

IBM WebSphere Business Integration
Adapters



Adapter for PeopleSoft (バージョン 7.x) ユーザーズ・ ガイド

バージョン 3.5.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、67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、Adapter for PeopleSoft 7.x バージョン 3.5.0 および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： IBM WebSphere Business Integration Adapters
Adapter for PeopleSoft (Version 7.x) User Guide
Version 3.5.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 2004.7

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

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

© Copyright International Business Machines Corporation 2000, 2003. All rights reserved.

© Copyright IBM Japan 2004

目次

本書について	v
対象読者	v
関連資料	v
表記上の規則	vi
本リリースの新機能	vii
リリース 3.5.x の新機能	vii
リリース 3.4.x の新機能	vii
リリース 3.3.x の新機能	viii
第 1 章 コネクターの概要	1
コネクタと PeopleSoft 間の通信	1
第 2 章 コネクターのインストールと構成	3
アダプター環境	3
前提条件	4
コネクターのインストール	5
必要な変更	5
イベント表の構造	8
コネクターの構成	9
コネクターの複数バージョンの作成	11
コネクターの開始	13
コネクターの停止	14
アプリケーションへの接続の切断	15
第 3 章 コネクタ用のビジネス・オブジェクトの開発	17
ビジネス・オブジェクト・アプリケーション固有のテキスト	17
コネクタ用のビジネス・オブジェクトの変更	19
階層ビジネス・オブジェクトのトランザクション処理	22
ビジネス・オブジェクト定義ファイルのサンプル	22
BO_PsftEmployee ビジネス・オブジェクト	23
付録 A. コネクターの標準構成プロパティ	29
新規プロパティと削除されたプロパティ	29
標準コネクタ・プロパティの構成	29
標準プロパティの要約	31
標準構成プロパティ	35
付録 B. Connector Configurator	49
Connector Configurator の概要	49
Connector Configurator の始動	50
System Manager からの Configurator の実行	51
コネクタ固有のプロパティ・テンプレートの作成	51
新しい構成ファイルを作成	54
既存ファイルの使用	55
構成ファイルの完成	57
構成ファイル・プロパティの設定	57
構成ファイルの保管	64
構成ファイルの変更	65
構成の完了	65

グローバル化環境における Connector Configurator の使用	66
特記事項	67
プログラミング・インターフェース情報	69
商標	69

本書について

IBM[®] WebSphere[®] Business Integration Adapter ポートフォリオは、主要な e-business テクノロジー、エンタープライズ・アプリケーション、およびレガシー・システムとメインフレーム・システムに、統合接続性を提供します。製品セットには、ビジネス・プロセスの統合に向けてコンポーネントをカスタマイズ、作成、および管理するためのツールとテンプレートが含まれています。

本書では、IBM WebSphere Business Integration Adapter for PeopleSoft 7.x のインストール、構成、およびビジネス・オブジェクト開発について説明します。

対象読者

本書は、IBM WebSphere Business Integration Adapter Framework のコンサルタントおよびお客様を対象としています。本書では、以下の領域の知識があることを前提としています。

- コネクターの開発
- ビジネス・オブジェクトの開発
- PeopleSoft アプリケーションのアーキテクチャー
- PeopleSoft ツール

関連資料

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

以下のサイトから、関連資料をインストールすることができます。

- アダプターの一般情報、WebSphere Message Brokers (WebSphere MQ Integrator、WebSphere MQ Integrator Broker、WebSphere Business Integration Message Broker) におけるアダプターの使用、および WebSphere Application Server におけるアダプターの使用については、次のアドレスの IBM WebSphere Business Integration Adapters InfoCenter を参照してください。
<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- WebSphere InterChange Server におけるアダプターの使用については、次のアドレスの IBM WebSphere InterChange Server InfoCenters を参照してください。
<http://www.ibm.com/websphere/integration/wicserver/infocenter>
<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>
- WebSphere Message Brokers の詳細については、以下を参照してください。
<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>
- WebSphere Application Server の詳細については、以下を参照してください。
<http://www.ibm.com/software/webservers/appserv/library.html>

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

表記上の規則

本書は下記の規則に従って編集されています。

Courier フォント	コマンド名、ファイル名、入力情報、システムが画面に出力した情報など、記述されたとおりの値を示します。
太字	初出語を示します。 変数名または相互参照を示します。
青い文字	オンラインで表示したときのみ見られる青の部分は、相互参照用のハイパーリンクです。青い文字ストリングをクリックすることにより、参照先オブジェクトに飛ぶことができます。

本リリースの新機能

リリース 3.5.x の新機能

2004 年 6 月更新。

このアダプターは、PeopleTools version 7.63 をサポートしています。

リリース 3.4.x の新機能

2003 年 12 月更新。

アダプターのインストール情報は、本書から移動しました。この情報の新たな入手先については、第 2 章を参照してください。

Adapter for PeopleSoft 7.x は Microsoft Windows NT 上ではサポートされなくなりました。

2003 年 3 月更新。

「CrossWorlds」という名前は、現在ではシステム全体を表したり、コンポーネント名やツール名を修飾するためには使用されなくなりました。コンポーネント名およびツール名自体は、以前とほとんど変わりません。例えば、「CrossWorlds System Manager」は現在では「System Manager」となり、「CrossWorlds InterChange Server」は「WebSphere InterChange Server」となっています。

IBM WebSphere Business Integration Adapter for PeopleSoft 7.x には、PeopleSoft 7.x 用のコネクタが組み込まれています。このアダプターは、InterChange Server (ICS) と WebSphere MQ Integrator のどちらの統合ブローカーでも動作します。統合ブローカーとは、異種のアプリケーション・セット間の統合を実行するアプリケーションです。統合ブローカーは、データ・ルーティングなどのサービスを提供します。

このアダプターには、以下の要素が含まれます。

- PeopleSoft に固有のアプリケーション・コンポーネント。
- PeopleSoft ODA。
- サンプル・ビジネス・オブジェクト。¥connectors¥PeopleSoft7¥samples ディレクトリにあります。
- IBM WebSphere Adapter フレームワーク。コンポーネントは以下のとおりです。
 - コネクタ・フレームワーク
 - 開発ツール (Business Object Designer と Connector Configurator を含む)
 - API (ODK、JCDK、および CDK を含む)

本書では、このアダプターを InterChange Server (ICS) と WebSphere MQ Integrator の両方の統合ブローカーと一緒に使用する場合の情報を提供します。

重要: コネクターは国際化に対応していないため、ISO Latin-1 データのみが処理されることが確実である場合を除いて、コネクターと InterChange Server バージョン 4.1.1 を併用しないでください。

リリース 3.3.x の新機能

PeopleSoft 7.x 用 IBM CrossWorlds コネクターのバージョン 3.3.1 の新機能および変更点は、以下のとおりです。

- Message Agent キューに、`keylist` と `poll_delay` という 2 つの新しいフィールド・マッピングが追加されました。

第 1 章 コネクターの概要

この章では、IBM WebSphere Business Integration Adapter for PeopleSoft 7.x のコネクター・コンポーネントについて説明します。このコンポーネントを使用することで、サポートされている統合ブローカーは、PeopleSoft 7.55 および 7.57 アプリケーションとビジネス・オブジェクトを交換することが可能になります。

コネクターは、コネクター・フレームワークとアプリケーション固有のコンポーネントの 2 つの部分から成り立ちます。コネクター・フレームワークのコードは、すべてのコネクターに共通です。コネクター・フレームワークは、統合ブローカーとアプリケーション固有のコンポーネントの間を中継します。アプリケーション固有のコンポーネントには、特定のアプリケーションに合わせて調整されたコードが入っています。コネクター・フレームワークは、統合ブローカーとアプリケーション固有のコンポーネントの間で以下のサービスを提供します。

- ビジネス・オブジェクトの受信と送信
- 始動メッセージおよび管理メッセージの交換の管理

本書には、コネクター・フレームワークとアプリケーション固有のコンポーネントに関する情報が記載されています。これらのコンポーネントの両方を、本書ではコネクターと呼びます。

統合ブローカーとコネクターの関係の詳細については、「システム管理ガイド」または「WebSphere Business Integration Adapters MQ Integrator 用インプリメンテーション・ガイド」を参照してください。

コネクターを使用できるデータベースは、Informix、SQLBase、Oracle、Sybase、Microsoft、DB2、DB2ODBC、DB2MDI、DB2400、DB2Unix です。

コネクターと PeopleSoft 間の通信

ビジネス・オブジェクトの操作

コネクターは、PeopleSoft Message Agent API と各ビジネス・オブジェクトのメッセージ定義セットを使用して、データを PeopleSoft に挿入します。

イベント通知

イベント通知では、コネクターは、統合ブローカーに関係のあるイベントが PeopleSoft で発生したときに、PeopleCode を使用してイベント・キュー表を取り込みます。コネクターは、構成可能な間隔でイベント・キュー表をポーリングし、PeopleSoft Message Agent API を使用してイベントを検索し、処理して、アーカイブします。

処理されたイベントとアンサブスクライブされたイベントは、PeopleCode によりイベント作業表にアーカイブされます。このコードによって、イベントはイベント表から除去されて、アーカイブ表にコピーされます。イベント表とアーカイブ表は、コネクターがサポートする各データベースのデータベース・アップグレードと共に

インストールされます。イベント・キュー表の詳細については、5 ページの『選択アプリケーション・アップグレードの実行』を参照してください。

第 2 章 コネクタのインストールと構成

この章では、IBM WebSphere Business Integration Adapter for PeopleSoft 7.x をインストールする方法と、コネクタが動作するように PeopleSoft アプリケーションを構成する方法について説明します。この章には、次のセクションがあります。

- 『アダプター環境』
- 4 ページの『前提条件』
- 5 ページの『コネクタのインストール』
- 5 ページの『必要な変更』
- 8 ページの『イベント表の構造』
- 9 ページの『コネクタの構成』
- 11 ページの『コネクタの複数バージョンの作成』
- 13 ページの『コネクタの開始』
- 14 ページの『コネクタの停止』
- 15 ページの『アプリケーションへの接続の切断』

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、環境の要件を把握しておく必要があります。環境要件は、以下のセクションでリストされています。

- 『ブローカーの互換性』
- 4 ページの『アダプターのプラットフォーム』
- 4 ページの『グローバル化セッション』

ブローカーの互換性

アダプターが使用するアダプター・フレームワークは、アダプターと通信する統合ブローカーのバージョンとの互換性を備えている必要があります。Adapter for PeopleSoft 7 のバージョン 3.5.0 は、以下のアダプター・フレームワークおよび統合ブローカーでサポートされます。

- **アダプター・フレームワーク:**
 - WebSphere Business Integration Adapter Framework バージョン 2.1、2.2、2.3.x、および 2.4。
- **統合ブローカー:**
 - WebSphere InterChange Server、バージョン 4.1.1、4.2、4.2.1、4.2.2
 - WebSphere MQ Integrator、バージョン 2.1.0
 - WebSphere MQ Integrator Broker、バージョン 2.1.0
 - WebSphere Business Integration Message Broker、バージョン 5.0

例外については、『リリース情報』を参照してください。

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

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

WebSphere Message Brokers については、「*WebSphere Message Brokers 使用アダプター・インプリメンテーション・ガイド*」を参照してください。

アダプターのプラットフォーム

アダプターは以下のソフトウェアでサポートされています。

オペレーティング・システム:

- Windows 2000

サード・パーティー・ソフトウェア

- PeopleTools 7.55 と 7.57、および PeopleTools 7.63

グローバル化

このアダプターは DBCS (2 バイト文字セット) で使用可能ではなく、翻訳されていません。

前提条件

このセクションでは、コネクターをインストールして実行する前にインストールしておかなければならないソフトウェア・コンポーネントと、行っておかなければならない作業を説明します。

