrokerType の値が ICS の 場合のみ有効です。
AgentTraceLevel	0 から 5	0	ICS では動的、 その他の場合は コンポーネント 再始動	

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
ApplicationName	アプリケーション名	コネクターのアプリケーション名に指定された値	コンポーネント再始動	
BiDi.Application	以下の双方向属性の任意の有効な組み合わせ 最初の文字: I、V 2 番目の文字: L、R 3 番目の文字: Y、N 4 番目の文字: S、N 5 番目の文字: H、C、N	ILYNN (5 文字)	コンポーネント再始動	このプロパティは、BiDi.Transformation の値が true の場合のみ有効です。
BiDi.Broker	以下の双方向属性の任意の有効な組み合わせ 最初の文字: I、V 2 番目の文字: L、R 3 番目の文字: Y、N 4 番目の文字: S、N 5 番目の文字: H、C、N	ILYNN (5 文字)	コンポーネント再始動	このプロパティは、BiDi.Transformation の値が true の場合のみ有効です。BrokerType の値が ICS の場合、プロパティは読み取り専用です。
BiDi.Metadata	以下の双方向属性の任意の有効な組み合わせ 最初の文字: I、V 2 番目の文字: L、R 3 番目の文字: Y、N 4 番目の文字: S、N 5 番目の文字: H、C、N	ILYNN (5 文字)	コンポーネント再始動	このプロパティは、BiDi.Transformation の値が true の場合のみ有効です。
BiDi.Transformation	true または false	false	コンポーネント再始動	このプロパティは、BrokerType の値が WAS でない場合のみ有効です。
BrokerType	ICS	ICS	コンポーネント再始動	
CharacterEncoding	サポートされる任意のコード。 次のリストはその一部です。 ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437	ascii7	コンポーネント再始動	このプロパティは、C++ コネクターでのみ有効です。
CommonEventInfrastructure	true または false	false	コンポーネント再始動	
CommonEventInfrastructureURL	URL スtring。 例えば、 corbaloc:iiop: host:2809。	デフォルト値はありません。	コンポーネント再始動	このプロパティは、CommonEventInfrastructure の値が true の場合のみ有効です。

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
ConcurrentEventTriggeredFlows	1 から 32,767	1	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
ContainerManagedEvents	ブランクまたは JMS	ブランク	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
ControllerEventSequencing	true または false	true	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
ControllerStoreAndForwardMode	true または false	true	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
ControllerTraceLevel	0 から 5	0	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
DeliveryQueue	任意の有効な JMS キュー名	<CONNECTORNAME>/DELIVERYQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
DeliveryTransport	IDL または JMS	RepositoryDirectory の値が <REMOTE> の場合は IDL。それ以外の場合は JMS。	コンポーネント再始動	RepositoryDirectory の値が <REMOTE> ではない場合、このプロパティの有効な値は JMS のみです。
DuplicateEventElimination	true または false	false	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
EnableOidForFlowMonitoring	true または false	false	コンポーネント再始動	このプロパティは、BrokerType の値が ICS の場合のみ有効です。
FaultQueue	任意の有効なキュー名	<CONNECTORNAME>/FAULTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory、CxCommon.Messaging.jms.SonicMQFactory、または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

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
jms.ListenerConcurrency	1 から 32767	1	コンポーネント再始動	このプロパティは、 <code>jms.TransportOptimized</code> の値が <code>true</code> の場合のみ有効です。
jms.MessageBrokerName	<code>jms.FactoryClassName</code> の値が IBM の場合は、 <code>crossworlds.queue.manager</code> を使用します。	<code>crossworlds.queue.manager</code>	コンポーネント再始動	このプロパティは、 <code>DeliveryTransport</code> の値が JMS の場合のみ有効です。
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	このプロパティは、 <code>DeliveryTransport</code> の値が JMS の場合のみ有効です。
jms.Password	任意の有効なパスワード		コンポーネント再始動	このプロパティは、 <code>DeliveryTransport</code> の値が JMS の場合のみ有効です。
jms.TransportOptimized	<code>true</code> または <code>false</code>	<code>false</code>	コンポーネント再始動	このプロパティは、 <code>DeliveryTransport</code> の値が JMS で、 <code>BrokerType</code> の値が ICS の場合のみ有効です。
jms.UserName	任意の有効な名前		コンポーネント再始動	このプロパティは、 <code>DeliveryTransport</code> の値が JMS の場合のみ有効です。
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	このプロパティは、 <code>RepositoryDirectory</code> の値が <code><REMOTE></code> に設定され、 <code>BrokerType</code> の値が ICS の場合のみ有効です。
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	このプロパティは、 <code>RepositoryDirectory</code> の値が <code><REMOTE></code> に設定され、 <code>BrokerType</code> の値が ICS の場合のみ有効です。
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	このプロパティは、 <code>RepositoryDirectory</code> の値が <code><REMOTE></code> に設定され、 <code>BrokerType</code> の値が ICS の場合のみ有効です。
ListenerConcurrency	1 から 100	1	コンポーネント再始動	このプロパティは、 <code>DeliveryTransport</code> の値が MQ の場合のみ有効です。
Locale	これは、サポートされるロケールの一部です。 en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR	en_US	コンポーネント再始動	
LogAtInterchangeEnd	<code>true</code> または <code>false</code>	<code>false</code>	コンポーネント再始動	このプロパティは、 <code>RepositoryDirectory</code> の値が <code><REMOTE></code> に設定され、 <code>BrokerType</code> の値が ICS の場合のみ有効です。

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
MaxEventCapacity	1 から 2147483647	2147483647	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
MessageFileName	有効なファイル名	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	このプロパティは、DuplicateEventElimination の値が true で、ContainerManagedEvents に値がない場合にのみ有効です。
OADAutoRestartAgent	true または false	false	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
OADMaxNumRetry	正整数	1000	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
OADRetryTimeInterval	正整数 (単位: 分)	10	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
PollEndTime	HH = 0 から 23 MM = 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)	10000	ブローカーが ICS の場合は動的。 そうでない場合は、コンポーネント再始動。	
PollQuantity	1 から 500	1	エージェント再始動	このプロパティは、ContainerManagedEvents の値が JMS の場合のみ有効です。
PollStartTime	HH = 0 から 23 MM = 0 から 59	HH:MM	コンポーネント再始動	
RepositoryDirectory	ブローカーが ICS の場合は <REMOTE>。 それ以外の場合は任意の有効なローカル・ディレクトリー。	ICS の場合、値は <REMOTE> に設定されます。	エージェント再始動	
RequestQueue	有効な JMS キュー名	<CONNECTORNAME> /REQUESTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

