IBM WebSphere Business Integration
Adapters

Adapter for WebSphere MQ Workflow ユーザーズ・ガイド

Adapter バージョン 2.5.0

IBM WebSphere Business Integration
Adapters

Adapter for WebSphere MQ Workflow ユーザーズ・ガイド

Adapter バージョン 2.5.0

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、 131 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 WebSphere MQ Workflow バージョン 2.5.0 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 WebSphere MQ Workflow User Guide
Adapter Version 2.5.0

発　行： 日本アイ・ビー・エム株式会社

担　当： ナショナル・ランゲージ・サポート

第1刷 2004.1

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

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

目次

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、WebSphere MQ Workflow 用のアダプターのインストール、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、お客様に設置場所で製品のサポートや管理を担当するコンサルタント、開発者、およびシステム管理者を対象としています。

本書の前提条件

本書の読者は、WebSphere Business Integration システム、ビジネス・オブジェクトとコラボレーションの開発、WebSphere MQ Workflow アプリケーション、および WebSphere MQ Workflow の実行時とビルド時のコンポーネントについて十分な知識と経験を持っている必要があります。

関連文書

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

以下のサイトから、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

本書では、以下のような規則を使用しています。

Courier フォント	コマンド名、ファイル名、入力情報、システムが画面に出力した情報など、リテラル値を示します。
太字	初出語を示します。
イタリック	変数名または相互参照を示します。
青のアウトライン	青のアウトラインは、マニュアルをオンラインで表示するときにのみ見られるもので、相互参照用のハイパーリンクを示します。アウトラインの内側をクリックすると、参照先オブジェクトにジャンプします。
{ }	構文の記述行の場合、中括弧 { } で囲まれた部分は、選択対象のオプションです。1 つのオプションのみを選択する必要があります。
[ ]	構文の記述行の場合、大括弧 [] で囲まれた部分は、オプションのパラメーターです。
...	構文の記述行の場合、省略符号 ... は直前のパラメーターが繰り返されることを示します。例えば、option[,...] は複数のオプションをコンマで区切って指定できることを意味します。
< >	命名規則により、1 つの名前の個々の要素を互いに区別するために、不等号括弧によって個々の要素が囲まれます。例えば、<server_name><*connector_name*>tmp.log のように使用します。
/、¥	本書では、ディレクトリー・パスに円記号 (¥) を使用します。UNIX システムの場合には、円記号 (¥) をスラッシュ (/) に置き換えてください。すべての製品のパス名は、ご使用のシステムで製品がインストールされているディレクトリーを基準とした相対パス名です。
%*text*% および $*text*	% 記号で囲まれたテキストは、Windows の text システム変数またはユーザー変数の値を示します。UNIX 環境での同等の表記は $*text* です。これは、*text* UNIX 環境変数の値を示します。
ProductDir	IBM WebSphere Business Integration Adapters 製品がインストールされるディレクトリーを表します。

本リリースの新機能

リリース 2.5.0 の新機能

アダプターのインストール情報は、本書から除去されました。この情報の新規掲載場所については 2 章を参照してください。

このリリースから、Adapter for WebSphere MQ Workflow は Microsoft Windows NT でサポートされなくなりました。

リリース 2.4.x の新機能

アダプターは、WebSphere Application Server を統合ブローカーとして使用できるようになりました。

始動スクリプト、ファイル名、およびディレクトリー内の名前 `MQWorkflow` は `WebSphereMQWorkflow` に変更されました。

アダプターは以下のプラットフォーム上で稼働します。

- Solaris 7、8
- AIX 5.x
- HP UX 11.i

今回のリリースでは、アダプターに IBM WebSphere MQ Workflow 3.3.2 または 3.4 が必要です。また、Java API 通信モードはサポートされなくなりました。