ユーザー・アカウントのセットアップ

PeopleSoft にコネクター用のユーザー・アカウントを作成しなければなりません。ユーザー・アカウントには、有効な PeopleSoft ユーザー名とパスワードを任意で使うことができます。このユーザー・アカウントは、適切なアプリケーション・パネルと PeopleSoft システムの cw イベント・パネルのデータを検索、挿入、更新、削除する特権を持っていないければなりません。例えば、コネクターが顧客データを処理する場合には、コネクターのユーザー・アカウントは、関連するすべての顧客パネルのデータにアクセスし、変更する特権を持っていないければなりません。

ユーザー・アカウントは、タイムアウトすることがないように構成しなければなりません。コネクターは、PeopleSoft サーバーのビジネス・オブジェクトを挿入したり取り出したりする場合に Message Agent を使用するので、コネクターが PeopleSoft システムにアクセスするときには Message Agent が実行されていなければなりません。コネクターがタイムアウトになると、Message Agent はクローズします。その結果、コネクターは、PeopleSoft システムにアクセスするたび、つまりポーリング、検索、作成、更新するたびに、ログにエラー・メッセージを書き込むこととなります。

タイムアウトすることがないようにユーザー・アカウントを構成するには、以下のようになります。

1. People Tools Security Administrator を開きます。

2. ユーザー・アカウントが入っているクラスを開きます。
3. 「タイムアウト時間 (分)(Time-Out Minutes)」グループ・ボックスで「タイムアウトしない (Never Time-Out)」ラジオ・ボタンを選択します。

以前のバージョンからのアップグレード

同じ製品の以前のバージョンからアップグレードする場合、リポジトリ定義ファイル内の構成プロパティが変更されていることがあります。9 ページの『コネクタの構成』を参照してください。

以前にインストールしたコネクタの削除

コネクタの以前のバージョンがインストールされている場合は、そのコネクタと、PeopleSoft 環境にあるそのコネクタのカスタマイズされたコンポーネントを削除してください。これには、サポートしているすべてのメニュー、パネル、レコード、イベント表、アーカイブ表と、IBM WebSphere Business Integration Adaptor ビジネス・プロセスが含まれます。

コネクタのインストール

WebSphere Business Integration Adapter 製品のインストールについては、「*WebSphere Business Integration Adapters* インストール・ガイド」を参照してください。この資料は、次の Web サイトの WebSphere Business Integration Adapters Infocenter にあります。

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

必要な変更

コネクタのインストールには、メニュー定義、パネル定義、レコード定義、およびコネクタのサポートに必要な PeopleCode が収められた「データ・ムーバー (Data Mover)」データ・ファイル・セットが含まれています。これらのファイルを使って、以下のものが作成されます。

- イベント通知を保管するイベント・キュー表と、処理したイベントを保管するアーカイブ表。
- イベント表間でイベントを移動する場合に使用する Application Engine。
- 一時的な値と PeopleCode を保管するための派生作業レコード。
- ビジネス・オブジェクト/動詞の組み合わせごとの PeopleSoft メッセージ定義セット。メッセージ定義は、IBM WebSphere Business Integration Adaptor ビジネス・プロセスの Integration アクティビティ、または ICS を統合ブローカーとして使用している場合は、サポートされているコラボレーションに関連するアクティビティとビジネス・プロセスに保管されます。例えば、Employee オブジェクト/動詞の組み合わせに対するメッセージ定義は、CWEMPLOYEE ビジネス・プロセスの CWEmployee アクティビティにあります。

選択アプリケーション・アップグレードの実行

必要な Business Integration オブジェクトを PeopleSoft データベースにコピーする場合は、選択アプリケーション・アップグレードを実行してください。アップグレード・プロセスでは、ソース・データベース、つまりアプリケーション・アップグ

レード・データベース (AUDB) から PeopleSoft 環境内のターゲット・データベースにオブジェクトがコピーされます。Application Upgrader の詳細については、「PeopleSoft Administration Tools」ガイドを参照してください。

アプリケーションのアップグレードを実行する前に、以下のステップに従って AUDB を作成し、準備してください。

1. アップグレード・データベースとして使用する新しいデータベースを作成します。
2. PSDMT.EXE ファイルをダブルクリックして、PeopleTools Data Mover を立ち上げます。「データ・ムーバー (Data Mover)」ウィンドウが表示されます。
3. アップグレード・データベースに、ファイル
`%ProductDir%\connectors\PeopleSoft\PeopleSoft7\dependencies\DataMover\CWUpgrPrj.dat` をインポートします。
4. 「データ・ムーバー (Data Mover)」で、スクリプト
`%PeopleSoft%\script\auimport.dms` を開きます。
5. 「ファイル」>「スクリプトの実行 (Run Script)」を選択するか、「スクリプトの実行 (Run Script)」ボタンをクリックして、スクリプトを実行します。メッセージ定義データがインポートされます。「正常終了 (Successful Completion)」メッセージが表示されることを確認します。

この時点で、AUDB とターゲット・データベースとの間で比較レポートを実行することをお勧めします。ただし、ターゲット・データベース内にビジネス統合オブジェクトが入っていないことが分かっている場合には、このレポートを省略して、アプリケーションのアップグレードを実行することができます。

アプリケーションのアップグレードの実行手順は、以下のとおりです。

1. AUDB にログオンします。
2. Application Designer で、CWUpgrPrj プロジェクトを開きます。
3. 「ツール」メニューから、「コピー」>「アップグレード」を選択します。
4. 「サインオン (Signon)」をクリックして、ターゲット・データベースにサインオンします。「サインオン (Signon)」画面が表示されます。ターゲット・データベースの名前と、オペレーターの ID およびパスワードを入力します。
5. 「コピー」ダイアログ・ボックスで、「プロジェクトのエクスポート (Export Project)」チェック・ボックスを選択します。リスト・ボックス内の他のすべての項目も選択されていることを確認して、「コピー」をクリックします。

PeopleSoft がソース・アプリケーションの場合は、次の 2 つのステップを省略することができます。

6. イベント表のインストールを行うには、Data Designer に移動し、XR_EVENT、XR_FUTURE_EVENT、TMP_FUTURE_EVENT、XR_ARCHIVE に対し、「SQL 表の作成 (SQL Table Create)」を選択します。
7. Data Designer に移動し、XR_SRCH に対し「SQL ビューの作成 (SQL View Create)」を実行します。
8. PeopleSoft Security Administrator で、コネクターの Operator クラスを開き、このクラスに EVENT_NOTIFICATION メニュー・グループを使用する特権を付与します。

アプリケーションのアップグレードが正常に実行されたことをテストするには、以下のようにします。

1. PeopleSoft のウィンドウをすべて閉じます。
2. アプリケーションからログアウトした後、再度ログインします。
3. 「スタート・メニュー (Start Menu)」を選択し、「イベント通知 (Event Notification)」メニューが表示されることを確認します。

Application Engine のインポート

注: コネクターが PeopleSoft をターゲット・アプリケーションとしてサポートしていない場合は、Application Engine は必要ありません。

Application Engine は、将来のイベント表からイベント表にイベントを移動するバッチ・プロセスです。Application Engine をインポートするには、以下のステップを実行します。

1. 「データ・ムーバー (Data Mover)」ウィンドウで、実動データベースにファイル `%ProductDirS%\connectors\PeopleSoft\PeopleSoft7\dependencies\DataMover\CWAE.dat` をインポートします。
2. 「データ・ムーバー (Data Mover)」で、スクリプト `CWAEIn.dms` を開きます。
3. 「ファイル」>「スクリプトの実行 (Run Script)」を選択するか、「スクリプトの実行 (Run Script)」ボタンをクリックして、スクリプトを実行します。
4. スクリプトが正常に実行されたら、Process Scheduler を開き、毎晩 12 時に `XR_MOVE` が実行されるようにスケジュールします。Process Scheduler の使用の詳細については、「PeopleSoft Administration Tools」ガイドを参照してください。

メッセージ定義のインストール

次に、WebSphere Business Integration Adapter Framework ビジネス・オブジェクトをサポートするメッセージ定義をインポートします。このメッセージ定義により、コネクターは Message Agent API を介して PeopleSoft システムと通信できるようになります。インポート・スクリプト・ファイル `CWBPIn.dms` は、`%ProductDirS%\connectors\PeopleSoft\PeopleSoft7\dependencies\DataMover` にあります。このファイルに入っているスクリプトは、以下のとおりです。

表 1. メッセージ定義スクリプト

スクリプト	メッセージ定義
<code>CWBP.dat</code>	イベント通知メッセージ定義
<code>CWEMPL.dat</code>	従業員メッセージ定義
<code>CWDEPT.dat</code>	部門メッセージ定義
<code>CWEMPL_757.dat</code>	PeopleTools 7.57 を使用する場合の従業員メッセージ定義

PeopleSoft システムにメッセージ定義をインポートするには、以下のステップを実行します。

1. 「データ・ムーバー (Data Mover)」ウィンドウで、サポートする必要があるビジネス・オブジェクトに対し `.dms` スクリプトを開きます。

- 「ファイル」>「スクリプトの実行 (Run Script)」を選択するか、「スクリプトの実行 (Run Script)」ボタンをクリックして、スクリプトを実行します。「正常終了 (Successful Completion)」メッセージが表示されることを確認します。メッセージ定義データがインポートされます。

注: メッセージ定義のインポートに失敗した場合は、delbp.sql を使用して WebSphere Business Integration Adapter Framework メッセージ定義をすべて削除してから、インポート・ステップを繰り返してください。delbp.sql ファイルは %ProductDir%\%connectors%\PeopleSoft\PeopleSoft7\%dependencies\DataMover にあります。

パネルへの PeopleCode の埋め込み

Department ビジネス・オブジェクトと Employee ビジネス・オブジェクトを起動できるようにするには、特定のパネルにフィールドを埋め込まなければなりません。表 2 に、各ビジネス・オブジェクトごとに埋め込む必要があるフィールドを詳しく示します。

表 2. ビジネス・オブジェクト起動フィールド

ビジネス・オブジェクト	目的	フィールド名	パネル	レベル
Department	イベント通知	DEPT_TRIGGER	DEPT_TBLGBL_SBP	レベル 0
Employee	イベント通知	EMPL_PERS_TRIGGER	PERSONAL_DATA2	レベル 0
Employee	イベント通知	EMPL_JOB_TRIGGER	JOB_DATA1	レベル 0
Employee	要求	EMPL_OTHERPHONE	PERSONAL_DATA2	レベル 1 (phone スクロール・バー)

該当するパネルにフィールドを埋め込むには、以下のようになります。

- Panel Designer で、表 2 にリストされている該当ビジネス・オブジェクトに対しパネルを開きます。
- メニュー・バーで、「追加」>「編集ボックス (Edit Box)」を選択します。
- パネルをクリックして、パネルに編集ボックスをドロップします。
- フィールドを右クリックして、ポップアップ・メニューから「レコード (Record)」を選択します。
- 「編集ボックス・レコード (Edit Box Record)」ダイアログ・ボックスで、「レコード名 (Record Name)」に XR_EVENT_WRK と入力し、「フィールド名 (Field Name)」に前述の表の関連するフィールド名を入力します。
- 「OK」をクリックし、「編集ボックス (Edit Box)」をもう一度右クリックします。
- ポップアップ・メニューから「使用 (Use)」を選択します。
- 「非表示 (Invisible)」チェック・ボックスを選択し、「OK」をクリックします。

イベント表の構造

PeopleSoft アプリケーションでイベントが発生すると、そのイベントは XR_EVENT というイベント・キュー表に入ります。コネクタは、構成可能な間隔でこの表をポーリングし、イベントを順次処理します。