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
ResponseQueue	有効な JMS キュー名	<CONNECTORNAME> /RESPONSEQUEUE	コンポーネント 再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ 有効です。
RestartRetryCount	0 から 99	3	ICS の場合は 動的、 その他の場合は コンポーネント 再始動	
RestartRetryInterval	1 から 2147483647 までの値 (分単位)	1	ICS の場合は 動的、 その他の場合は コンポーネント 再始動	
RHF2MessageDomain	mrm または xml	mrm	コンポーネント 再始動	このプロパティは、 DeliveryTransport の値が JMS で、 WireFormat の値が CwXML の場合のみ 有効です。
SourceQueue	任意の有効な WebSphere MQ キュー名	<CONNECTORNAME> /SOURCEQUEUE	エージェント 再始動	このプロパティは、 ContainerManagedEvents の値が JMS の場合のみ 有効です。
SynchronousRequest Queue	任意の有効なキュー名	<CONNECTORNAME> /SYNCHRONOUSREQUEST QUEUE	コンポーネント 再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ 有効です。
SynchronousRequest Timeout	0 から任意の数 (ミリ秒)	0	コンポーネント 再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ 有効です。
SynchronousResponse Queue	任意の有効なキュー名	<CONNECTORNAME> /SYNCHRONOUSRESPONSE QUEUE	コンポーネント 再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ 有効です。
TivoliMonitorTransaction Performance	true または false	false	コンポーネント 再始動	
WireFormat	CwXML または CwBO	CwXML	エージェント 再始動	RepositoryDirectory の 値が <REMOTE> に 設定されていない場合、 このプロパティの値は、 CwXML で なければなりません。 RepositoryDirectory の 値が <REMOTE> に 設定されている場合、 値は CwBO で なければなりません。
WsifSynchronousRequest Timeout	0 から任意の数 (ミリ秒)	0	コンポーネント 再始動	BrokerType の値が ICS の場合、この プロパティは無効です。

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
XMLNamespaceFormat	short または long	short	エージェント再始動	BrokerType の値が ICS の場合、このプロパティは無効です。

標準プロパティ

このセクションでは、標準コネクタ構成プロパティについて説明します。

AdapterHelpName

AdapterHelpName プロパティは、コネクタ固有の全般ヘルプ・ファイルがあるディレクトリの名前です。ディレクトリは、`<ProductDir>\¥bin¥Data¥App¥Help` 内に配置される必要があり、少なくとも言語ディレクトリ `enu_usa` が含まれていなければなりません。ロケールに応じて、その他のディレクトリが含まれることがあります。

デフォルト値は、テンプレート名が有効であればテンプレート名、有効でなければ空白です。

AdminInQueue

AdminInQueue プロパティは、統合ブローカーがコネクタへ管理メッセージを送信するときに使用するキューを指定します。

デフォルト値は `<CONNECTORNAME>/ADMININQUEUE` です。

AdminOutQueue

AdminOutQueue プロパティは、コネクタが統合ブローカーへ管理メッセージを送信するときに使用するキューを指定します。

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

AgentConnections

AgentConnections プロパティは、ORB (オブジェクト・リクエスト・ブローカー) が初期化するときにかかれる ORB 接続の数を制御します。

このプロパティのデフォルト値は 1 です。

AgentTraceLevel

AgentTraceLevel プロパティは、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を設定します。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。

デフォルト値は 0 です。

ApplicationName

ApplicationName プロパティは、コネクタ・アプリケーションの名前を一意的に識別します。この名前は、システム管理者が統合環境をモニターするために使用します。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。

デフォルトではコネクタの名前です。

BiDi.Application

BiDi.Application プロパティは、このアダプターがサポートする任意のビジネス・オブジェクトの形式で、外部アプリケーションからアダプターに入ってくるデータの双方向フォーマットを指定します。このプロパティは、アプリケーション・データの双方向属性を定義します。これらの属性は以下のとおりです。

- テキストのタイプ: 暗黙または可視 (I または V)
- テキストの方向: 左から右または右から左 (L または R)
- 対称スワッピング: オンまたはオフ (Y または N)
- 成形 (アラビア語): オンまたはオフ (S または N)
- 数字成形 (アラビア語): ヒンディ語、コンテキスト、または名詞 (H、C、または N)

このプロパティは、**BiDi.Transformation** プロパティの値が `true` に設定されている場合のみ有効です。

デフォルト値は `ILYNN` (暗黙、左から右、オン、オフ、名詞) です。

BiDi.Broker

BiDi.Broker プロパティは、サポートされる任意のビジネス・オブジェクトの形式で、アダプターから統合ブローカーに送信されるデータの双方向フォーマットを指定します。データの双方向属性を定義します。属性は、前述の **BiDi.Application** の下にリストされています。

このプロパティは、**BiDi.Transformation** プロパティの値が `true` に設定されている場合のみ有効です。**BrokerType** プロパティが `ICS` の場合、プロパティ値は読み取り専用です。

デフォルト値は `ILYNN` (暗黙、左から右、オン、オフ、名詞) です。

BiDi.Metadata

BiDi.Metadata プロパティは、メタデータの双方向フォーマットまたは属性を定義します。メタデータは、外部アプリケーションへのリンクを確立および保守するために、コネクターが使用します。属性の設定は、双方向機能を使用する各アダプターに固有です。アダプターが双方向処理をサポートする場合、詳細についてはアダプター固有のプロパティに関するセクションを参照してください。

このプロパティは、**BiDi.Transformation** プロパティの値が `true` に設定されている場合のみ有効です。

デフォルト値は ILYNN (暗黙、左から右、オン、オフ、名詞) です。

BiDi.Transformation

BiDi.Transformation プロパティは、システムが実行時に双方向変換を実行するかどうかを定義します。

プロパティ値が true に設定されている場合、BiDi.Application、BiDi.Broker、および BiDi.Metadata プロパティが使用可能です。プロパティ値が false に設定されている場合は、それらは非表示になります。

デフォルト値は false です。

BrokerType

BrokerType プロパティは、使用している統合ブローカーのタイプを識別します。値は ICS です。

CharacterEncoding

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

注: Java ベースのコネクターでは、このプロパティは使用しません。C++ ベースのコネクターでは、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、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。サポートされる他の値をリストに追加するには、製品ディレクトリー (<ProductDir>) 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、本書の付録『Connector Configurator Express』を参照してください。

ConcurrentEventTriggeredFlows

ConcurrentEventTriggeredFlows プロパティは、コネクターがイベントのデリバリー時に並行処理できるビジネス・オブジェクトの数を決定します。この属性の値を、並行してマップおよびデリバリーされるビジネス・オブジェクトの数に設定します。例えば、このプロパティの値を 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。

このプロパティを 1 よりも大きい値に設定すると、ソース・アプリケーションのコネクターが、複数のイベント・ビジネス・オブジェクトを同時にマップして、複数のコラボレーション・インスタンスにそれらのビジネス・オブジェクトを同時にデリバリーすることができます。これにより、統合ブローカーへのビジネス・オブジェクトのデリバリーにかかる時間、特にビジネス・オブジェクトが複雑なマップを使用している場合のデリバリー時間が短縮されます。ビジネス・オブジェクトのコラボレーションに到達する速度を増大させると、システム全体のパフォーマンスを向上させることができます。