リリース 2.3.x での新機能

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。

保証付きイベント・デリバリー機能が拡張されました。詳細は、30 ページの『保証付きイベント・デリバリーの使用可能化』を参照してください。

リリース 2.2.x の新機能

本リリースでは、コネクターは MQ Workflow 3.3.2 XML API 動詞処理をサポートします。詳しくは、11 ページの『XML API の動詞処理』および 23 ページの『XML API を使用するための WebSphere MQ Workflow のアップグレード』を参照してください。新しいコネクター固有プロパティーが追加された結果サポートされる機能については、28 ページの『JavaCorbaApi』を参照してください。

リリース 2.1.x の新機能

コネクターは国際化されています。詳しくは、 17 ページの『ロケール依存型データの処理』および 81 ページの『付録 A. コネクターの標準構成プロパティー』を参照してください。

このガイドでは、このアダプターを ICS と共に使用するための情報を提供します。

注: 保証付きイベント・デリバリー機能を使用するには、ICS のリリース 4.1.1.2 をインストールする必要があります。

リリース 1.2.x の新機能

IBM WebSphere Business Integration Adapter for MQ Workflow には、MQ Workflow 用のコネクターが含まれています。このアダプターは、InterChange Server (ICS) 統合ブローカーと共に動作します。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。アダプターの構成は次のとおりです。

- MQ Workflow に固有のアプリケーション・コンポーネント
- サンプル・ビジネス・オブジェクト
- 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。
  - コネクター・フレームワーク
  - 開発ツール (Business Object Designer と IBM CrossWorlds System Manager を含む)
  - API (CDK を含む)

本書では、このアダプターを ICS と共に使用するため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 ICS バージョン 4.1.1 を併用しないでください。

コネクターは AIX 4.3.3 パッチ・レベル 9 に対応しています。

第 1 章 概要

Connector for WebSphere MQ Workflow は、WebSphere Business Integration Adapter for WebSphere MQ のランタイム・コンポーネントの 1 つです。WebSphere MQ Workflow は IBM のワークフロー管理システムです。

このコネクターを使用すると、WebSphere 統合ブローカーと WebSphere MQ Workflow の間でビジネス・オブジェクトを交換できます。WebSphere MQ Workflow は、WebSphere MQ Workflow クライアント・プロセス (ノード) と通信するだけでなく、本書で説明するコネクターなどの外部アプリケーションとも通信します。

本章では、コネクター・コンポーネントと、ビジネス・インテグレーション・システム・アーキテクチャーについて説明します。本章の内容は次のとおりです。

コネクター・アーキテクチャー

コネクターは、アプリケーション固有のコンポーネントとコネクター・フレームワークから成り立っています。アプリケーション固有のコンポーネントには、特定のアプリケーションに合わせたコードが格納されています。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なので、コネクター・フレームワークは、統合ブローカーとアプリケーション固有のコンポーネントとの仲介役の機能を果たします。コネクター・フレームワークは、統合ブローカーとアプリケーション固有のコンポーネントとの間で以下のようなサービスを提供します。

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

Connector for WebSphere MQ Workflow は、データ構造体の交換を実行し、データ交換に関連するプロセスを制御することにより、コラボレーションと WebSphere MQ Workflow ノードを連結する橋の役割を果たします。コネクターは、外部アプリケーションであるにもかかわらず、WebSphere MQ Workflow システム内の内部「ノード」とほぼ同様に機能します。コネクターは、ノードと同様にプロセス指向の機能を実行します。

WebSphere MQ Workflow ノードは、コネクターの指定されたキューに要求を発行するように構成されます。コネクターは、そのキューをポーリングし、WebSphere MQ Workflow 要求メッセージを検索します。コネクターは、Document Object

Model (DOM) パーサーと XML データ・ハンドラーを使用して、データを抽出し、そのデータを要求および対応するコラボレーションに適したビジネス・オブジェクトに変換します。逆方向の処理では、コネクターはコラボレーションからビジネス・オブジェクト要求を受信し、そのビジネス・オブジェクト要求を MQ Workflow プロセス要求に変換し、そのプロセス要求を MQ Workflow の XML 入力キューに発行します。また、コネクターと MQ Workflow サーバーを直接バインドして、MQ Workflow プロセスへの制御を拡張することもできます。