以下に、イベント・キュー表とアーカイブ・キュー表の構造を示します。ただし、PROCESS_KEY フィールドがあるのはアーカイブ表だけです。

表3. イベント・キュー表とアーカイブ・キュー表の構造

名前	タイプ	制約	説明
PROCESS_KEY	Datetime	非ヌル	イベント結果セットの固有キー。このキーは、イベントが処理された時点の日時スタンプです。アーカイブ表でのみ使用します。
CWPRIORITY	char 3		イベントの優先順位。
EVENT_KEY	Datetime	非ヌル	イベントの固有キー。このキーは、イベントが処理された時点の日時スタンプです。
OBJ_SETID	char 5		PeopleSoft オブジェクトの TableSetID。
OBJECT_ID	char 18		PeopleSoft オブジェクトの固有 ID。
OBJECT	char 18		PeopleSoft オブジェクトの名前。
DOVERB	char 15		PeopleSoft オブジェクトで取られたアクションの名前。
KEYLIST	char 51		非 1 次オブジェクト・キーのリスト。キーは、name=value の形式で、コロン (;) で区切られてリストされません。
PRCS_FLG	char 3		イベントの状態を示すプロセス・フラグ。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • N = 未処理 • P = 正常に処理済み • NS = サブスクライブされていないイベント • E = エラー

コネクタの構成

コネクタには、標準の構成プロパティとコネクタ固有の構成プロパティの 2 つのタイプの構成プロパティがあります。コネクタを稼働させる前に、これらのプロパティの値を設定してください。

コネクタは、始動時に構成値を取得します。ランタイム・セッション時に、1 つ以上のコネクタ・プロパティの値を変更することができます。AgentTraceLevel などのコネクタ構成プロパティに対する変更は、直ちに有効になります。その他のコネクタ・プロパティに対する変更の場合は、変更後にコンポーネントを再始動するか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 (直ちに有効になる) か静的 (コネクタ・コンポーネントの再始動またはシステムの再始動が必要になる) かを判別するには、統合ブローカーの管理ユーティリティを参照してください。例えば、ICS を使用している場合は、System Manager の「コネクタ・プロパティ (Connector Properties)」ウィンドウの「更新メソッド」の列を参照してください。

標準コネクター・プロパティ

標準の構成プロパティでは、すべてのコネクターによって使用される情報が提供されます。これらのプロパティの詳細については、29 ページの『付録 A. コネクターの標準構成プロパティ』を参照してください。

重要: このコネクターはすべての統合ブローカーをサポートするため、すべてのブローカーの構成プロパティがこのコネクターに関係します。

注: このコネクターは単一スレッドなので、AgentConnections プロパティを利用することはできません。

コネクター固有のプロパティ

コネクター固有の構成プロパティでは、ランタイムにコネクターが必要とする情報を指定します。コネクター固有のプロパティでは、コネクター内の静的情報やロジックを、コード変更したり再ビルドしたりすることなく変更することもできます。

表 4 に、コネクターのコネクター固有構成プロパティをリストします。プロパティの説明については、以下の各セクションを参照してください。

表 4. コネクター固有の構成プロパティ

名前	可能な値	デフォルト値
ApplicationPassword		PS
ApplicationUserName		PS
AppServerMachineNameOrIP		
ConnectErrors		Unable to connect to destination:ORACLE not available:TNS:listener failed:Could not connect to application server:Database access is not allowed:Failed to establish MsgAPI service context for operator
disableCrossReferencing	True/False	相互参照を有効または無効にします。デフォルト値は False で、相互参照は使用可能となっています。
PollQuantity	1 から 500 (最大)	25
PortNumber		7000
PeopleToolsVersion		7.57
Priority	0 から n	0
UseDefaults		false

ApplicationPassword

コネクターのユーザー・アカウントのパスワード。

ApplicationUserName

コネクターのユーザー・アカウントの名前。ApplicationUserName は PeopleSoft の OperatorID です。

AppServerMachineNameOrIP

マシンの名前または IP アドレス。

ConnectErrors

PeopleSoft アプリケーションまたはデータベースへの接続が切断したときにコネクタの終了を引き起こすストリングのリスト。これらのストリングは、接続が切断したときにコネクタのログに書き込まれます。追加ストリングを指定することもできます。ストリングはコロン (:) を区切り文字として使って区切ります。

disableCrossReferencing

相互参照のオン/オフを切り替えます。デフォルト値は False で、相互参照は使用可能となっています。

PollQuantity

コネクタがポーリング間隔ごとに検索するデータベース表内のイベントの数。

PortNumber

Message Agent 要求のポート番号。

PeopleToolsVersion

PeopleTools のバージョン (例えば 7.52)。

Priority

優先順位。範囲は 0 から n で、0 が最上位となります。ポーリング間隔ごとに、コネクタは優先順位が最も高いイベントから下位のイベントへと、PollQuantity 構成プロパティで指定されたイベントの数に達するか、イベント表内にイベントがなくなるまで、イベント選出を行います。コネクタは現時点では優先順位を減分しません。

UseDefaults

Create 操作で、UseDefaults が true に設定されているか、あるいは何も設定されていない場合、コネクタは isRequired ビジネス・オブジェクト属性ごとに有効な値またはデフォルト値が設定されているかどうかを検査します。値が設定されていると、Create は成功します。設定されていない場合は、失敗します。パラメーターが false に設定されている場合は、コネクタは有効な値だけを検査します。有効な値が指定されていないと、Create 操作は失敗します。

コネクタの複数バージョンの作成

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

- コネクタ・インスタンス用に新規ディレクトリーを作成します。
- 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

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

```
ProductDir¥connectors¥connectorInstance
```

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

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

```
ProductDir¥repository¥connectorInstance
```

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** 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 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、『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. 始動スクリプトのショートカットを作成します (Windows のみ)。

4. 初期コネクターのショートカット・テキストをコピーし、新規コネクター・インスタンスの名前に一致するように (コマンド行で) 初期コネクターの名前を変更します。

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

カスタム・コネクター作成の詳細については、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

コネクターの開始

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。始動スクリプトは、次に示すようなコネクターのランタイム・ディレクトリに存在していなければなりません。

```
ProductDir¥connectors¥connName
```

ここで、*connName* はコネクターを示します。始動スクリプトの名前は、表 5 に示すように、オペレーティング・システム・プラットフォームによって異なります。

表 5. 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
UNIX ベースのシステム	connector_manager_connName
Windows	start_connName.bat

コネクター始動スクリプトは、以下に示すいずれかの方法で起動することができます。

- Windows システムで「スタート」メニューから。

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

- コマンド行から。

- Windows システム:

```
start_connName connName brokerName [-cconfigFile ]
```

- UNIX ベースのシステム:

```
connector_manager_connName -start
```

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

- WebSphere InterChange Server の場合は、*brokerName* に ICS インスタンスの名前を指定します。
- WebSphere Message Brokers (WebSphere MQ Integrator、WebSphere MQ Integrator Broker、または WebSphere Business Integration Message Broker) または WebSphere Application Server の場合は、*brokerName* にブローカーを示す文字列を指定します。

注: Windows システム上の WebSphere Message Broker または WebSphere Application Server の場合は、`-c` オプションに続いてコネクタ構成ファイルの名前を指定しなければなりません。ICS の場合は、`-c` はオプションです。

- Adapter Monitor から (WebSphere Business Integration Adapters 製品のみ)。Adapter Monitor は System Manager 始動時に起動されます。

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

- System Monitor から (WebSphere InterChange Server 製品のみ)。

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

- 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、コネクタが始動します。

コマンド行の始動オプションなどのコネクタの始動方法の詳細については、以下の資料のいずれかを参照してください。

- WebSphere InterChange Server については、「システム管理ガイド」を参照してください。
- WebSphere Message Brokers については、「*WebSphere Message Brokers 使用アダプター・インプリメンテーション・ガイド*」を参照してください。
- WebSphere Application Server については、「アダプター実装ガイド (*WebSphere Application Server*)」を参照してください。

コネクタの停止

コネクタを停止する方法は、以下に示すように、コネクタが始動された方法によって異なります。

- コマンド行からコネクタを始動した場合は、コネクタ始動スクリプトを用いて、以下の操作を実行します。
 - Windows システムでは、始動スクリプトを起動すると、そのコネクタ用の別個の「コンソール」ウィンドウが作成されます。このウィンドウで、「Q」と入力して Enter キーを押すと、コネクタが停止します。
 - UNIX ベースのシステムでは、コネクタはバックグラウンドで実行されるため、別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を停止します。

```
connector_manager_connName -stop
```

ここで、`connName` はコネクタの名前です。

- Adapter Monitor から (WebSphere Business Integration Adapters 製品のみ)。Adapter Monitor は System Manager 始動時に起動されます。

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

- System Monitor から (WebSphere InterChange Server 製品のみ)

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

- 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 システムのシャットダウン時に、コネクターは停止します。

アプリケーションへの接続の切断

コネクターは、アプリケーションとの接続が切断されたと判断すると、終了して、統合ブローカーに状態を返します。ConnectErrors プロパティで、コネクターの終了を引き起こすエラー・ストリングを指定します。詳細については、10 ページの『コネクター固有のプロパティ』を参照してください。

第 3 章 コネクター用のビジネス・オブジェクトの開発

この章では、WebSphere Business Integration Adapter for PeopleSoft 7.x がビジネス・オブジェクトをどのように処理するかを説明し、コネクター用のビジネス・オブジェクトを開発、変更する場合の提案事項を紹介します。この章には、次のセクションがあります。

『ビジネス・オブジェクト・アプリケーション固有のテキスト』

ビジネス・オブジェクト・アプリケーション固有のテキスト

ビジネス・オブジェクト内のアプリケーション固有のテキストを使用して、アプリケーション・データベースのどこにデータが置かれているかを識別します。このセクションでは、コネクターで有効なアプリケーション固有のテキスト形式を説明します。

オブジェクト・アプリケーション固有のテキスト形式

コネクターは、ビジネス・オブジェクトのレベルでアプリケーション固有のテキストを使用して、そのビジネス・オブジェクトにどの PeopleSoft アクティビティを使用するかを判断します。アプリケーション固有のテキストでこれを指定するには、name-value 形式 `Activity=name` を使用します。アクティビティの命名規則は CW ビジネス・オブジェクト名 です。例えば、Employee ビジネス・オブジェクトの場合、PeopleSoft アクティビティの名前は CWEmployee になります。この場合のアプリケーション固有のテキストは以下のようになります。

```
[BusinessObjectDefinition]
Name = Psft_Employee
Version = 1.0.0
AppSpecificInfo = Activity=CWEmployee
...
```

アプリケーション固有のテキストにおける Name パラメーターは、PeopleSoft におけるアクティビティの名前と一致しなければなりません。この name-value 形式が必要なのは、トップレベルのビジネス・オブジェクトの場合だけです。コネクターは、ビジネス・オブジェクトのレベルでアプリケーション固有のテキストを使用して、ビジネス・オブジェクトをコネクターで処理するかどうか判断します。ビジネス・オブジェクトの AppSpecificInfo フィールドにストリング NoOp が入っていると、そのビジネス・オブジェクトは処理されません。

NoOp ストリングを使用する場合は、アクティビティを name-value 形式で指定することも必要になります。この場合、name-value 形式で指定するアクティビティは、NoOp ストリングの直前にコロンで区切って挿入します。例えば、次のようになります。

```
[BusinessObjectDefinition]
Name = Psft_Department
Version = 1.0.0
AppSpecificInfo = Activity=CWDepartment:NoOp
...
```