ソース・アプリケーションから宛先アプリケーションまでのフロー全体に並行処理を実装するには、以下のプロパティを構成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、複数のスレッドを使用できるようにコラボレーションを構成する必要があります。
- 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を、複数の要求を並行して処理できるように構成する必要があります。

`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。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は 1 です。

ContainerManagedEvents

`ContainerManagedEvents` 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、1 つの 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設定する必要があります。

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

また、`MimeType` および `DHClass` (データ・ハンドラー・クラス) 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。`DataHandlerConfigMOName` (オプションのメタオブジェクト名) を追加することもできます。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。

これらのプロパティはアダプター固有ですが、以下に値の例をいくつか示します。

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

「データ・ハンドラー」タブのこれらの値のフィールドは、`ContainerManagedEvents` プロパティを JMS という値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。

`ContainerManagedEvents` プロパティは、`DeliveryTransport` プロパティの値が JMS に設定されている場合のみ有効です。

デフォルト値はありません。

ControllerEventSequencing

ControllerEventSequencing プロパティは、コネクタ・コントローラーでイベント順序付けを使用可能にします。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType は ICS) のみ有効です。

デフォルト値は true です。

ControllerStoreAndForwardMode

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

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

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

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType プロパティの値が ICS) のみ有効です。

デフォルト値は true です。

ControllerTraceLevel

ControllerTraceLevel プロパティは、コネクタ・コントローラーのトレース・メッセージのレベルを設定します。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 0 です。

DeliveryQueue

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

このプロパティは、DeliveryTransport プロパティの値が JMS に設定されている場合のみ有効です。

デフォルト値は <CONNECTORNAME>/DELIVERYQUEUE です。

DeliveryTransport

DeliveryTransport プロパティは、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。Java Messaging Service の場合、値は JMS です。

- RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合、DeliveryTransport プロパティの値には IDL または JMS を指定することができ、デフォルトは IDL です。
- RepositoryDirectory プロパティの値がローカル・ディレクトリーの場合、値に使用できるのは JMS のみです。

RepositoryDirectory プロパティの値が IDL である場合、コネクタは、CORBA IIOP を使用してサービス呼び出し要求と管理メッセージを送信します。

デフォルト値は JMS です。

JMS

JMS トランスポート機構は、Java Messaging Service (JMS) を使用した、コネクタークライアント・コネクタークラウドとの間の通信を可能にします。

JMS をデリバリー・トランスポートとして選択した場合は、

.jms.MessageBrokerName、.jms.FactoryClassName、.jms.Password、.jms.UserName などの追加の JMS プロパティが Connector Configurator Express 内にリストされます。jms.MessageBrokerName と jms.FactoryClassName は、このトランスポートの必須プロパティです。

InterChange Server Express (ICS) が統合ブローカーである場合、以下の環境では、コネクタに JMS トランスポート機構を使用すると、メモリー制限が発生することもあります。

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) コネクタークントローラーと (クライアント側の) コネクタークラウドの両方を始動するのは困難な場合があります。ご使用のシステムのプロセス・ヒープ・サイズが 768MB 未満である場合には、次の変数およびプロパティを設定してください。

- CWSharedEnv.sh スクリプト内で LDR_CNTRL 環境変数を設定する。

このスクリプトは、製品ディレクトリー (<ProductDir>) の下の ¥bin ディレクトリーにあります。テキスト・エディターを使用して、CWSharedEnv.sh スクリプトの最初の行として次の行を追加します。

```
export LDR_CNTRL=MAXDATA=0x30000000
```

この行は、ヒープ・メモリーの使用量を最大 768 MB (3 セグメント * 256 MB) に制限します。プロセス・メモリーがこの制限値を超えると、ページ・スワッピングが発生し、システムのパフォーマンスに悪影響を与える場合があります。

- IPCCBaseAddress プロパティの値を 11 または 12 に設定する。このプロパティの詳細については、「WebSphere Business Integration Server Express インストール・ガイド (Windows 版)」、「WebSphere Business Integration Server Express

インストール・ガイド (Linux 版)」、または「*WebSphere Business Integration Server Express* インストール・ガイド (i5/OS 版)」を参照してください。

DuplicateEventElimination

このプロパティの値が `true` の場合は、JMS 対応コネクタで重複イベントをデリバリー・キューに配信しないようにできます。この機能を使用するには、コネクタ開発時に、コネクタに対し、アプリケーション固有のコード内でビジネス・オブジェクトの `ObjectEventId` 属性として一意のイベント ID が設定されている必要があります。

注: このプロパティの値が `true` の場合、保証付きイベント・デリバリーを提供するには、`MonitorQueue` プロパティを使用可能にする必要があります。

デフォルト値は `false` です。

EnableOidForFlowMonitoring

このプロパティの値が `true` の場合、アダプター・ランタイムは、着信 `ObjectEventID` にフロー・モニターの外部キーのマークを付けます。

このプロパティは、`BrokerType` プロパティが `ICS` に設定されている場合のみ有効です。

デフォルト値は `false` です。

FaultQueue

コネクタでメッセージを処理中にエラーが発生すると、コネクタは、そのメッセージ (および状況標識と問題説明) を `FaultQueue` プロパティで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。

jms.FactoryClassName

`jms.FactoryClassName` プロパティは、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。`DeliveryTransport` プロパティの値が `JMS` に設定されている場合、このプロパティを設定する必要があります。

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

jms.ListenerConcurrency

`jms.ListenerConcurrency` プロパティは、JMS コントローラーの並行リスナーの数を指定します。コントローラー内部で、並行してメッセージを取り出して処理するスレッドの数を指定します。

このプロパティは、`jms.OptimizedTransport` プロパティの値が `true` の場合のみ有効です。

デフォルト値は `1` です。

jms.MessageBrokerName

`jms.MessageBrokerName` は、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 (`DeliveryTransport` プロパティで) 指定する場合、このコネクタ・プロパティを設定する必要があります。

リモート・メッセージ・ブローカーに接続した場合、このプロパティでは以下の値を指定する必要があります。

`QueueMgrName:Channel:HostName:PortNumber`

ここで、以下のように説明されます。

`QueueMgrName` は、キュー・マネージャー名です。

`Channel` は、クライアントが使用するチャンネルです。

`HostName` は、キュー・マネージャーの配置先のマシン名です。

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

例えば、次のように指定します。

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

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

jms.NumConcurrentRequests

`jms.NumConcurrentRequests` プロパティは、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はブロックされ、処理を続行するには他のいずれかの要求が完了するのを待機する必要があります。

デフォルト値は 10 です。

jms.Password

`jms.Password` プロパティは、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。

デフォルト値はありません。

jms.TransportOptimized

`jms.TransportOptimized` プロパティは、WIP (処理中の作業) が最適化されるかどうかを決定します。WIP を最適化するには、WebSphere MQ プロバイダーが必要です。最適化された WIP が作動するためには、メッセージング・プロバイダーが以下の操作を実行できなければなりません。