MQ Workflow システムの XML メッセージ API が使用されるようにコネクターを構成してください。MQ Workflow システムの XML API は、メッセージ要求の非同期処理および同期処理に使用されます。ワークフロー・アクションを起動するために使用される IBM MQ Workflow プログラムである User-defined Program Execution Server (UPES) を作成すると、MQ Workflow ノードは、別のノードと通信する場合と同様にコネクターと通信できます。XML API を使用すると、コネクターは、ビジネス・オブジェクトを要求するコラボレーションからのメッセージに応答し、MQ Workflow 内でアクションを要求するメッセージを伝達します。

MQ Workflow との間で交換されるすべての XML メッセージは、単一の Document Type Definition (DTD) である `WfMessage` に準拠します。Document Object Model Parser (DOM) パーサーは、コネクターがその時点で使用する `WfMessage` から Workflow データ構造体を抽出して、データ構造体を保持するコンテナー・オブジェクトを作成します。

注: MQ Workflow プロセスはさまざまな入出力データ構造体を含むことができますが、コラボレーションとコネクターの間のトランザクションでは、各トランザクションでオブジェクト・タイプを 1 つしか使用できません。MQ Workflow 用のコネクターでこの制限による不都合が生じないようにするには、1 つの要求オブジェクトと 1 つ以上の応答オブジェクトを子オブジェクトとして持つコンテナー・オブジェクトを構築する必要があります。詳細については、37 ページの『メタオブジェクトの構成』を参照してください。

コネクターは、ポーリング中に `WfMessage` に格納されているデータ構造体を識別することにより、どのトップレベル・オブジェクトを作成するかを認識します。具体的には、*<boprefix>* 構成プロパティーに付加されたデータ構造体の名前により、どのトップレベル・オブジェクトを作成するかを決定します。このトップレベル・オブジェクト内の最初の (非メタデータ関連) 子オブジェクトには、データ構造体が取り込まれます。親コンテナー・オブジェクトに割り当てられる動詞は、`WfMessage` 内の `ProgramParameters` フィールドに基づいています。親コンテナー・オブジェクトは、InterChange Server に送付されます。

アプリケーションとコネクターの間の通信方法

概要ですでに説明したように、MQ Workflow 用のコネクターは、XML API の通信モードをサポートします。

XML API を使用すると、コネクターは、MQ Workflow 内でアクションを起動するメッセージを送信したり、コネクターによるポーリング中に MQ Workflow からの同期要求を処理したりできます。コネクターは、要求 MQ メッセージの発行先である固定キューをポーリングし、内容を処理し、応答メッセージを (必要な場合は別

のキューに) 戻します。コネクターに対して発行されるすべての XML メッセージは、ビジネス内容に加えて、実行するコラボレーションや使用する動詞などの処理情報を指示します。

XML API を使用するには、コネクターの入力キューを指定する UPES を構成する必要があります。

コネクターと UPES

コネクターは、XML メッセージ API を使用するときに、新規のイベントがあるかどうかを検査するために MQ Workflow を直接ポーリングすることはありません。MQ Workflow ノードがコネクターのキューなどの外部キューに要求を発行するように、MQ Workflow ノードを構成する必要があります。こうしておくと、コネクターはこれらの外部キューをポーリングできます。

MQ Workflow ノードが UPES を使用して外部キューに要求を発行するように、MQ Workflow ノードを構成します。UPES は、MQ Workflow サーバーからの要求を受け入れるように設計されたプログラムです。UPES は、受け入れた要求に応じてサーバーと対話することができ、追加のデータを検索したり結果を戻したりします。MQ Workflow は、要求から XML メッセージへの (およびその逆方向の) 変換を処理します。

図 1 に示すように、コネクターは、MQ Workflow システム内の UPES ノードとほぼ同様に機能します。コネクターは MQ Workflow サーバーの側からは透過的です。