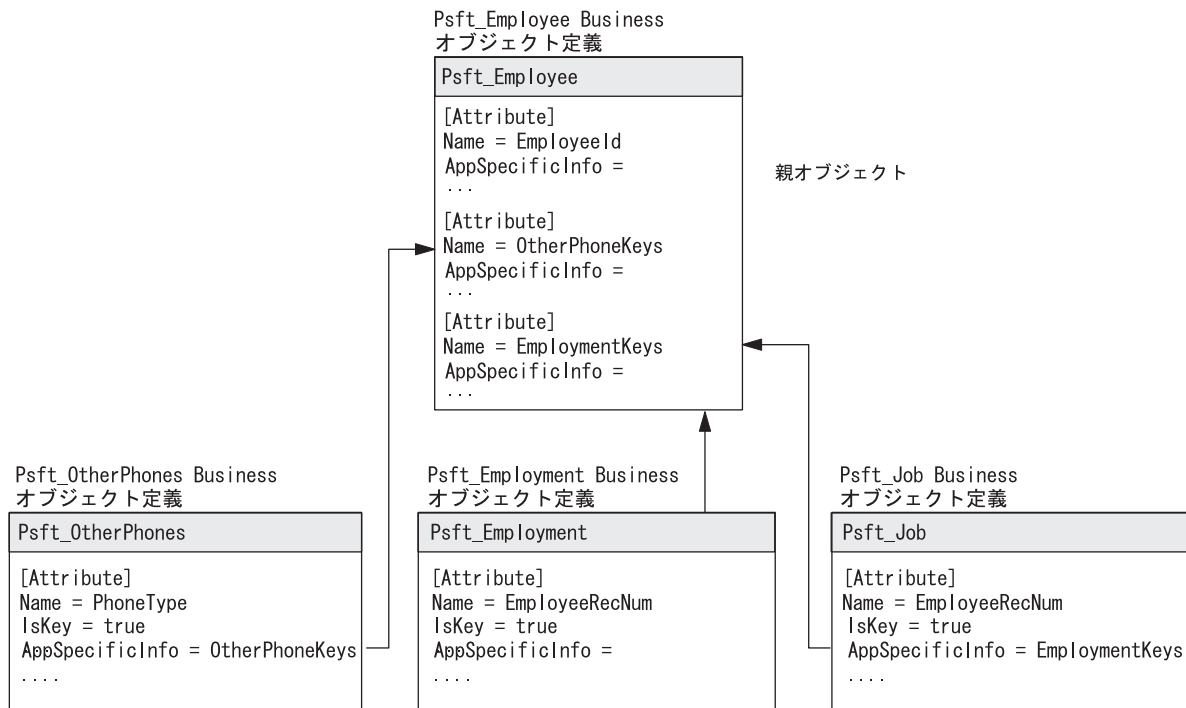
属性のアプリケーション固有のテキスト形式

コネクタは、属性のアプリケーション固有のテキストを使用して、階層ビジネス・オブジェクト内の子オブジェクトのキーを検索します。

コネクタのインストールの一環として、ビジネス・オブジェクト/動詞の組み合わせをサポートするメッセージ定義が PeopleSoft システムにインポートされます。これらのメッセージ定義には、トップレベルのビジネス・オブジェクトの属性に対応する 1 つ以上の専用フィールドが含まれていることがあります。専用フィールドはそれぞれ、子オブジェクトのキーを収集する PeopleCode に関連付けられます。

コネクタがイベント通知を受け取り、ビジネス・オブジェクトのデータの完全セットについて照会すると、PeopleCode が実行されて、専用フィールドで子キーが動的にアセンブルされます。コネクタは、子キーのリストである、各専用フィールドの値を取得して、それぞれの専用フィールドに対応する属性に入れます。コネクタは、リストを解析して、子オブジェクトの特定の属性に対する特定のキーを識別することができます。専用フィールドの内容は、PeopleSoft データベースには保管されません。

以下の図に、親ビジネス・オブジェクトと子ビジネス・オブジェクトの例、およびアプリケーション固有のテキストの形式を示します。Psft_Employee 親ビジネス・オブジェクトには、属性 OtherPhoneKeys があり、この属性は、このビジネス・オブジェクトのメッセージ定義で定義されている専用フィールドに対応します。子オブジェクトの場合は、アプリケーション固有のテキストに、キー属性に対する専用属性の名前が入っています。コネクタは、アプリケーション固有のテキスト内から、子オブジェクト属性のキーを識別します。



コネクタのビジネス・オブジェクトを変更する場合は、子オブジェクトのキー属性に、親オブジェクトの特殊な属性を指すアプリケーション固有のテキストが入っていないなければなりません。

動詞のアプリケーション固有のテキスト形式

コネクタは、現時点では動詞にアプリケーション固有のプロパティを使用していません。ビジネス・オブジェクトの定義を作成する場合、このフィールドはリンクのままにしてください。

コネクタ用のビジネス・オブジェクトの変更

注: コネクタは、子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を表す属性を、キー属性として指定することはサポートしていません。

キーのアセンブルに `PeopleCode` を使用している場合、コネクタのビジネス・オブジェクトを変更または拡張するときには、`XR_EVENT_WRK` 表内の `PeopleCode` と、製品と一緒にインストールされたメッセージ定義を変更してください。例えば、階層ビジネス・オブジェクトに子ビジネス・オブジェクトを追加する場合は、以下の通常ステップを実行する必要があります。

1. 子ビジネス・オブジェクトのコンテナ属性と、子キーを保持する属性の 2 つの新しい属性を追加するように、トップレベルのビジネス・オブジェクトを変更します。
2. 子ビジネス・オブジェクトを追加します。子ビジネス・オブジェクトのキーには、トップレベルのオブジェクトにおける属性を指すアプリケーション固有のテキストが入っていないなければなりません。例えば、`Psft_Employee` ビジネス・オブジェクトの場合は、子オブジェクト `Psft_OtherPhones` に、`PhoneType` 属性がトップレベルのビジネス・オブジェクト属性 `OtherPhoneKeys` を指す `OtherPhoneKeys` に設定された `AppSpecificInfo` フィールドがあります。
3. 子オブジェクトのすべてのキーを連結するセクションを追加するように、`XR_EVENT_WRK` 表内の `PeopleCode` を変更します。`Psft_Employee` ビジネス・オブジェクトの場合は、`Psft_OtherPhones` の `PeopleCode` により、ストリング `CELL:FAX:HOME` が生成されます (キーはコロンで区切られます)。
4. 子ビジネス・オブジェクトのメッセージ定義を追加し、トップレベルのビジネス・オブジェクトのメッセージ定義を変更します。例えば、`Psft_Employee` ビジネス・オブジェクトには、子オブジェクト用の以下のメッセージ定義がありません。
 - Retrieve OtherPhones
 - Create OtherPhones
 - Update OtherPhones
5. トップレベルのビジネス・オブジェクト用の `Retrieve Employee` メッセージ定義に、`OtherPhonesKeys` を追加します。

ビジネス・オブジェクトを変更する場合の規則

コネクタのビジネス・オブジェクトを変更する場合は、以下の規則に従ってください。

複数のパネル・グループにまたがるオブジェクト

トップレベルのビジネス・オブジェクトまたは子ビジネス・オブジェクトが 2 つの異なるパネル・グループにまたがる場合は、そのビジネス・オブジェクトまたは子オブジェクトを 2 つのビジネス・オブジェクトに分割します。

発効日および発効順序が指定されている行

1. PeopleSoft アプリケーション・オブジェクトの目的は、パネルがデータを処理する方法をモデル化することです。発効日の情報が含まれているパネルは、その発効日の情報が保管されているデータベース・レコードを識別します。この表/レコードは、EFFDT をキーの 1 つとして持っていなければなりません。この例では、Psft_EffDtEmployee オブジェクトとレコード PS_PERS_DATA_EFFDT を比較します。これは、従業員パネル・グループ内の発効日情報のあるサブパネルで参照されるメイン・レコードです。
2. EffectiveSequence は Job レコード・キーの一部で、これを利用して、同じ発効日のイベントを順に追跡することができます。「発効順序 (Effective Sequence)」は、同じ発効日に発生した複数のアクションを追跡する場合に使用する、手動入力フィールドです。自動データベース順序付け機能はありません。例えば、2000 年 4 月 25 日に、ある従業員に対し「Posn Chg」アクションを入力し、同じ日に他には何もアクションを実行しなかった場合、「発効順序 (Effective Sequence)」フィールドは 0 にしておきます。2000 年 4 月 25 日に、同じ従業員に対し後で「Pay Rt Chg」アクションも実行した場合は、「発効順序 (Effective Sequence)」フィールドを 1 に設定します。
3. 発効日と発効順序が指定されているすべての行に対し、メッセージ定義で発効日フィールドまたは発効順序を 2 回繰り返さなければなりません。これが必要なのは、メッセージ定義内の Create 操作 (例えば、CreateJob) だけです。
4. 発効日が指定されている行の場合は、ビジネス・オブジェクトの属性としてパネル上に表示するすべてのフィールドをリストします。データベースに新しい行を挿入すると、ユーザーが F7 キーを押してオンラインで行を挿入したものとして、コネクタは同じ動作を行います。どちらの場合も、スクロール内のすべての属性が新しいスクロールにコピーされます。ビジネス・オブジェクトの属性をすべて指定しないと、直前の行から新しい行に間違っデータのコピーしてしまう危険があります。
5. 発効日が指定されている行の場合は、属性のアプリケーション固有のテキストを、ビジネス・オブジェクトで EFFDT になるように指定する必要があります。
6. 発効日が指定されている行の場合は、追加メッセージ定義が必要です。メッセージ定義の名前は、ビジネス・オブジェクトの名前 + EFFDT でなければなりません。属性はキー (input) と EFFDT (output) です。例えば、次のようになります。

```
Message Definition = RetrieveJobEFFDT
Attributes = EmployeeId (input)
             EmployeeRecNum (input)
             JobEffectiveOnDate (output)
```

メッセージ定義

1. Message Agent はオンライン・パネルと同じ順序でデータを処理するので、メッセージ定義におけるフィールドの順序が重要なのは、以前に入力したデータから

フィールドを検証しなければならない場合だけです。それ以外は、メッセージ定義におけるフィールドのリスト順序を気にする必要はありません。

2. メッセージ定義で、メッセージ Output All Occurrences が機能するのは、レベル 1 のフィールドの場合だけです。詳細については、「Peoplebooks Development Tools」ガイドの「メッセージ定義の作成 (Create Message Definitions)」を参照してください。レベル 2 またはレベル 3 の行を取得する必要がある場合は、新しいパネルを作成し、レベル 2 またはレベル 3 のフィールドをレベル 1 に移動しなければなりません。

ビジネス・オブジェクト属性

属性は、ビジネス・オブジェクトとメッセージ定義の両方に存在しなければなりません。属性の名前のスペルは一致していなければなりません。ビジネス・オブジェクトでは、属性の順序は重要ではありません。

子ビジネス・オブジェクトのキー

子ビジネス・オブジェクトでは、親ビジネス・オブジェクトにすでに存在するキーを繰り返す必要はありません。コネクタは、階層ビジネス・オブジェクトを処理する場合、まず親ビジネス・オブジェクトに保管されているキーを取得してから、子ビジネス・オブジェクトを処理します。ただし、Message Agent 定義ではキーを繰り返す必要があります。

属性の名前

属性の名前として EFFDT または EFFSEQ を使用してはなりません。

AppSpecificInfo

AppSpecificInfo に割り当てることができる ID は 2 つだけです。ID はコロン (:) で区切ります。例えば、次のようになります。

```
AppSpecificInfo = EFFDT:Additional PayKeys
```

イベント通知

イベント通知を処理する場合、コネクタは、イベント表内の OBJECT_ID の値を取得して、[ビジネス・オブジェクト名]Id というフィールドにその値を保管します。このフィールド名は、PeopleSoft におけるキーの実際の名前と異なることがあります。例えば、PeopleSoft 7.5 では、Item 表のキーは InvItemId ですが、コネクタの場合には、Item ビジネス・オブジェクトのこのキーの名前は ItemId でなければなりません。

ビジネス・オブジェクトとメッセージ定義における属性の名前

コネクタでは、ビジネス・オブジェクトと、そのオブジェクトの各動詞のメッセージ定義とで、属性の名前が正確に一致しなければなりません。例えば、Psft_Item における属性の名前と、以下のメッセージ定義における属性の名前は正確に一致しなければなりません。