1. メッセージをキューから削除せずに読み取る。
2. メッセージ全体を受信側のメモリー空間に転送することなく、固有の ID を使用してメッセージを削除する。
3. 固有の ID を使用してメッセージを読み取る (リカバリーのために必要)。
4. 読み取られなかったイベントが現れるポイントを追跡する。

JMS API は、上記の条件 2 および 4 を満たさないため、最適化された WIP には使用できませんが、MQ Java API は 4 つの条件をすべて満たすため、最適化された WIP には必要です。

このプロパティーは、DeliveryTransport の値が JMS で、BrokerType の値が ICS の場合のみ有効です。

デフォルト値は false です。

jms.UserName

jms.UserName プロパティーは、JMS プロバイダーのユーザー名を指定します。このプロパティーの値はオプションです。

デフォルト値はありません。

JvmMaxHeapSize

JvmMaxHeapSize プロパティーは、エージェントの最大ヒープ・サイズ (メガバイト単位) を指定します。

このプロパティーは、RepositoryDirectory プロパティーの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 128M です。

JvmMaxNativeStackSize

JvmMaxNativeStackSize プロパティーは、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) を指定します。

このプロパティーは、RepositoryDirectory プロパティーの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 128K です。

JvmMinHeapSize

JvmMinHeapSize プロパティーは、エージェントの最小ヒープ・サイズ (メガバイト単位) を指定します。

このプロパティーは、RepositoryDirectory プロパティーの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 1M です。

ListenerConcurrency

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

このプロパティは、MQ トランスポートを使用するコネクタのみで有効です。
DeliveryTransport プロパティの値には MQ を設定してください。

デフォルト値は 1 です。

Locale

Locale プロパティは、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、通貨規格で使用される記号などの国/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。

`ll` は、2 文字の言語コード (小文字を使用) です。

`TT` は、2 文字の国または地域コード (大文字を使用) です。

`codeset` は、関連文字コード・セットの名前です (オプションの場合があります)。

デフォルトでは、サポートされるロケールの一部のみがリストされます。サポートされる他の値をリストに追加するには、<ProductDir>%bin ディレクトリーにある %Data%Std%stdConnProps.xml ファイルを変更します。詳細については、本書の付録『Connector Configurator Express』を参照してください。

コネクタが国際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、そのアダプターのユーザズ・ガイドを参照してください。

デフォルト値は `en_US` です。

LogAtInterchangeEnd

LogAtInterchangeEnd プロパティは、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。

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 ファイルで MESSAGE_RECIPIENT の値として指定された宛先に対する E メール・メッセージが生成されます。例えば、LogAtInterChangeEnd の値を 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。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は false です。

MaxEventCapacity

MaxEventCapacity プロパティは、コントローラー・バッファ内のイベントの最大数を指定します。このプロパティは、フロー制御機能によって使用されます。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

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

MessageFileName プロパティは、コネクタ・メッセージ・ファイルの名前を指定します。メッセージ・ファイルの標準位置は、製品ディレクトリーの `¥connectors¥messages` です。メッセージ・ファイルが標準位置に格納されていない場合は、メッセージ・ファイル名を絶対パスで指定します。

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

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

デフォルト値は `InterchangeSystem.txt` です。

MonitorQueue

MonitorQueue プロパティは、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を指定します。

このプロパティは、DeliveryTransport プロパティの値が JMS で、DuplicateEventElimination の値が true の場合のみ有効です。

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

OADAutoRestartAgent

OADAutoRestartAgent プロパティは、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、WebSphere 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。WebSphere MQ によりトリガーされる OAD 機能の構成方法については、「*WebSphere Business Integration Server Express インストール・ガイド (Windows 版)*」、「*WebSphere Business Integration Server Express インストール・ガイド (Linux 版)*」、または「*WebSphere Business Integration Server Express インストール・ガイド (i5/OS 版)*」を参照してください。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は false です。

OADMaxNumRetry

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

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は 1000 です。

OADRetryTimeInterval

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

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は 10 です。

PollEndTime

PollEndTime プロパティは、イベント・キューのポーリングを停止する時刻を指定します。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、値を含まない HH:MM であるため、この値は必ず変更する必要があります。

アダプター・ランタイムが以下のことを検出した場合、

- PollStartTime が設定されて、PollEndTime が設定されていない、または
- PollEndTime が設定されて、PollStartTime が設定されていない

PollFrequency プロパティに構成された値を使用してポーリングします。

PollFrequency

PollFrequency プロパティは、あるポーリング・アクションの終了から次のポーリング・アクションの開始までの時間をミリ秒単位で指定します。これはポーリング・アクション間の間隔ではありません。この論理を次に説明します。

- ポーリングし、PollQuantity プロパティの値により指定される数のオブジェクトを取得します。
- これらのオブジェクトを処理します。一部のコネクタでは、これは個別のスレッドで部分的に実行されます。これにより、次のポーリング・アクションまで処理が非同期に実行されます。

- PollFrequency プロパティで指定された間隔にわたって遅延します。
- このサイクルを繰り返します。

このプロパティでは、以下の値が有効です。

- ポーリング・アクション間のミリ秒数 (正整数)。
- 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。
- 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。

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。このようなコネクタが存在する場合には、アダプタのインストールと構成に関する章で制約事項が説明されています。

PollQuantity

PollQuantity プロパティは、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。

このプロパティは、DeliveryTransport プロパティの値が JMS で、ContainerManagedEvents プロパティに値がある場合のみ有効です。

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

- 一度ポーリングされると、コネクタはメッセージの本文を検出し、それを添付ファイルとして読み取ります。本文の MIME タイプにはデータ・ハンドラーが指定されていないので、コネクタはメッセージを無視します。
- コネクタは BO の最初の添付を処理します。この MIME タイプには対応するデータ・ハンドラーがあるので、コネクタはビジネス・オブジェクトを Visual Test Connector に送信します。
- 二度目にポーリングされると、コネクタは 2 番目の BO 添付ファイルを処理します。この MIME タイプには対応するデータ・ハンドラーがあるので、コネクタはビジネス・オブジェクトを Visual Test Connector に送信します。
- それが受け入れられると、3 番目の BO 添付ファイルが送信されます。

PollStartTime

PollStartTime プロパティは、イベント・キューのポーリングを開始する時刻を指定します。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、値を含まない HH:MM であるため、この値は必ず変更する必要があります。

アダプタ・ランタイムが以下のことを検出した場合、

- PollStartTime が設定されて、PollEndTime が設定されていない、または
 - PollEndTime が設定されて、PollStartTime が設定されていない
- PollFrequency プロパティに構成された値を使用してポーリングします。

RepositoryDirectory

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

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

統合ブローカーが WebSphere Message Broker または WAS の場合は、この値はデフォルトで <ProductDir>¥repository に設定されます。ただし、これには任意の有効なディレクトリー名を設定することができます。

RequestQueue

RequestQueue プロパティは、統合ブローカーが、ビジネス・オブジェクトをコネクタに送信するときに使用されるキューを指定します。

このプロパティは、DeliveryTransport プロパティの値が JMS の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/REQUESTQUEUE です。