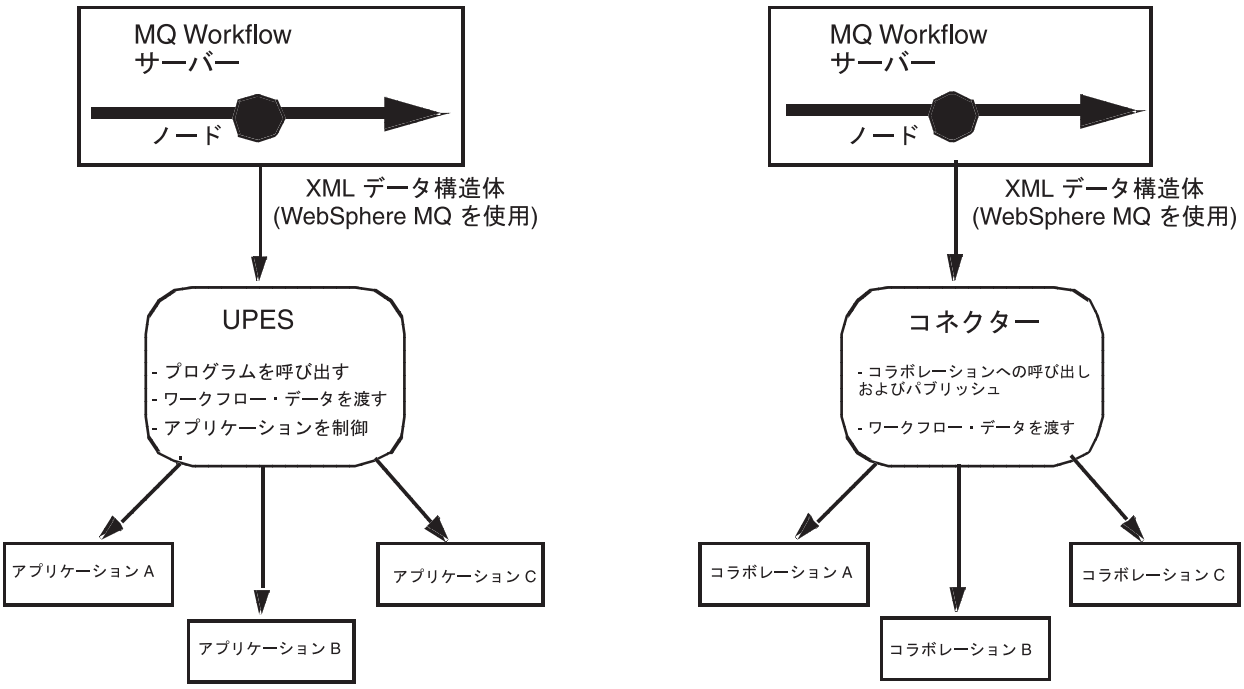

図 1. MQ Workflow UPES としてのコネクター機能

コネクターが開始する要求: XML API

コネクターがコラボレーションの代わりにメッセージを受信すると、コネクターは、MQ Workflow サーバーの XML 入力キューに XML 要求メッセージを発行します。オプションとして (同期的に)、コネクターは、MQ Workflow サーバーによって戻される応答メッセージを待機します。サーバーは、ワークフロー・プロセスを起動し、必要に応じて応答を発行します。