- Create Item
- Retrieve Item
- Update Item

メッセージ定義を更新する場合は、属性の名前のスペルと大文字の使用がビジネス・オブジェクトの属性と一致することを確認してください。これらのメッセージ定義は、IBM WebSphere Business Integration ビジネス・プロセスと、コネクタールンテグレーション・アクティビティーにあります。

階層ビジネス・オブジェクトのトランザクション処理

コネクターが、Create 動詞または Update 動詞を使って階層ビジネス・オブジェクトを処理しているときに障害が発生すると、1 つ以上の子ビジネス・オブジェクトが処理されないことがあります。

コネクターでは、トランザクションは、階層ビジネス・オブジェクト全体に対する操作として定義されていません。代わりに、コネクターでは、階層ビジネス・オブジェクト内の個々の親または子ビジネス・オブジェクトに対する操作ごとに、別々のトランザクションが定義されます。コネクターは、Create 動詞が指定されたビジネス・オブジェクトを受け取った場合には、まず親ビジネス・オブジェクトを処理してから、個々の子ビジネス・オブジェクトを処理します。通常は、階層内の各ビジネス・オブジェクトごとにトランザクションを開始し、アプリケーション内にエンティティーを作成してから、トランザクションをコミットします。

コネクターが 1 つ以上の子エンティティーの作成に成功した後で、別の子エンティティーを作成しているときに障害が発生すると、コネクターは、そのエンティティーをバックアウトして、この特定のビジネス・オブジェクトに対する Create 操作をロールバックします。コネクターは、残りのすべてのエンティティーの作成が終了するまで、子ビジネス・オブジェクトの処理を続けます。障害が発生したので、コネクターは、ビジネス・オブジェクトに対する操作全体について、統合ブローカーに BON_FAIL を戻します。また、アプリケーションでの Create 操作または Update 操作が完了しない場合もあります。Update トランザクションの場合も、同じことが当てはまります。

コネクターが BON_FAIL を戻すと、階層ビジネス・オブジェクト全体が、未処理イベント・キューに入ります。システム管理者は、宛先のアプリケーションを検査し、障害の原因となった問題を解決しなければなりません。その後で、コネクターにビジネス・オブジェクトを再実行依頼する必要があります。コネクターは、アプリケーション・エンティティーを作成する前にその存在を確認するので、ビジネス・オブジェクトが再実行依頼されても、すでに作成されている子ビジネス・オブジェクトのデータが重複することはありません。

ビジネス・オブジェクト定義ファイルのサンプル

この製品には、3 つのサンプル・ビジネス・オブジェクト定義ファイルが付属しています。

- BO_Psft_DEPT
- BO_PsftEmployee
- SavePostChange

BO_Psft_DEPT ビジネス・オブジェクト

次に示す例は BO_Psft DEPT ビジネス・オブジェクトです。

```

[BusinessObjectDefinition] Name = DeptTbl Version = 1.0.0 AppSpecificInfo =
CiName=DEPT [Attribute] Name = Company Type = String
MaxLength = 255 IsKey = true IsForeignKey = false IsRequired =
false AppSpecificInfo = get=getCompany:set=setCompany
IsRequiredServerBound = false [End] [Attribute] Name =
BudgetLvl Type = String MaxLength = 1 IsKey = false
IsForeignKey = false IsRequired = false AppSpecificInfo =
get=getBudgetLvl:set=setBudgetLvl IsRequiredServerBound = false
[End] [Attribute] Name = Descr Type = String MaxLength = 1
IsKey = false IsForeignKey = false IsRequired = false
AppSpecificInfo = get=getDescr:set=setDescr IsRequiredServerBound =
false [End] [Attribute] Name = DescrShort Type = String
MaxLength = 1 IsKey = false IsForeignKey = false IsRequired =
false AppSpecificInfo = get=getDescrshort:set=setDescrshort
IsRequiredServerBound = false [End] [Attribute] Name =
ObjectEventId Type = String MaxLength = 255 IsKey = false
IsForeignKey = false IsRequired = false IsRequiredServerBound =
false [End] [Verb] Name = Create [End] [Verb] Name
= Delete [End] [Verb] Name = Retrieve [End] [Verb]
Name = Update [End] [End] [BusinessObjectDefinition] Name = Psft_dept
Version = 1.0.0 AppSpecificInfo = CiName=DEPT [Attribute] Name =
Deptid Type = String MaxLength = 255 IsKey = true
IsForeignKey = false IsRequired = true AppSpecificInfo =
get=getDeptid:set=setDeptid:GetKey=true IsRequiredServerBound = false
[End] [Attribute] Name = Setid Type = String MaxLength
= 1 IsKey = true IsForeignKey = false IsRequired = true
AppSpecificInfo = get=getSetid:set=setSetid:GetKey=true
IsRequiredServerBound = false [End] [Attribute] Name = DptTbl
Type = DeptTbl ContainedObjectVersion = 1.0.0 Relationship =
Containment Cardinality = n MaxLength = 1 IsKey = false
IsForeignKey = false IsRequired = true AppSpecificInfo =
get=getDeptTbl:KEEPRELATIONSHIP=true IsRequiredServerBound = false
[End] [Attribute] Name = ObjectEventId Type = String
MaxLength = 255 IsKey = false IsForeignKey = false IsRequired =
false IsRequiredServerBound = false [End] [Verb] Name =
Create [End] [Verb] Name = Delete [End] [Verb] Name
= Retrieve [End] [Verb] Name = Update [End] [End]

```

BO_PsftEmployee ビジネス・オブジェクト

次に示す例は BO_PsftEmployee ビジネス・オブジェクトです。

```

[BusinessObjectDefinition] Name = PSFTEmployee Version = 1.0.0
AppSpecificInfo = cIName=Emp [Attribute] Name = EMPID Type =
String Cardinality = 1 MaxLength = 255 IsKey = true
IsForeignKey = false IsRequired = true AppSpecificInfo =
get=getEmpId:set=setEmpId:keepRelationship=false:uid=true:
findKey=true:getKey=true:createKey=true IsRequiredServerBound = false

```

```

    [End]      [Attribute]      Name = EMPL_RCD      Type = String
Cardinality = 1      MaxLength = 1      IsKey = true      IsForeignKey = false
    IsRequired = true      AppSpecificInfo =
get=getEmplRcd:set=setEmplRcd:keepRelationship=false:uid=true:
findKey=true:getKey=true:createKey=true      IsRequiredServerBound = false
    [End]      [Attribute]      Name = NAME      Type = String      Cardinality
= 1      MaxLength = 1      IsKey = true      IsForeignKey = false
IsRequired = true      AppSpecificInfo =
get=getName:set=setName:keepRelationship=false:uid=false:
findKey=true:getKey=false:createKey=false      IsRequiredServerBound = false
    [End]      [Attribute]      Name = LAST_NAME_SRCH      Type = String
Cardinality = 1 MaxLength = 1      IsKey = true      IsForeignKey = false
IsRequired = true      AppSpecificInfo =
get=getLastNameSrch:set=setLastNameSrch:keepRelationship=false:uid=false:
    findKey=true:getKey=false:createKey=false      IsRequiredServerBound =
false      [End]      [Attribute]      Name = NAME_AC      Type = String
Cardinality = 1      MaxLength = 1      IsKey = true      IsForeignKey = false
    IsRequired = true      AppSpecificInfo =
get=getNameAc:set=setNameAc:keepRelationship=false:uid=false:
findKey=true:getKey=false:createKey=false      IsRequiredServerBound = false
    [End]      [Attribute]      Name = PER_STATUS      Type = String
Cardinality = 1      MaxLength = 1      IsKey = true      IsForeignKey = false
    IsRequired = true      AppSpecificInfo =
get=getPerStatus:set=setPerStatus:keepRelationship=false:uid=false:
findKey=true:getKey=false:createKey=false      IsRequiredServerBound = false
    [End]      [Attribute]      Name = EMPLID_0      Type = String
Cardinality = 1      MaxLength = 1      IsKey = false      IsForeignKey =
false      IsRequired = false      AppSpecificInfo =
get=getEmplid0:set=setEmplid0:keepRelationship=false:uid=false:
findKey=false:getKey=false:createKey=false      IsRequiredServerBound =
false      [End]      [Attribute]      Name = ORIG_HIRE_DT      Type = String
    Cardinality = 1      MaxLength = 1      IsKey = false      IsForeignKey =
false      IsRequired = false      AppSpecificInfo =
get=getOrigHireDt:set=setOrigHireDt:keepRelationship=false:uid=false:
findKey=false:getKey=false:createKey=false      IsRequiredServerBound =
false      [End]      [Attribute]      Name = SEX      Type = String
Cardinality = 1      MaxLength = 1      IsKey = false      IsForeignKey =
false      IsRequired = true      AppSpecificInfo =
get=getSex:set=setSex:keepRelationship=false:uid=false:findKey=false:
getKey=false:createKey=false      IsRequiredServerBound = false      [End]
    [Attribute]      Name = BIRTHDATE      Type = String      Cardinality = 1
    MaxLength = 1      IsKey = false      IsForeignKey = false      IsRequired
= false      AppSpecificInfo =
get=getBirthdate:set=setBirthdate:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
FT_STUDENT      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = true

```



```

AppSpecificInfo = get=getFtStudent:set=setFtStudent:keepRelationship=false:
    uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false    [End]    [Attribute]    Name =
BENEFIT_RCD_NBR    Type = String    Cardinality = 1    MaxLength = 1
IsKey = false    IsForeignKey = false    IsRequired = false
AppSpecificInfo = get=getBenefitRcdNbr:set=setBenefitRcdNbr:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false    [End]    [Attribute]    Name =
HOME_HOST_CLASS    Type = String    Cardinality = 1    MaxLength = 1
IsKey = false    IsForeignKey = false    IsRequired = true
AppSpecificInfo = get=getHomeHostClass:set=setHomeHostClass:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false    [End]    [Attribute]    Name =
HIRE_DT    Type = String    Cardinality = 1    MaxLength = 1    IsKey =
false    IsForeignKey = false    IsRequired = false    AppSpecificInfo =
get=getHireDt:set=setHireDt:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false    [End]    [Attribute]    Name =
CMPNY_SENIORITY_DT    Type = String    Cardinality = 1    MaxLength = 1
    IsKey = false    IsForeignKey = false    IsRequired = false
AppSpecificInfo = get=getCmpnySeniorityDt:
set=setCmpnySeniorityDt:keepRelationship=false:uid=false:
findKey=false:getKey=false:createKey=false    IsRequiredServerBound =
false    [End]    [Attribute]    Name = SERVICE_DT    Type = String
Cardinality = 1    MaxLength = 1    IsKey = false    IsForeignKey =
false    IsRequired = false    AppSpecificInfo =
get=getServiceDt:set=setServiceDt:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false    [End]    [Attribute]    Name =
PROF_EXPERIENCE_DT    Type = String    Cardinality = 1    MaxLength = 1
    IsKey = false    IsForeignKey = false    IsRequired = false
AppSpecificInfo = get=getProfExperienceDt:
set=setProfExperienceDt:keepRelationship=false:uid=false:
findKey=false:getKey=false:createKey=false    IsRequiredServerBound =
false    [End]    [Attribute]    Name = LAST_VERIFICATN_DT    Type =
String    Cardinality = 1    MaxLength = 1    IsKey = false
IsForeignKey = false    IsRequired = false    AppSpecificInfo =
get=getLastVerificatnDt:set=setLastVerificatnDt:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false    [End]    [Attribute]    Name =
EXPECTED_RETURN_DT    Type = String    Cardinality = 1    MaxLength = 1
    IsKey = false    IsForeignKey = false    IsRequired = false
AppSpecificInfo = get=getExpectedReturnDt:set=
setExpectedReturnDt:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false    [End]    [Attribute]    Name =
LAST_DATE_WORKED    Type = String    Cardinality = 1    MaxLength = 1
    IsKey = false    IsForeignKey = false    IsRequired = false