ResponseQueue

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

このプロパティは、DeliveryTransport プロパティの値が JMS の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/RESPONSEQUEUE です。

RestartRetryCount

RestartRetryCount プロパティは、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に接続された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クライアント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

RestartRetryInterval プロパティは、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にリンクされた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クライアント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。

プロパティに使用可能な値の範囲は 1 から 2147483647 です。

デフォルト値は 1 です。

RHF2MessageDomain

RHF2MessageDomain プロパティを使用すると、JMS ヘッダーにあるフィールド・ドメイン名の値を構成できます。JMS トランスポートを介してデータを WebSphere Message Broker に送信するときに、アダプター・フレームワークにより JMS ヘッダー情報、ドメイン名、および固定値 mrm が書き込まれます。構成可能ドメイン名によって、WebSphere Message Broker がメッセージ・データを処理する方法を追跡できます。

ヘッダーの例を示します。

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

BrokerType の値が ICS の場合、このプロパティは無効です。また、このプロパティは、DeliveryTransport プロパティの値が JMS で、WireFormat プロパティの値が CwXML の場合のみ有効です。

可能な値は、mrm および xml です。デフォルト値は mrm です。

SourceQueue

SourceQueue プロパティは、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、52 ページの『ContainerManagedEvents』を参照してください。

このプロパティは、DeliveryTransport の値が JMS で、ContainerManagedEvents の値が指定されている場合のみ有効です。

デフォルト値は <CONNECTORNAME>/SOURCEQUEUE です。

SynchronousRequestQueue

SynchronousRequestQueue プロパティは、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、同期要求キューにメッセージを送信し、同期応答キューでブローカーからの応答を待機します。コネクタに送信される応答メッセージには、元のメッセージの ID を指定する関連 ID が含まれています。

このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/SYNCHRONOUSREQUESTQUEUE です。

SynchronousRequestTimeout

SynchronousRequestTimeout プロパティは、コネクタが同期要求への応答を待機する時間をミリ秒単位で指定します。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 (およびエラー・メッセージ) を障害キューに移動します。

このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

デフォルト値は 0 です。

SynchronousResponseQueue

SynchronousResponseQueue プロパティは、同期要求に対する応答メッセージを、ブローカーからコネクタ・フレームワークにデリバリーします。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。

このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

デフォルトは <CONNECTORNAME>/SYNCHRONOUSRESPONSEQUEUE です。

TivoliMonitorTransactionPerformance

TivoliMonitorTransactionPerformance プロパティは、IBM Tivoli Monitoring for Transaction Performance (ITMTP) を実行時に起動するかどうかを指定します。

デフォルト値は false です。

WireFormat

WireFormat プロパティは、トランスポートのメッセージ・フォーマットを指定します。

- RepositoryDirectory プロパティの値がローカル・ディレクトリーの場合、値は CwXML です。
- RepositoryDirectory プロパティの値がリモート・ディレクトリーの場合、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。

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
- 構成ファイルを作成する
- 構成ファイル内のプロパティを設定する

この付録では、次のトピックについて説明します。

- 『Connector Configurator Express の概要』
- 67 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』
- 70 ページの『新規構成ファイルの作成』
- 74 ページの『構成ファイル・プロパティの設定』

Connector Configurator Express の概要

Connector Configurator Express によって、InterChange Server Express 統合ブローカーで使用するアダプターのコネクタ・コンポーネントを構成することができます。

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**を作成する。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、InterChange Server Express の場合はコラボレーションとともに使用するマップを指定し、必要に応じてメッセージング、ロギング、トレース、およびデータ・ハンドラー・パラメーターを指定する必要があります。

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に必要なプロパティ) とが含まれます。

標準プロパティはすべてのコネクタにより使用されるので、標準プロパティを最初から必要はありません。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、67 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。

Linux でのコネクタの実行

Connector Configurator Express は、Windows 環境内でのみ実行されます。Linux 環境でコネクタを実行する場合は、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 Linux 環境へコピーします。

Connector Configurator Express 内のいくつかのプロパティはディレクトリー・パスを使用します。このパスは、Windows のディレクトリー・パスの規則がデフォルトになっています。Linux 環境で構成ファイルを使用する場合、これらのパスの Linux の規則に対応するように、ディレクトリー・パスを修正する必要があります。正しいオペレーティング・システム規則が拡張検証に使用されるように、ツールバー・ドロップ・リストでターゲット・オペレーティング・システムを選択します。

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 の実行

どのブローカーを実行している場合にも、System Manager を実行せずに Connector Configurator Express を実行し、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」>「すべてのプログラム」から、「**IBM WebSphere Business Integration Express**」>「**Toolset Express**」>「開発」>「**Connector Configurator Express**」をクリックします。
- 「ファイル」>「新規」>「コネクタ構成」を選択します。
- 「システム接続: **Integration Broker**」の隣のプルダウン・メニューをクリックして、ICS 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 (73 ページの『構成ファイルの完成』を参照)。

System Manager からの Configurator の実行

System Manager から Connector Configurator 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、「コネクター」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、「新規コネクター」ダイアログ・ボックスが表示されます。
4. 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックして、ICS を選択します。

既存の構成ファイルを編集するには、以下のステップを実行します。

- 「System Manager」ウィンドウの「コネクター」フォルダーでいずれかの構成ファイルを選択し、右クリックします。Connector Configurator Express が開き、この構成ファイルの統合ブローカー・タイプおよびファイル名が上部に表示されます。
- Connector Configurator Express で「ファイル」>「開く」を選択します。プロジェクトまたはプロジェクトが保管されているディレクトリーからコネクター構成ファイルを選択します。
- 「標準のプロパティ」タブをクリックし、この構成ファイルに含まれているプロパティを確認します。

コネクター固有のプロパティ・テンプレートの作成

コネクターの構成ファイルを作成するには、コネクター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。

コネクター固有プロパティのテンプレートを新規に作成するか、または既存のコネクター定義をテンプレートとして使用します。

- テンプレートの新規作成については、『新規テンプレートの作成』を参照してください。
- 既存のファイルを使用する場合には、既存のテンプレートを変更し、新しい名前でのこのテンプレートを保管します。既存のテンプレートは `¥ProductDir¥bin¥Data¥App` ディレクトリーにあります。

新規テンプレートの作成

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

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

1. 「ファイル」>「新規」>「コネクタ固有プロパティ・テンプレート」をクリックします。
2. 「コネクタ固有プロパティ・テンプレート」 ダイアログ・ボックスが表示されます。
 - 「新規テンプレート名を入力してください」の下の「名前」フィールドに、新規テンプレートの名前を入力します。テンプレートから新規構成ファイルを作成するためのダイアログ・ボックスを開くと、この名前が再度表示されます。
 - テンプレートに含まれているコネクタ固有のプロパティ定義を調べるには、「テンプレート名」表示でそのテンプレートの名前を選択します。そのテンプレートに含まれているプロパティ定義のリストが「テンプレートのプレビュー」表示に表示されます。
3. テンプレートを作成するときには、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。ご使用のコネクタで使用するコネクタ固有のプロパティが表示されるテンプレートが見つからない場合は、自分で作成する必要があります。
 - 既存のテンプレートを変更する場合には、「変更する既存のテンプレートを選択してください: 検索テンプレート」の下の「テンプレート名」テーブルのリストから、テンプレート名を選択します。
 - このテーブルには、現在使用可能なすべてのテンプレートの名前が表示されます。テンプレートを検索することもできます。