図 2 に、メッセージ要求通信がコラボレーションからコネクターを経由して MQ Workflow に到達するまでを示します。

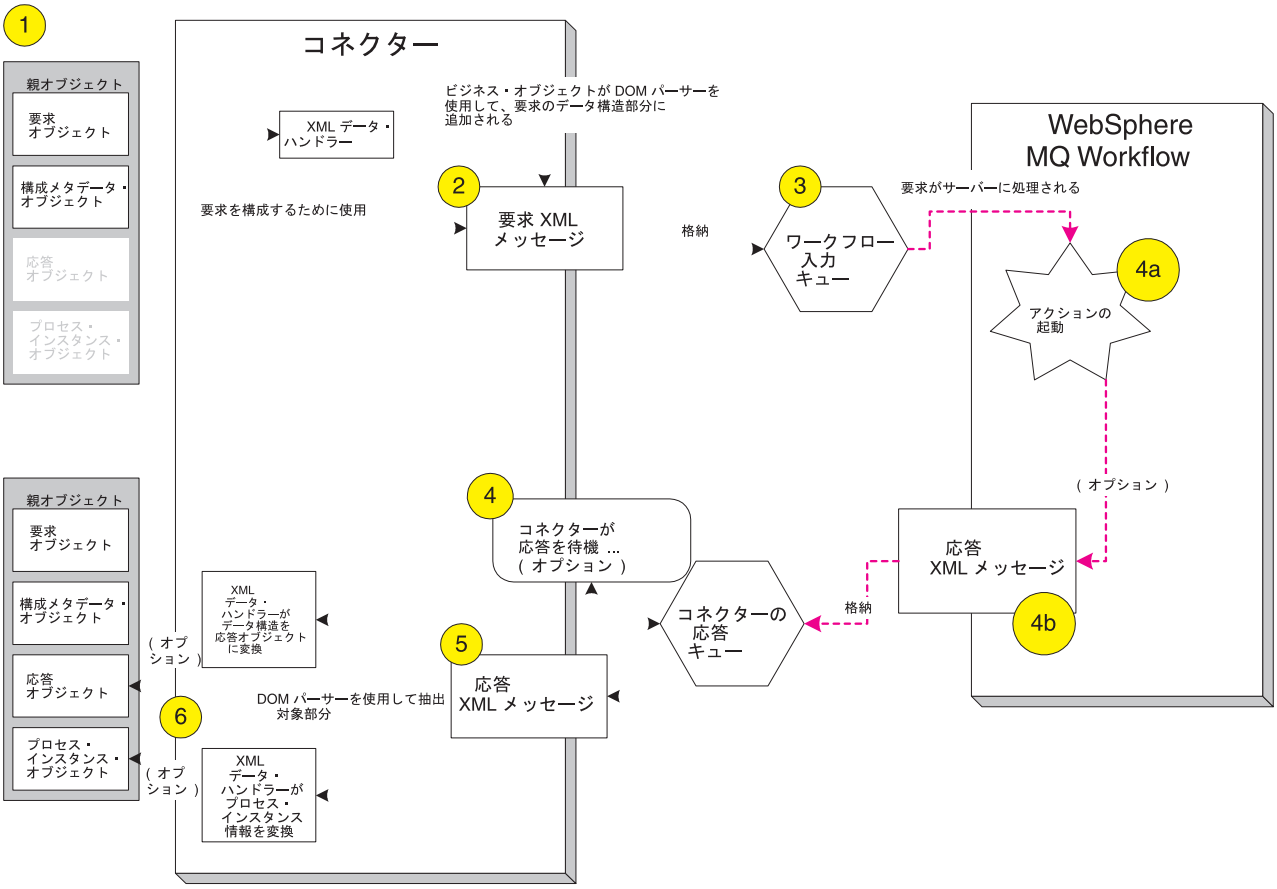

図 2. XML API 通信モード: コネクターが開始する要求

1. コネクターは、MQ Workflow を宛先とするビジネス・オブジェクトの要求をコラボレーションから受信します。
2. コネクターは、XML データ・ハンドラーを使用して、親ビジネス・オブジェクトに格納された要求オブジェクトを XML に変換します。XML 直列化ビジネス・オブジェクトは、DOM パーサーを使用して、MQ Workflow メッセージ DTD に準拠したより大きい XML メッセージに取り込まれます。どのワークフロー・プロセスを作成および実行するかについての情報は、親ビジネス・オブジェクトに格納された構成メタオブジェクト内にカプセル化されます。
3. コネクターは、MQ Workflow サーバーの XML 入力キューに要求を通知します。