```

```

AppSpecificInfo = get=getLastDateWorked:
set=setLastDateWorked:keepRelationship=false:uid=false:
findKey=false:getKey=false:createKey=false      IsRequiredServerBound =
false      [End]      [Attribute]      Name = LAST_INCREASE_DT      Type =
String      Cardinality = 1      MaxLength = 1      IsKey = false
IsForeignKey = false      IsRequired = false      AppSpecificInfo =
get=getLastIncreaseDt:      set=setLastIncreaseDt:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
OWN_5PERCENT_CO      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = true
AppSpecificInfo = get=getOwn5percentCo:
set=setOwn5percentCo:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
BUSINESS_TITLE      Type = String      Cardinality = 1      MaxLength = 1
IsKey = true      IsForeignKey = false      IsRequired = true
AppSpecificInfo = get=getBusinessTitle:
set=setBusinessTitle:keepRelationship=false:
uid=false:findKey=false:getKey=true:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
REPORTS_TO      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getReportsTo:set=setReportsTo:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
SUPERVISOR_ID      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getSupervisorId:set=setSupervisorId:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
PROBATION_DT      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getProbationDt:set=setProbationDt:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
SECURITY_CLEARANCE      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getSecurityClearance:
set=setSecurityClearance:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name = PHONE
Type = String      Cardinality = 1      MaxLength = 1      IsKey = false
IsForeignKey = false      IsRequired = false      AppSpecificInfo =
get=getPhone:set=setPhone:      keepRelationship=false:uid=false:
findKey=false:getKey=false:createKey=false      IsRequiredServerBound =
false      [End]      [Attribute]      Name = TIME_RPT_LOCK      Type = String
Cardinality = 1      MaxLength = 1      IsKey = false      IsForeignKey =

```

```

false      IsRequired = false      AppSpecificInfo =
get=getTimeRptLock:set=setTimeRptLock:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
JOB_REPORTING      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = true
AppSpecificInfo = get=getJobReporting:set=setJobReporting:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
DED_TAKEN      Type = String      Cardinality = 1      MaxLength = 1      IsKey
= false      IsForeignKey = false      IsRequired = true      AppSpecificInfo
= get=getDedTaken:set=setDedTaken:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
DED_SUBSET_ID      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getDedSubsetId:set=setDedSubsetId:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
CAN_ABORIGINAL      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getCanAboriginal:set=setCanAboriginal:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
CAN_VISBL_MINORITY      Type = String      Cardinality = 1      MaxLength = 1
    IsKey = false      IsForeignKey = false      IsRequired = true
AppSpecificInfo = get=getCanVisblMinority:set=setCanVisblMinority:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
CURRENT_SEQ      Type = String      Cardinality = 1      MaxLength = 1
IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getCurrentSeq:set=setCurrentSeq:
keepRelationship=false:uid=false:findKey=false:getKey=false:createKey=false
    IsRequiredServerBound = false      [End]      [Attribute]      Name =
PERS_DATA_EFFDT      Type = PERS_DATA_EFFDT      ContainedObjectVersion =
1.0.0      Relationship = Containment      Cardinality = n      MaxLength = 1
    IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getPersDataEffdt:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
EMAIL_ADDRESSES      Type = EMAIL_ADDRESSES      ContainedObjectVersion =
1.0.0      Relationship = Containment      Cardinality = n      MaxLength = 1
    IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getEmailAddresses:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name =
PERSONAL_PHONE      Type = PERSONAL_PHONE      ContainedObjectVersion = 1.0.0
    Relationship = Containment      Cardinality = n      MaxLength = 1

```

```

IsKey = false      IsForeignKey = false      IsRequired = false
AppSpecificInfo = get=getPersonalPhone:keepRelationship=false:
uid=false:findKey=false:getKey=false:createKey=false
IsRequiredServerBound = false      [End]      [Attribute]      Name = PERS_NID
      Type = PERS_NID      ContainedObjectVersion = 1.0.0      Relationship =
Containment      Cardinality = n      MaxLength = 1      IsKey = false
IsForeignKey = false      IsRequired = false      IsRequiredServerBound =
false      [End]      [Attribute]      Name = JOB      Type = JOB
ContainedObjectVersion = 1.0.0      Relationship = Containment
Cardinality = n      MaxLength = 1      IsKey = false      IsForeignKey =
false      IsRequired = false      IsRequiredServerBound = false      [End]
      [Attribute]      Name = ObjectEventId      Type = String      MaxLength =
255      IsKey = false      IsForeignKey = false      IsRequired = false
IsRequiredServerBound = false      [End]      [Verb]      Name = Create
[End]      [Verb]      Name = Delete      [End]      [Verb]      Name = Retrieve
      [End]      [Verb]      Name = Update      [End] [End]

```

付録 A. コネクターの標準構成プロパティ

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

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (WebSphere Message Brokers (WMQI) と総称)
- WebSphere Application Server (WAS)

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

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

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

新規プロパティと削除されたプロパティ

以下の標準プロパティは、本リリースで追加されました。

新規プロパティ

- XMLNamespaceFormat

削除されたプロパティ

- RestartCount

標準コネクター・プロパティの構成

アダプター・コネクターには 2 つのタイプの構成プロパティがあります。

- 標準構成プロパティ
- コネクター固有の構成プロパティ

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

Connector Configurator の使用

Connector Configurator からコネクタ・プロパティを構成します。Connector Configurator には、System Manager からアクセスします。Connector Configurator の使用法の詳細については、本書の Connector Configurator に関するセクションを参照してください。

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

プロパティ値の設定と更新

プロパティ・フィールドのデフォルトの長さは 255 文字です。

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

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

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

• 動的

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

• エージェント再始動 (ICS のみ)

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

• コンポーネント再始動

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

• サーバー再始動

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

特定のプロパティの更新方法を確認するには、「Connector Configurator」ウィンドウ内の「更新メソッド」列を参照するか、次に示す 31 ページの表 6 の「更新メソッド」列を参照してください。

標準プロパティの要約

表 6 は、標準コネクタ構成プロパティの早見表です。標準プロパティの依存関係は `RepositoryDirectory` に基づいているため、コネクタによっては使用されないプロパティがあり、使用する統合ブローカーによってプロパティの設定が異なる可能性があります。

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

注: 表 6 の「注」列にある「`Repository Directory` は `REMOTE`」という句は、ブローカーが `InterChange Server` であることを示します。ブローカーが `WMQI` または `WAS` の場合には、リポジトリ・ディレクトリは `LOCAL` に設定されます。

表 6. 標準構成プロパティの要約

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

表 6. 標準構成プロパティの要約 (続き)

プロパティ名	可能な値	デフォルト値	更新メソッド	注
ControllerStoreAndForwardMode	true または false	true	動的	Repository Directory は <REMOTE> (ブローカーは ICS)
ControllerTraceLevel	0 から 5	0	動的	Repository Directory は <REMOTE> (ブローカーは ICS)
DeliveryQueue		CONNECTORNAME/DELIVERYQUEUE	コンポーネント再始動	JMS トランスポートのみ
DeliveryTransport	MQ、IDL、または JMS	JMS	コンポーネント再始動	Repository Directory がローカルの場合は、値は JMS のみ
DuplicateEventElimination	true または false	false	コンポーネント再始動	JMS トランスポートのみ: Container Managed Events は <NONE> でなければならぬ
FaultQueue		CONNECTORNAME/FAULTQUEUE	コンポーネント再始動	JMS トランスポートのみ
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory または CxCommon.Messaging.jms.SonicMQFactory または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	JMS トランスポートのみ
jms.MessageBrokerName	FactoryClassName が IBM の場合は crossworlds.queue.manager を使用。FactoryClassName が Sonic の場合は localhost:2506 を使用。	crossworlds.queue.manager	コンポーネント再始動	JMS トランスポートのみ
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	JMS トランスポートのみ
jms.Password	任意の有効なパスワード		コンポーネント再始動	JMS トランスポートのみ
jms.UserName	任意の有効な名前		コンポーネント再始動	JMS トランスポートのみ
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	Repository Directory は <REMOTE> (ブローカーは ICS)

表 6. 標準構成プロパティの要約 (続き)

プロパティ名	可能な値	デフォルト値	更新メソッド	注
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	Repository Directory は <REMOTE> (ブローカーは ICS)
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	Repository Directory は <REMOTE> (ブローカーは ICS)
ListenerConcurrency	1 から 100	1	コンポーネント再始動	Delivery Transport は MQ でなければならぬ
Locale	en_US、 ja_JP、 ko_KR、 zh_CN、 zh_TW、 fr_FR、 de_DE、 it_IT、 es_ES、 pt_BR 注: これは、サポートされるロケールの一部です。	en_US	コンポーネント再始動	
LogAtInterchangeEnd	true または false	false	コンポーネント再始動	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
MessageFileName	パスまたはファイル名	CONNECTORNAMEConnector.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	CONNECTORNAME/MONITORQUEUE	コンポーネント再始動	JMS トランスポートのみ: DuplicateEvent Elimination は true でなければならぬ
OADAutoRestartAgent	true または false	false	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)

表 6. 標準構成プロパティの要約 (続き)

プロパティ名	可能な値	デフォルト値	更新メソッド	注
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE> でなければならぬ (ブローカーは ICS)
PollEndTime	HH:MM	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒) no (ポーリングを使用不可にする) key (コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力された場合にのみポーリングする)	10000	動的	
PollQuantity	1 から 500	1	エージェント再始動	JMS トランスポートのみ: Container Managed Events を指定
PollStartTime	HH:MM (HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
RepositoryDirectory	メタデータ・リポジトリの場所		エージェント再始動	ICS の場合は <REMOTE> に設定する。WebSphere MQ Message Brokers および WAS の場合: C:¥crossworlds¥repository に設定する
RequestQueue	有効な JMS キュー名	CONNECTORNAME/REQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
ResponseQueue	有効な JMS キュー名	CONNECTORNAME/RESPONSEQUEUE	コンポーネント再始動	Delivery Transport が JMS の場合: Repository Directory が <REMOTE> の場合のみ必要
RestartRetryCount	0 から 99	3	動的	
RestartRetryInterval	適切な正数 (単位: 分): 1 から 2147483547	1	動的	
RHF2MessageDomain	mrm、xml	mrm	コンポーネント再始動	Delivery Transport が JMS であり、かつ WireFormat が CwXML である

表 6. 標準構成プロパティの要約 (続き)

プロパティ名	可能な値	デフォルト値	更新メソッド	注
SourceQueue	有効な WebSphere MQ 名	CONNECTORNAME/SOURCEQUEUE	エージェント再始動	Delivery Transport が JMS であり、かつ Container Managed Events が指定されている場合のみ
SynchronousRequestQueue		CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
SynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	Delivery Transport は JMS
SynchronousResponseQueue		CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
WireFormat	CwXML、CwBO	CwXML	エージェント再始動	Repository Directory が <REMOTE> でない場合は CwXML。Repository Directory が <REMOTE> であれば CwBO
WsifSynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	WAS のみ
XMLNameSpaceFormat	short、long	short	エージェント再始動	WebSphere MQ Message Brokers および WAS のみ

標準構成プロパティ

このセクションでは、各標準コネクタ構成プロパティの定義を示します。

AdminInQueue

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

デフォルト値は CONNECTORNAME/ADMININQUEUE です。

AdminOutQueue

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

デフォルト値は CONNECTORNAME/ADMINOUTQUEUE です。

AgentConnections

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

AgentConnections プロパティは、orb.init[] により開かれる ORB (オブジェクト・リクエスト・ブローカー) 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 に設定されます。必要に応じてこの値を変更できます。

AgentTraceLevel

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

ApplicationName

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

BrokerType

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

CharacterEncoding

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

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

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

ConcurrentEventTriggeredFlows

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

コネクタがイベントのデリバリー時に並行処理できるビジネス・オブジェクトの数を決定します。この属性の値を、並行してマップおよびデリバリーできるビジネス・オブジェクトの数に設定します。例えば、この属性の値を 5 に設定すると、5 個のビジネス・オブジェクトが並行して処理されます。デフォルト値は 1 です。