一般特性の指定

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

- **一般:**
 - プロパティ・タイプ
 - プロパティ・サブタイプ
 - 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・タイプ」がストリングの場合、「プロパティ・サブタイプ」を選択できます。これは、構成ファイルの保管時に構文検査を提供するオプションの値です。デフォルトは空白・スペースで、プロパティのサブタイプが指定されていないことを意味します。

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」のフィールドに値を入力します。

新規プロパティ値を作成するには、以下のステップを実行します。

1. 「値」列見出しの左側の正方形を右マウス・ボタンでクリックします。
2. ポップアップ・メニューから「追加」を選択して、「プロパティ値」ダイアログ・ボックスを表示します。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。
3. 新規プロパティ値を入力し、「OK」をクリックします。右側の「値」パネルに値が表示されます。

「値」パネルには、3つの列からなるテーブルが表示されます。

「値」の列には、「プロパティ値」ダイアログ・ボックスで入力した値と、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。

パス名の設定

パス名の設定の一般的な規則のいくつかを以下に示します。

- Windows および Linux でのファイル名の最大長は 255 文字です。
- Windows では、絶対パス名は [Drive:][Directory]%filename の形式に従う必要があります。例えば、C:%WebSphereAdapters%bin%Data%Std%StdConnProps.xml のようにします。
Linux では、最初の文字は / でなければなりません。
- キュー名では、先頭または途中にスペースを使用することはできません。

新規構成ファイルの作成

構成ファイルを新規に作成するには、構成ファイルの名前を指定し、統合ブローカーを選択する必要があります。

ファイルの拡張検証のために、オペレーティング・システムも選択します。ツールバーには「ターゲット・システム」というドロップ・リストがあり、ここで、プロパティの拡張検証用のターゲット・オペレーティング・システムを選択できます。選択可能なオプションは、「Windows」、「Linux」、および「i5/OS」、「その他」(Windows でも Linux でもない場合)、および「なし (拡張検証なし)」(拡張検証をオフに切り替え) です。始動時のデフォルトは「Windows」です。

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

- 「System Manager」ウィンドウで、「ツール」メニューから「**Connector Configurator Express**」を選択します。Connector Configurator Express が開きます。
- スタンドアロン・モードで、Connector Configurator Express を起動します。

構成ファイルの拡張検証用のオペレーティング・システムを設定するには、以下のステップを実行します。

- メニュー・バーの「ターゲット・システム:」ドロップ・リストをプルダウンします。
- 使用中のオペレーティング・システムを選択します。

次に、「ファイル」>「新規」>「コネクタ構成」を選択します。「新規コネクタ」ウィンドウで、新規コネクタの名前を入力します。

また、統合ブローカーも選択する必要があります。選択したブローカーによって、構成ファイルに記述されるプロパティが決まります。ブローカーを選択するには、以下のステップを実行します。

- 「**Integration Broker**」フィールドで、ICS を選択します。
- この章で後述する説明に従って「新規コネクタ」ウィンドウの残りのフィールドに入力します。

コネクタ固有のテンプレートからの構成ファイルの作成

コネクタ固有のテンプレートを作成すると、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. メニュー・バーの「ターゲット・システム:」ドロップ・リストを使用して、構成ファイルの拡張検証用のオペレーティング・システムを設定します (前述の『新規構成ファイルの作成』を参照してください)。
2. 「ファイル」>「新規」>「コネクタ構成」をクリックします。
3. 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ず。

- **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**

「ICS」をクリックします。

- **コネクタ固有プロパティ・テンプレートを選択 (Select Connector-Specific Property Template)**

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。

4. 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。

5. 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している必要があります。

6. 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

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は、特定のコネクタのプロパティと、適用可能なデフォルト値がリストされたテキスト・ファイルです。コネクタの配布パッケージの %repository ディレクトリー内には、このようなファイルが格納されていることがあります (通常、このファイルの拡張子は .txt です。例えば、XML コネクタの場合は CN_XML.txt です)。
- ICS リポジトリ・ファイル。コネクタの以前の ICS インプリメンテーションで使用した定義は、そのコネクタの構成で使用されたりリポジトリ・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、ディレクトリーから *.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)
 - ICS リポジトリ (*.in、*.out)

ICS 環境でのコネクタの構成にリポジトリ・ファイルが使用された場合には、このオプションを選択します。リポジトリ・ファイルに複数のコネクタ定義が含まれている場合は、ファイルを開くとすべての定義が表示されません。

- すべてのファイル (*.*)

コネクタのアダプター・パッケージに *.txt ファイルが付属していた場合、または別の拡張子で定義ファイルが使用可能である場合は、このオプションを選択します。

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

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

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

構成ファイルの完成

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

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

1. 「標準のプロパティ」タブで、BrokerType プロパティの値フィールドを選択します。ドロップダウン・メニューで、値 ICS を選択します。
2. 選択したブローカーに関連付けられているコネクタ・プロパティが「標準のプロパティ」タブに表示されます。表に、「プロパティ名」、「値」、「タイプ」、「サブタイプ」（「タイプ」がstringである場合）、「説明」、および「更新メソッド」が表示されます。
3. ここでファイルを保管するか、または 77 ページの『サポートされるビジネス・オブジェクト定義の指定』の説明に従い残りの構成フィールドに値を入力することができます。
4. 構成が完了したら、「ファイル」>「保管」>「プロジェクトに」を選択するか、または「ファイル」>「保管」>「ファイルに」を選択します。

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

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

構成ファイルを作成する前に、プロパティの拡張検証用のターゲット・オペレーティング・システムを選択することができる「ターゲット・システム」ドロップ・リストを使用します。

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

「ターゲット・システム」ドロップ・リストから「Windows」、「Linux」、および「i5/OS」、または「その他」を選択することによって拡張検証機能を使用する場合、システムはタイプだけでなくプロパティ・サブタイプを検証し、検証に失敗した場合は警告メッセージを表示します。

構成ファイル・プロパティの設定

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

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

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

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

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

- 関連付けられたマップ
- セキュリティー

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

標準プロパティとコネクタ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すべてではなく一部のみです。

- アプリケーション固有のプロパティは、コネクタのアプリケーション固有コンポーネント (アプリケーションと直接対話するコンポーネント) のみに適用されます。各コネクタには、そのコネクタのアプリケーションだけで使用されるアプリケーション固有のプロパティがあります。これらのプロパティには、デフォルト値が用意されているものもあれば、そうでないものもあります。また、一部のデフォルト値は変更することができます。各アダプター・ガイドのインストールおよび構成の章に、アプリケーション固有のプロパティおよび推奨値が記述されています。