4. MQ Workflow サーバーは、要求を受信し、ビジネス・オブジェクトの内容によって指定されたアクションを実行します。これと並行して、コネクターは、a) 正常に復帰する (応答が予期されない場合) か、または b) オプションとして応答メッセージを待機します。
5. コネクターによって発行される要求メッセージに応じて、MQ Workflow サーバーは、並行して実行されているプロセスのプロセス・インスタンス ID (PID) のみが格納された応答を即時に戻すことがあります。MQ Workflow サーバーは、プロセスが完了するまで結果の出力データ構造体を戻さずに待機することもあります。
6. コネクターは、XML DOM パーサーを使用して、応答メッセージから出力データ構造体とプロセス ID を抽出します。必要に応じて、これらの構造体は、XML データ・ハンドラーを使用してビジネス・オブジェクトに変換され、親ビジネス・オブジェクトに追加されます。その後、この親オブジェクトは、待機しているコラボレーションに戻されます。

MQ Workflow が開始する要求

図 3 に、MQ Workflow が開始した、コネクターに対する要求を示します。

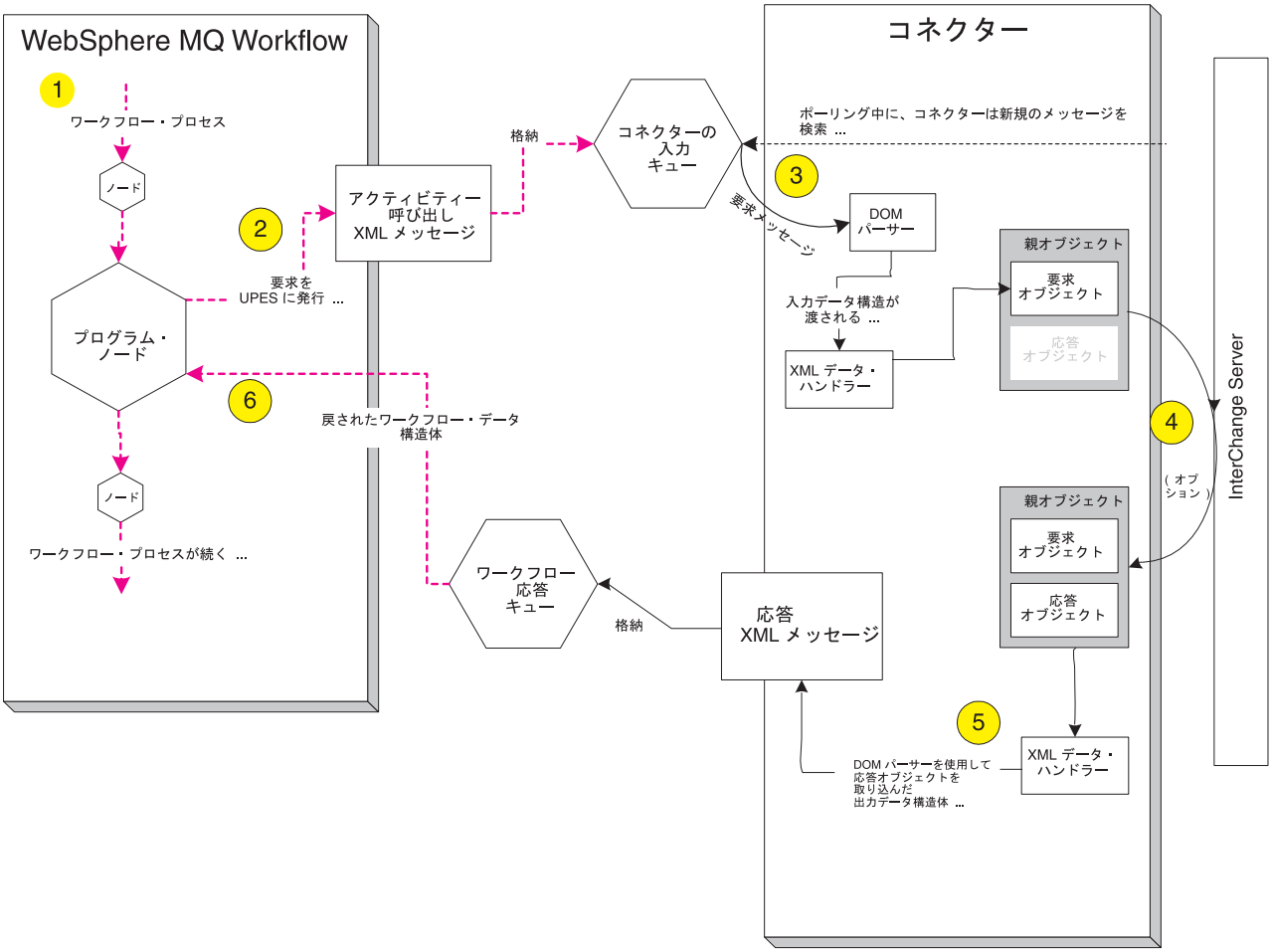

図 3. MQ Workflow が開始する要求

1. ワークフローの実装時に、UPES に送信されたメッセージはコネクターの入力キューに発行されるように UPES が定義されます。
2. コラボレーションからのアクションを要求するために、MQ Server は、要求データ構造体の UPES への経路を指定します (上記の「プログラム・ノード」を参照)。その後、MQ Workflow は、XML ベースのプログラム呼び出しメッセージを、指定されたコネクターの入力キューに送信します。このメッセージは、ワークフロー・データ構造体を格納します。
3. コネクターは、キューのポーリング中にメッセージを検索し、XML DOM パーサーを使用して内容を解析し、XML データ・ハンドラーを使用して入力データ構造体をビジネス・オブジェクトに変換します。
4. UPES (プログラム・ノード) 定義時にユーザー指定によって要求の非同期実行が指定されている場合、コネクターはすべての待機中のコラボレーションにビジネス・オブジェクトを送付し、MQ Workflow サーバーに応答を戻しません。