このプロパティを 1 よりも大きい値に設定すると、ソース・アプリケーションのコネクタが、複数のイベント・ビジネス・オブジェクトを同時にマップして、複

数のコラボレーション・インスタンスにそれらのビジネス・オブジェクトを同時にデリバリーすることができます。これにより、統合ブローカーへのビジネス・オブジェクトのデリバリーにかかる時間、特にビジネス・オブジェクトが複雑なマップを使用している場合のデリバリー時間が短縮されます。ビジネス・オブジェクトのコラボレーションに到達する速度を増大させると、システム全体のパフォーマンスを向上させることができます。

ソース・アプリケーションから宛先アプリケーションまでのフロー全体に並行処理を実装するには、次のようにする必要があります。

- `Maximum number of concurrent events` プロパティの値を増加して、コラボレーションが複数のスレッドを使用できるように構成します。
- 宛先アプリケーションのアプリケーション固有コンポーネントが複数の要求を並行して実行できることを確認します。つまり、このコンポーネントがマルチスレッド化されているか、またはコネクタ・エージェント並列処理を使用でき、複数プロセスに対応するよう構成されている必要があります。`Parallel Process Degree` 構成プロパティに、1 より大きい値を設定します。

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

ContainerManagedEvents

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

デフォルト値はありません。

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

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

また、`MimeType`、`DHClass` (データ・ハンドラー・クラス)、および `DataHandlerConfigMOName` (オプションのメタオブジェクト名) プロパティを設定したデータ・ハンドラーも構成する必要があります。これらのプロパティの値を設定するには、`Connector Configurator` の「データ・ハンドラー」タブを使用します。

これらのプロパティはアダプター固有ですが、例の値は次のようになります。

- `MimeType` = `text/xml`
- `DHClass` = `com.crossworlds.DataHandlers.text.xml`
- `DataHandlerConfigMOName` = `MO_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` を JMS に設定した場合にのみ表示されます。

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

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

ControllerStoreAndForwardMode

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

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

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

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

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

デフォルト値は true です。

ControllerTraceLevel

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

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

DeliveryQueue

DeliveryTransport が JMS の場合のみ適用されます。

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

デフォルト値は CONNECTORNAME/DELIVERYQUEUE です。

DeliveryTransport

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

- `RepositoryDirectory` がリモートの場合は、`DeliveryTransport` プロパティの指定可能な値は `MQ`、`IDL`、または `JMS` であり、デフォルトは `IDL` になります。
- `RepositoryDirectory` がローカル・ディレクトリーの場合は、指定可能な値は `JMS` のみです。

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

WebSphere MQ および IDL

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

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

- 非同期 (ASYNC) 通信:
`WebSphere MQ` を使用すると、アプリケーション固有のコンポーネントは、サーバーが利用不能である場合でも、イベントをポーリングして永続的に格納することができます。
- サーバー・サイド・パフォーマンス:
`WebSphere MQ` を使用すると、サーバー・サイドのパフォーマンスが向上します。最適化モードでは、`WebSphere MQ` はイベントへのポインターのみをリポジトリ・データベースに格納するので、実際のイベントは `WebSphere MQ` キュー内に残ります。これにより、サイズが大きい可能性のあるイベントをリポジトリ・データベースに書き込む必要がありません。
- エージェント・サイド・パフォーマンス:
`WebSphere MQ` を使用すると、アプリケーション固有のコンポーネント側のパフォーマンスが向上します。`WebSphere MQ` を使用すると、コネクタのポーリング・スレッドは、イベントを選出した後、コネクタのキューにそのイベントを入れ、次のイベントを選出します。この方法は `IDL` よりも高速で、`IDL` の場合、コネクタのポーリング・スレッドは、イベントを選出した後、ネットワーク経由でサーバー・プロセスにアクセスしてそのイベントをリポジトリ・データベースに永続的に格納してから、次のイベントを選出する必要があります。

JMS

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

`JMS` をデリバリー・トランスポートとして選択した場合は、

`jms.MessageBrokerName`、`jms.FactoryClassName`、`jms.Password`、`jms.UserName` などの追加の `JMS` プロパティが `Connector Configurator` 内に表示されます。このうち最初の 2 つは、このトランスポートの必須プロパティです。

重要: 以下の環境では、コネクタに `JMS` トランスポート機構を使用すると、メモリ制限が発生することもあります。

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

この環境では、WebSphere MQ クライアント内でメモリーが使用されるため、(サーバー側の) コネクター・コントローラーと (クライアント側の) コネクターの両方を始動するのは困難な場合があります。ご使用のシステムのプロセス・ヒープ・サイズが 768M 未満である場合には、次のように設定することをお勧めします。

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

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

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

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

- IPCCBaseAddress プロパティーの値を 11 または 12 に設定する。このプロパティーの詳細については、「システム・インストール・ガイド (UNIX 版)」を参照してください。

DuplicateEventElimination

このプロパティーを true に設定すると、JMS 対応コネクターによるデリバリー・キューへの重複イベントのデリバリーが防止されます。この機能を使用するには、コネクターに対し、アプリケーション固有のコード内でビジネス・オブジェクトの **ObjectEventId** 属性として一意のイベント ID が設定されている必要があります。これはコネクター開発時に設定されます。

このプロパティーは、false に設定することもできます。

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

FaultQueue

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

デフォルト値は CONNECTORNAME/FAULTQUEUE です。

JvmMaxHeapSize

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

デフォルト値は 128M です。

JvmMaxNativeStackSize

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

デフォルト値は 128K です。

JvmMinHeapSize

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

デフォルト値は 1M です。

jms.FactoryClassName

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

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

jms.MessageBrokerName

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

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

リモート・メッセージ・ブローカーに接続すると、このプロパティは次の (必須) 値をとります。

QueueMgrName:<Channel>:<HostName>:<PortNumber>

各変数の意味は以下のとおりです。

QueueMgrName: キュー・マネージャー名です。

Channel: クライアントが使用するチャンネルです。

HostName: キュー・マネージャーの配置先のマシン名です。

PortNumber: 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、次のようになります。

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

jms.NumConcurrentRequests

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

デフォルト値は 10 です。

jms.Password

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

デフォルトはありません。

jms.UserName

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

デフォルトはありません。

ListenerConcurrency

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

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

Locale

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

ロケール名は、次の書式で指定します。

ll_TT.codeset

ここで、以下のように説明されます。

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

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

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

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

LogAtInterchangeEnd

RepositoryDirectory が <REMOTE> の場合のみ適用可能です。

統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。ブローカーのログ宛先にログを記録すると、電子メール通知もオンになります。これにより、

エラーまたは致命的エラーが発生すると、InterchangeSystem.cfg ファイルに指定された MESSAGE_RECIPIENT に対する電子メール・メッセージが生成されます。

例えば、LogAtInterChangeEnd を true に設定した場合にコネクタからアプリケーションへの接続が失われると、指定されたメッセージ宛先に、電子メール・メッセージが送信されます。デフォルト値は false です。

MaxEventCapacity

コントローラー・バッファ内のイベントの最大数。このプロパティはフロー制御が使用し、RepositoryDirectory プロパティの値が <REMOTE> の場合にのみ適用されます。

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

MessageFileName

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

MonitorQueue

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

デフォルト値は CONNECTORNAME/MONITORQUEUE です。

OADAutoRestartAgent

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

コネクタが自動再始動およびリモート再始動機能を使用するかどうかを指定します。この機能では、MQ により起動される Object Activation Daemon (OAD) を使用して、異常シャットダウン後にコネクタを再始動したり、System Monitor からリモート・コネクタを始動したりします。

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

デフォルト値は false です。

OADMaxNumRetry

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

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

デフォルト値は 1000 です。

OADRetryTimeInterval

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

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

デフォルト値は 10 です。

PollEndTime

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

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

PollFrequency

これは、前回のポーリングの終了から次のポーリングの開始までの間の間隔です。PollFrequency 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 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。一部のアダプターでは、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 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。

PollFrequency は以下の値のいずれかに設定します。

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

- ワード no。コネクタはポーリングを実行しません。このワードは小文字で入力します。

デフォルト値は 10000 です。

重要: 一部のコネクタでは、このプロパティの使用が制限されています。このようなコネクタが存在する場合には、アダプター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

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

電子メール・メッセージもイベントと見なされます。コネクタは、電子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。

コネクタは、1 回目のポーリングを受けると、メッセージの本文を選出します。これは、本文が添付とも見なされるからです。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、コネクタは本文を無視します。

コネクタは PO の最初の添付を処理します。この添付の MIME タイプには対応する DH 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。

2 回目のポーリングを受けると、コネクタは PO の 2 番目の添付を処理します。この添付の MIME タイプには対応する DH 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。

これが受け入れられると、PO の 3 番目の添付が届きます。

PollStartTime

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

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

RequestQueue

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

デフォルト値は CONNECTOR/REQUESTQUEUE です。

RepositoryDirectory

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

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

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

ResponseQueue

DeliveryTransport が JMS の場合のみ適用可能で、RepositoryDirectory が `<REMOTE>` の場合のみ必須です。

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

RestartRetryCount

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

デフォルト値は 3 です。

RestartRetryInterval

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

デフォルト値は 1 です。

RHF2MessageDomain

WebSphere Message Brokers および WAS のみ

このプロパティにより、JMS ヘッダーのドメイン名フィールド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 WMQI 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。この構成可能なドメイン名により、ユーザーは WMQI ブローカーによるメッセージ・データの処理方法を追跡できます。

サンプル・ヘッダーを以下に示します。

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

デフォルト値は `mrm` ですが、このプロパティには `xml` も設定できます。このプロパティは、DeliveryTransport が JMS に設定されており、かつ WireFormat が CwXML に設定されている場合にだけ表示されます。

SourceQueue

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

JMS イベント・ストアを使用する JMS 対応コネクタでの保証付きイベント・デリバリーをサポートするコネクタ・フレームワークに、JMS ソース・キューを指定します。詳細については、37 ページの『ContainerManagedEvents』を参照してください。

デフォルト値は `CONNECTOR/SOURCEQUEUE` です。

SynchronousRequestQueue

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

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

デフォルトは `CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE` です。

SynchronousResponseQueue

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

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

デフォルトは `CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE` です。

SynchronousRequestTimeout

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

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

デフォルト値は 0 です。

WireFormat

トランスポートのメッセージ・フォーマットです。

- `RepositoryDirectory` がローカル・ディレクトリーの場合は、設定は `CwXML` になります。
- `RepositoryDirectory` の値が `<REMOTE>` の場合には、設定値は `CwBO` です。

WsifSynchronousRequestTimeout

WAS 統合ブローカーでのみ使用されます。

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

デフォルト値は 0 です。

XMLNameSpaceFormat

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

ビジネス・オブジェクト定義の XML 形式でネーム・スペースを `short` と `long` のどちらにするかをユーザーが指定できるようにするための、強力なプロパティです。

デフォルト値は `short` です。

付録 B. Connector Configurator

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

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

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

注:

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

この付録では、次のトピックについて説明します。

- 『Connector Configurator の概要』
- 50 ページの『Connector Configurator の始動』
- 51 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』
- 54 ページの『新しい構成ファイルを作成』
- 57 ページの『構成ファイル・プロパティの設定』
- 66 ページの『グローバル化環境における Connector Configurator の使用』

Connector Configurator の概要

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

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (WebSphere Message Brokers (WMQI) と総称)
- WebSphere Application Server (WAS)

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

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

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

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

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

標準プロパティはすべてのコネクタにより使用されるので、標準プロパティを新規に定義する必要はありません。ファイルを作成すると、Connector Configurator により標準プロパティがこの構成ファイルに挿入されます。ただし、Connector Configurator で各標準プロパティの値を設定する必要があります。