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

- 背景がグレーのフィールドは、標準のプロパティを表します。値を変更することはできますが、名前の変更およびプロパティの除去はできません。
- 背景が白のフィールドは、アプリケーション固有のプロパティを表します。これらのプロパティは、アプリケーションまたはコネクタの特定のニーズによって異なります。値の変更も、これらのプロパティの除去も可能です。
- 「値」フィールドは構成できます。
- プロパティごとに「更新メソッド」フィールドが表示されます。これは、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。この設定を構成することはできません。

標準コネクタ・プロパティの設定

標準のプロパティの値を変更するには、以下の手順を実行します。

1. 値を設定するフィールド内でクリックします。
2. 値を入力するか、ドロップダウン・メニューが表示された場合にはメニューから値を選択します。

注: プロパティの「タイプ」が「string」である場合、「サブタイプ」列にサブタイプ値が含まれている場合があります。このサブタイプは、プロパティの拡張検証に使用されます。

3. 標準のプロパティの値をすべて入力後、以下のいずれかを実行することができます。
 - 変更内容を破棄し、元の値を保持したままで Connector Configurator Express を終了するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「いいえ」をクリックします。
 - Connector Configurator Express 内の他のカテゴリーの値を入力するには、そのカテゴリーのタブを選択します。「標準のプロパティ」 (またはその他のカテゴリー) で入力した値は、次のカテゴリーに移動しても保持されます。ウィンドウを閉じると、すべてのカテゴリーで入力した値を一括して保管するかまたは破棄するかを確認するプロンプトが出されます。
 - 修正した値を保管するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出さ

れたら「はい」をクリックします。「ファイル」メニューまたはツールバーから「保管」>「ファイルに」をクリックする方法もあります。

特定の標準プロパティーに関する詳細を参照するには、「標準のプロパティー」タブ付きシート内のそのプロパティーの「説明」列内の項目を左マウス・ボタンでクリックします。全般ヘルプをインストール済みの場合は、右側に矢印ボタンが表示されます。ボタンをクリックすると、「ヘルプ」ウィンドウが開き、標準プロパティーの詳細が表示されます。

注: ホット・ボタンが表示されない場合、そのプロパティーについては全般ヘルプが見つかりません。

インストール済みの場合、全般ヘルプ・ファイルは
<ProductDir>%bin%Data%Std%Help%<RegionalSetting>% にあります。

コネクタ固有の構成プロパティーの設定

コネクタ固有の構成プロパティーの場合、プロパティー名の追加または変更、値の構成、プロパティーの削除、およびプロパティーの暗号化が可能です。プロパティー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。プロパティーを追加するときは「追加」をクリックします。子プロパティーを追加するには、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. プロパティーまたは子プロパティーの値を入力します。

注: プロパティーの「タイプ」が「string」である場合、「サブタイプ」ドロップ・リストからサブタイプを選択できます。このサブタイプは、プロパティーの拡張検証に使用されます。

3. プロパティーを暗号化するには、「暗号化」ボックスを選択します。
4. 特定のプロパティーに関する詳細を参照するには、そのプロパティーの「説明」列内の項目を左マウス・ボタンでクリックします。全般ヘルプをインストール済みの場合は、ホット・ボタンが表示されます。ホット・ボタンをクリックすると、「ヘルプ」ウィンドウが開き、標準プロパティーの詳細が表示されます。

注: ホット・ボタンが表示されない場合、そのプロパティーについては全般ヘルプが見つかりません。

5. 75 ページの『標準コネクタ・プロパティーの設定』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。

全般ヘルプ・ファイルがインストール済みで、AdapterHelpName プロパティーがブランクである場合、Connector Configurator Express は、<ProductDir>%bin%Data%App%Help%<RegionalSetting>% にあるアダプター固有の全般ヘルプ・ファイルを指します。それ以外の場合、Connector Configurator Express は、<ProductDir>%bin%Data%App%Help%<AdapterHelpName>%<RegionalSetting>%にあるアダプター固有の全般ヘルプ・ファイルを指します。標準プロパティーについての付録で説明されている AdapterHelpName プロパティーを参照してください。

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

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

標準プロパティについての付録の『標準プロパティ値の概要』にある更新メソッドの説明を参照してください。

サポートされるビジネス・オブジェクト定義の指定

コネクターで使用するビジネス・オブジェクトを指定するには、Connector Configurator 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。

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。

ご使用のブローカーが InterChange Server Express の場合

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

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

1. 「**ビジネス・オブジェクト名**」リスト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・リストが表示されます。
2. 追加するビジネス・オブジェクトをクリックします。
3. ビジネス・オブジェクトの「**エージェント・サポート**」(以下で説明)を設定します。
4. 「Connector Configurator Express」ウィンドウの「ファイル」メニューで、「**プロジェクトに保管**」をクリックします。追加したビジネス・オブジェクト定義に指定されたサポートを含む、変更されたコネクター定義が、System Manager の ICL (Integration Component Library) プロジェクトに保管されます。

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

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

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

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

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

ビジネス・オブジェクトがコネクター・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。

- **明示的バインディング**

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、各コネクタでサポートされるそれぞれのビジネス・オブジェクトにマップを自動的にバインドしようとしています。複数のマップでその入力データとして同一のビジネス・オブジェクトが使用されている場合、サーバーは、他のマップのスーパーセットである 1 つのマップを見つけて、バインドしようとしています。

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

以下の手順を実行して、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、サーバーをリブートします。

セキュリティ

Connector Configurator Express 内の「セキュリティ」タブを使用して、メッセージにさまざまなプライバシー・レベルを設定することができます。DeliveryTransport プロパティが JMS に設定されている場合のみ、この機能を使用できます。

デフォルトでは、「プライバシー」はオフになっています。使用可能にするには、「プライバシー」ボックスにチェック・マークを付けます。

「鍵ストア・ターゲット・システムの絶対パス名」は、以下の値です。

- Windows の場合:
`<ProductDir>%connectors%security%<connectorname>.jks`
- Linux および i5/OS の場合:
`/ProductDir/connectors/security/<connectorname>.jks`

このパスおよびファイルは、コネクタを始動するシステム、すなわちターゲット・システム上に存在していなければなりません。

ターゲット・システムが現在実行中のシステムである場合のみ、右側の「参照」ボタンを使用できます。「プライバシー」が使用可能であり、メニュー・バーの「ターゲット・システム」が Windows に設定されている場合を除き、これはグレーアウトされています。

「メッセージのプライバシー・レベル」は、3 つのメッセージ・カテゴリ (全メッセージ、全管理メッセージ、および全ビジネス・オブジェクト・メッセージ) で以下のように設定されます。