5. 要求の非同期実行が指定され、コラボレーション名がコマンド行パラメーターで指定されている場合 (62 ページの図 20 を参照)、コネクターは特定のコラボレーションにビジネス・オブジェクトを送付し、戻りを待機します。コネクターはその戻りに基づいて適切な応答メッセージを構成します。応答にはビジネス・オブジェクトの変更点が含まれています。次に、応答は MQ Workflow サーバーに送信されます。
6. 要求の同期実行が指定され、コラボレーション名がコマンド行パラメーターで指定されていない場合 (62 ページの図 20 を参照) コネクターはすべての待機中のコラボレーションにビジネス・オブジェクトを送付し、MQ Workflow サーバーに応答を戻しません。これは前述の非同期の場合と同様ですが、MQ Workflow サーバーは依然としてコネクターからの応答を待機します。したがって、サービス呼び出し要求を使用して対応する応答をコネクターに送信するのはコラボレーションの役割です。
7. 要求元の MQ Workflow UPES (プログラム・ノード) は、応答コードと結果のビジネス・データ構造体を受信します。

イベント処理

イベントは、MQ Workflow UPES がコネクターの入力キューに要求を通知するときに発生します。コネクターは、入力キューのポーリングによって、イベントを検出します。ここでは、イベント処理プロセスについて説明します。

イベント通知

イベントを検出した後で、コネクターは以下のステップを実行します。

1. コネクターは、XML メッセージを DOM パーサーにロードします。
2. コネクターは、これが WfMessage であることを検証し、認定されたテンプレートがあるかどうかを検査します。
3. コネクターは、メッセージに含まれる入力データ構造体を識別して、<boprefix><*data structure name*> というタイプのビジネス・オブジェクトを作成します。例えば、コネクター構成プロパティー *boprefix* の値が WfRequest_ で、コネクターが MyCustomer という名前のデータ構造体を格納するメッセージを受信する場合、コネクターは、その構造体がトップレベル・ビジネス・オブジェクト WfRequest_MyCustomer の子オブジェクトの要件に準拠