標準プロパティの範囲は、ブローカーと構成によって異なる可能性があります。特定のプロパティに特定の値が設定されている場合にのみ使用できるプロパティがあります。Connector Configurator の「標準のプロパティ」ウィンドウには、特定の構成で設定可能なプロパティが表示されます。

ただし**コネクタ固有プロパティ**の場合は、最初にプロパティを定義し、その値を設定する必要があります。このため、特定のアダプターのコネクタ固有プロパティのテンプレートを作成します。システム内で既にテンプレートが作成されている場合には、作成されているテンプレートを使用します。システム内でまだテンプレートが作成されていない場合には、52 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

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

Connector Configurator の始動

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

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

スタンドアロン・モードでの Configurator の実行

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

これを行うには、以下のステップを実行します。

- 「スタート」>「プログラム」から、「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Tools」>「Connector Configurator」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ます。使用しているブローカーに応じて、ICS、WebSphere Message Brokers、または WAS を選択します。

Connector Configurator を個別に実行して構成ファイルを生成してから、System Manager に接続してこの構成ファイルを System Manager プロジェクトに保存することもできます (57 ページの『構成ファイルの完成』を参照)。

System Manager からの Configurator の実行

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

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

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

既存の構成ファイルを編集するには、以下のステップを実行します。

- 「System Manager」ウィンドウの「コネクタ」フォルダーでいずれかの構成ファイルを選択し、右クリックします。Connector Configurator が開き、この構成ファイルの統合ブローカー・タイプおよびファイル名が上部に表示されます。
- Connector Configurator 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。

コネクタ固有のプロパティ・テンプレートの作成

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

コネクタ固有プロパティのテンプレートを新規に作成するか、または既存のコネクタ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、52 ページの『新規テンプレートの作成』を参照してください。
- 既存のファイルを使用する場合には、既存のテンプレートを変更し、新しい名前でのこのテンプレートを保管します。既存のテンプレートは `¥WebSphereAdapters¥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 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
 - 更新されたメソッド
 - 説明
- **フラグ**
 - 標準フラグ
- **カスタム・フラグ**
 - フラグ

プロパティの一般特性の選択を終えたら、「値」タブをクリックします。

値の指定

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

1. 「値」タブをクリックします。「一般」のパネルに代わって「値」の表示パネルが表示されます。
2. 「プロパティを編集」表示でプロパティの名前を選択します。
3. 「最大長」および「最大複数値」のフィールドに値を入力します。

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. 「プロパティを編集」リストでプロパティを選択し、右マウス・ボタンでクリックします。
2. ダイアログ・ボックスから「追加」を選択します。
3. 新規プロパティ値の名前を入力し、「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 により、XML 文書として入力した情報が、Connector Configurator がインストールされている %bin ディレクトリーの %data¥app の下に保管されます。

新しい構成ファイルを作成

構成ファイルを新規に作成するには、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。

- 「System Manager」ウィンドウで「コネクタ」フォルダーを右クリックし、「新規コネクタの作成」を選択します。Connector Configurator が開き、「新規コネクタ」ダイアログ・ボックスが表示されます。
- スタンドアロン・モードの場合は、Connector Configurator で「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 接続、WebSphere Message Brokers 接続、WAS 接続のいずれかを選択します。
- この章で後述する説明に従って「新規コネクタ」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. 「ファイル」>「新規」>「コネクタ構成」をクリックします。
2. 以下のフィールドを含む「新規コネクタ」ダイアログ・ボックス表示されません。

- **名前**

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

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

- **システム接続**

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

- 「コネクタ固有プロパティ・テンプレート」を選択します。

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

使用するテンプレートを選択し、「**OK**」をクリックします。

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

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

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

既存ファイルの使用

使用可能な既存ファイルは、以下の 1 つまたは複数の形式になります。

- **コネクタ定義ファイル。**

コネクタ定義ファイルは、特定のコネクタのプロパティと、適用可能なデフォルト値がリストされたテキスト・ファイルです。コネクタの配布パッケージ

ジの `¥repository` ディレクトリー内には、このようなファイルが格納されていることがあります (通常、このファイルの拡張子は `.txt` です。例えば、XML コネクタの場合は `CN_XML.txt` です)。

- ICS リポジトリー・ファイル。
コネクタの以前の ICS インプリメンテーションで使用した定義は、そのコネクタの構成で使用されたリポジトリー・ファイルで使用可能になります。そのようなファイルの拡張子は、通常 `.in` または `.out` です。
- コネクタの以前の構成ファイル。
これらのファイルの拡張子は、通常 `*.cfg` です。

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

既存ファイルを使用してコネクタを構成するには、Connector Configurator でそのファイルを開き、構成を修正し、そのファイルを再度保管する必要があります。

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

1. Connector Configurator 内で、「ファイル」>「開く」>「ファイルから」をクリックします。
2. 「ファイル・コネクタを開く」ダイアログ・ボックス内で、以下のいずれかのファイル・タイプを選択して、使用可能なファイルを調べます。
 - 構成 (`*.cfg`)
 - ICS リポジトリー (`*.in`、`*.out`)

ICS 環境でのコネクタの構成にリポジトリー・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。リポジトリー・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. ディレクトリー表示内で、適切なコネクタ定義ファイルへ移動し、ファイルを選択し、「開く」をクリックします。

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

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

構成ファイルの完成

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

構成画面のタイトルには、ファイル内で指定された統合ブローカーとコネクター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、コネクターを構成する前にブローカー値を変更してください。これを行うには、以下のステップを実行します。

1. 「標準のプロパティ」タブで、BrokerType プロパティの値フィールドを選択します。ドロップダウン・メニューで、値 ICS、WMQI、または WAS を選択します。
2. 選択したブローカーに関連付けられているプロパティが「標準のプロパティ」タブに表示されます。ここでファイルを保管するか、または 60 ページの『サポートされるビジネス・オブジェクト定義の指定』の説明に従い残りの構成フィールドに値を入力することができます。
3. 構成が完了したら、「ファイル」>「保管」>「プロジェクトに」を選択するか、または「ファイル」>「保管」>「ファイルに」を選択します。

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

複数のコネクター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、コネクター構成をすべて System Manager プロジェクトに保管するには「すべてプロジェクトに保管」をクリックします。

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

構成ファイル・プロパティの設定

新規のコネクタ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 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ーに対応する複数のタブがあります。

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

- 標準のプロパティ
- コネクター固有のプロパティ
- サポートされているビジネス・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ーが表示される場合があります。

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

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

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

標準プロパティとコネクタ固有プロパティの違いは、以下のとおりです。

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

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

- 背景がグレーのフィールドは、標準のプロパティを表します。値を変更することはできますが、名前の変更およびプロパティの除去はできません。
- 背景が白のフィールドは、アプリケーション固有のプロパティを表します。これらのプロパティは、アプリケーションまたはコネクタの特定のニーズによって異なります。値の変更も、これらのプロパティの除去も可能です。
- 「値」フィールドは構成できます。
- 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。
3. 標準のプロパティの値をすべて入力後、以下のいずれかを実行することができます。
 - 変更内容を破棄し、元の値を保持したままで Connector Configurator を終了するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「いいえ」をクリックします。
 - Connector Configurator 内の他のカテゴリーの値を入力するには、そのカテゴリーのタブを選択します。「標準のプロパティ」(またはその他のカテゴリー) で入力した値は、次のカテゴリーに移動しても保持されます。ウィンドウを閉じると、すべてのカテゴリーで入力した値を一括して保管するかまたは破棄するかを確認するプロンプトが出されます。
 - 修正した値を保管するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「はい」をクリックします。「ファイル」メニューまたはツールバーから「保管」>「ファイルに」をクリックする方法もあります。

アプリケーション固有の構成プロパティの設定

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

1. グリッドの左上端の部分で右マウス・ボタンをクリックします。ポップアップ・メニュー・バーが表示されます。プロパティを追加するときは「追加」をクリックします。子プロパティを追加するには、親の行番号で右マウス・ボタンをクリックし、「子を追加」をクリックします。
2. プロパティまたは子プロパティの値を入力します。
3. プロパティを暗号化するには、「暗号化」ボックスを選択します。
4. 58 ページの『標準コネクタ・プロパティの設定』の説明に従い、変更内容を保管するかまたは破棄するかを選択します。

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

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

コネクタ・プロパティの暗号化

「コネクタ固有プロパティ」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

付録 A 『コネクターの標準構成プロパティ』の 30 ページの『プロパティ値の設定と更新』にある更新メソッドの説明を参照してください。

サポートされるビジネス・オブジェクト定義の指定

コネクタ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 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。詳細は、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

ご使用のブローカーが ICS の場合

ビジネス・オブジェクト定義がコネクタ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ダウン・リストが表示されます。
2. 追加するビジネス・オブジェクトをクリックします。
3. ビジネス・オブジェクトの「エージェント・サポート」(以下で説明) を設定します。
4. 「Connector Configurator」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。追加したビジネス・オブジェクト定義に指定されたサポートを含む、変更されたコネクター定義が、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」ウィンドウの「編集」メニューから、「行を削除」をクリックします。リスト表示からビジネス・オブジェクトが除去されます。
3. 「ファイル」メニューから、「プロジェクトの保管」をクリックします。

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

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

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

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

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

ほとんどのコネクタの場合、選択可能な項目は「最大限の努力」のみです。

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

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

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

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

「メッセージ・セット ID」は、WebSphere Business Integration Message Broker 5.0 のオプションのフィールドです。この ID が提供される場合、一意である必要はありません。ただし、WebSphere MQ Integrator および Integrator Broker 2.1 の場合は、一意の ID を提供する必要があります。

ご使用のブローカーが WAS の場合

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

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

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

関係付けられたマップ (ICS のみ)

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

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

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

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

「関連付けられたマップ」タブには以下のフィールドが表示されます。

- **ビジネス・オブジェクト名**

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

- **関連付けられたマップ**

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

- **明示的**

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

明示的バインディングが必要なのは、特定のサポートされるビジネス・オブジェクトに複数のマップが存在する場合のみです。ICS は、ブート時、各コネクターでサポートされるそれぞれのビジネス・オブジェクトにマップを自動的にバインドしようとしています。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」ウィンドウの「**ファイル**」メニューで、「**プロジェクトに保管**」をクリックします。
4. プロジェクトを ICS に配置します。
5. 変更を有効にするため、サーバーをリブートします。

リソース (ICS)

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

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

メッセージング (ICS)

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

トレース/ログ・ファイル値の設定

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

ログとトレースの値を変更するには、以下の手順を実行します。

1. 「トレース/ログ・ファイル」タブをクリックします。
2. ログとトレースのどちらでも、以下のいずれかまたは両方へのメッセージの書き込みを選択できます。

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

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

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

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

データ・ハンドラー

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

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

構成ファイルの保管

コネクタの構成が完了したら、コネクタ構成ファイルを保管します。Connector Configurator では、構成中に選択したブローカー・モードでファイルを保管します。Connector Configurator のタイトル・バーには現在のブローカー・モード (ICS、WMQI、または WAS) が常に表示されます。

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

- System Manager から、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。
- System Manager から、指定したディレクトリーに *.con 拡張子付きファイルとして保管します。
- 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。
- WebSphere Application Server プロジェクトをセットアップしている場合には、このファイルを WebSphere Application Server プロジェクトに保管することもできます。

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

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

構成ファイルの変更

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

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

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

- Connector Configurator で既存の構成ファイルを開きます。
- 「標準のプロパティー」タブを選択します。
- 「標準のプロパティー」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、プロパティー画面の利用可能なタブおよびフィールド選択がただちに更改され、選択した新規ブローカーに適したタブとフィールドのみが表示されます。

構成の完了

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

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

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

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

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

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

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

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

特記事項

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

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

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

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

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

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

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

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

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

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

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

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

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

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

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

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

著作権使用許諾

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

プログラミング・インターフェース情報

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

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

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

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

商標

以下は、IBM Corporation の商標です。

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

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

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

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

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



IBM WebSphere Business Integration Adapter Framework V2.4