- “”。これがデフォルトです。メッセージ・カテゴリにプライバシー・レベルが設定されていない場合に使用します。
- none。デフォルトと同じではありません。メッセージ・カテゴリにプライバシー・レベルなしと故意に設定する場合にこれを使用します。
- integrity
- privacy
- integrity_plus_privacy

「鍵の保守」機能によって、サーバーおよびアダプターの公開鍵を生成、インポート、およびエクスポートすることができます。

- 「鍵の生成」を選択すると、鍵を生成する keytool のデフォルトを含む「鍵の生成」ダイアログ・ボックスが表示されます。
- 「セキュリティ」タブの「鍵ストア・ターゲット・システムの絶対パス名」で入力した値が、鍵ストア値のデフォルトになります。
- 「OK」を選択すると、記入項目が検証され、鍵証明書が生成され、「Connector Configurator Express」ログ・ウィンドウに出力が送られます。

証明書をアダプター鍵ストアにインポートする前に、サーバー鍵ストアからエクスポートする必要があります。「アダプター公開鍵のエクスポート」を選択すると、「アダプター公開鍵のエクスポート」ダイアログ・ボックスが表示されます。

- エクスポート証明書のデフォルトは、ファイル拡張子が <filename>.cer であることを除き、鍵ストアと同じ値です。

「サーバー公開鍵のインポート」を選択すると、「サーバー公開鍵のインポート」ダイアログ・ボックスが表示されます。

- インポート証明書のデフォルトは、<ProductDir>%bin%ics.cer になります (システムにファイルが存在する場合)。
- インポート証明書関連はサーバー名でなければなりません。サーバーが登録されていれば、ドロップ・リストからそれを選択することができます。

DeliveryTransport の値が IDL の場合のみ、「アダプター・アクセス制御」機能が使用可能です。デフォルトでは、アダプターはゲスト ID を使用してログインします。「ゲスト ID の使用」ボックスにチェック・マークが付けられていない場合は、「アダプター ID」および「アダプター・パスワード」フィールドが使用可能です。

トレース/ログ・ファイル値の設定

コネクタ構成ファイルまたはコネクタ定義ファイルを開くと、Connector Configurator 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 表示に書き込みます。

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

- ファイルに: ログ・メッセージまたはトレース・メッセージを指定し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、指定する格納場所に移動し、ファイル名を指定し、「保管」をクリックします。ログ・メッセージまたはトレース・メッセージは、指定した場所の指定したファイルに書き込まれます。

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

データ・ハンドラー

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

これらのプロパティーに使用する値については、『付録 A. コネクターの標準構成プロパティー』の `ContainerManagedEvents` の下の説明を参照してください。

構成ファイルの保管

コネクターの構成が完了したら、コネクター構成ファイルを保管します。`Connector Configurator Express` では、構成中に選択したブローカー・モードでファイルを保管します。`Connector Configurator Express` のタイトル・バーには、`InterChange Server Express` が現在使用しているブローカー・モードが常に表示されます。

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

- `System Manager` から、統合コンポーネント・ライブラリーに `*.con` 拡張子付きファイルとして保管します。
- 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。デフォルトでは、このファイルは `¥WebSphereAdapters¥bin¥Data¥App` に保管されます。

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

構成ファイルの変更

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

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

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

- `Connector Configurator Express` で既存の構成ファイルを開きます。
- 「標準のプロパティー」タブを選択します。

- 「標準のプロパティ」タブの「**BrokerType**」フィールドで、ご使用のブローカーに合った値を選択します。現行値を変更すると、プロパティ・ウィンドウ内の利用可能なタブおよびフィールド選択がただちに變更され、選択した新規ブローカーに適したタブとフィールドのみが表示されます。

構成の完了

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

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

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

Connector Configurator Express はグローバル化され、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、以下に太字で示した行を追加してください。

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

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

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	カーディナリティー	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	文書の最終変更日時。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

DominoItem 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	文書の最終変更日時。
Name	String	1	項目の名前。必須。
型	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 18
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
 イベント状況 33
イベント表, 構成 15
イベント表, コピー 14
イベント表, Lotus Domino Server にコピー 13
イベント表の構成 15
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 7
インストールのステップ 7
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, ログイン 33
オペレーティング・システム, サポートされる 8

[カ行]

カードディナリティ, 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
規則, 表記上の v
構成 16
 アダプター 17
 イベント表 15
 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 16
 Create イベント 16
 DHOP 12
 HTTP 12
 IOP 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
 Update イベント 16
構成, アダプターの 7
構成ビュー 3
構成プロパティ 18
 ApplicationPassword 19
 DocumentBOName 19
 DominoServerName 19
 EventDBName 19
 ForceDelete 19
 ForceSave 19
 InDoubtEvents 19
 MakeResponse 19
構成プロパティ, アプリケーション固有の 18
構成プロパティ, 標準 18
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 14
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 17

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 4
選択, データベース 16
属性, ビジネス・オブジェクトの 27, 85

[タ行]

データベース, 選択 16
ディレクトリ, Lotus Domino Server 上の 13
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 85
 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 2
ビュー, イベント表のデータの 2
表記上の規則 v
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 14
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
プロパティ、構成 18
プロパティ、構成、アプリケーション固
有の 18

[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 4

A

ApplicationPassword 19
ApplicationUserName、Lotus Domino
Server で構成 12

C

Create イベント 16

D

Delete イベント 16
Delete イベント、構成 16
DocumentBOName 19
DominoDocument ビジネス・オブジェクト
28
DominoItem 28
DominoItemValue 28
DominoServerName 19

E

EventDBName 19

F

ForceDelete 19
ForceSave 19

H

HTTP、構成 12

I

IIOP、構成 12
InDoubtEvents 19
InDoubtEvents、値 20
Internet Interoperability Protocol、構成
12

J

Java ランタイム、必要なバージョン 8
Java/Javascript/COM の構成 12
JRE、必要なバージョン 8

L

Lotus Domino Administrator、変更を加え
る 11
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、サポートされるバ
ージョン 8
Lotus Domino Server、変更 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 19

N

NCSO.jar 16
NCSO.jar、および始動スクリプト 17
NOTES.INI の構成 12

P

ProductDir、定義 vi

U

Update イベント 16

W

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

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032

東京都港区六本木 3-2-31

*IBM World Trade Asia Corporation
Licensing*

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation

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すが、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にお願いします。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。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。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めめかしたり、保証することはできません。この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。汎用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

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

商標

以下は、IBM Corporation の商標です。

IBM
IBM ロゴ
AIX
CICS
CrossWorlds
DB2
DB2 Universal Database
IMS
Informix
i5/OS
iSeries
Lotus
Lotus Domino
Lotus Notes
MQIntegrator
MQSeries
MVS
OS/400
Passport Advantage
SupportPac
WebSphere
z/OS

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。Linux は、Linus Torvalds の米国およびその他の国における商標です。他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

WebSphere Business Integration Server Express および Express Plus 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 バージョン 4.4 および WebSphere Business Integration Server Express Plus バージョン 4.4



Printed in Japan