すると見なします。コネクターは、XML データ・ハンドラーを使用して、XML データ構造体をビジネス・オブジェクトに変換します。

4. コネクターは、WfMessage 内の要素 ProgramParameters を探し出します。ProgramParameters フィールドは、ユーザーが実際の MQ Workflow 内で指定したアプリケーション・パラメーターを格納します。
5. データ・ハンドラーが作成したビジネス・オブジェクト内で動詞が指定されていない場合、コネクターは、要求メッセージの ProgramParameters 部分で指定されたデータを使用して動詞を設定します。
6. MQ Workflow が要求に対する応答を待機しているかどうか判定するため、コネクターは、XML 文書のコンテナー WfMessageHeader に含まれる要素 ResponseRequired の値が `yes`、`no`、または `iferror` のうちのどれであるかを調べます。`yes` の値は、MQ Workflow が要求の状況についてのコネクターからの応答メッセージを現在待機していることを示します。

   注: ResponseRequired 要素は、MQ Workflow Buildtime 構成 (64 ページの図 22 の Workflow サーバーの定義を参照) のプログラム・アクティビティー・プロパティーのモードに対応します。非同期モードを指定する場合は、`ResponseRequired=no` となり、同期モードを指定する場合は、`ResponseRequired=yes` となります。
7. 応答が予想される場合 (`ResponseRequired=yes`)、コネクターは WfMessage の ProgramParameters 要素に指定されたデータを評価して、要求の処理方法を決定します。
8. ProgramParameters にコラボレーションが指定されている場合、コネクターは `executeCollaboration()` メソッドを使用してコラボレーションに要求を送信します。このメソッドはコラボレーションへの同期要求であるため、メソッドの呼び出しからの戻りには時間がかかる場合があります。メソッドは自身の引き数で応答オブジェクトを戻します。コネクターは応答オブジェクトを取り込んで応答メッセージを生成し、応答を MQ Workflow サーバーに送信します。
9. ProgramParameters にコラボレーションが指定されていない場合、コネクターは gotApplEvent() メソッドを使用して、サブスクライブしているすべてのコラボレーションに要求を通知します。コネクターは、gotApplEvent() メソッドの引き数で示されたビジネス・オブジェクトにメタオブジェクトを取り込みます。メタオブジェクトには MQ Workflow アクティビティー情報 (ActImplCorrelID など) が含まれています。メソッドは要求をコラボレーションに通知するため、コネクターはメソッド呼び出しからの戻りを即時に受け取ります。メソッド呼び出しによって応答オブジェクトが戻されないため、応答メッセージは生成されず、MQ Workflow サーバーに送信されません。その代わりに、コラボレーションはサービス呼び出し要求を使用して、対応する応答オブジェクトを MQ Workflow サーバーに送信する必要があります。応答オブジェクトには MQ Workflow アクティビティー情報 (ActImplCorrelID など) を含むメタオブジェクトを組み込む必要があります。次に、コネクターはサービス呼び出し要求に基づいて適切な応答メッセージを構成します。応答メッセージには、MQ Workflow アクティビティー情報およびコラボレーションからの戻りコードのほか、応答ビジネス・オブジェクトが含まれています。応答メッセージは MQ Workflow サーバーに送信されます。メッセージのビジネスの内容とは無関係な問題のために上記のステップ 8 または 9 の途中でエラーが発生した場合、コネクターはエラーをログに記録し、応答 WfMessage を生成しません。

10. オプションとして、メッセージをアーカイブ・キュー、エラー・キュー、またはアンサブスクライブされたキューにアーカイブすることもできます。

検索

コネクターは、入力キュー内のメッセージを定期的な間隔でポーリングします。コネクターは、メッセージを検出すると、そのメッセージをキューから検索し、WfMessage の内容を取得するために DOM パーサーと XML データ・ハンドラーに渡します。コネクターは、WfMessage から抽出したデータ構造体を使用して、適切なビジネス・オブジェクトと動詞を生成します。イベント失敗時のシナリオについては、72 ページの『エラー処理』を参照してください。

コネクターがメッセージを処理するときには、最初に入力キューへのトランザクション・セッションが開始されます。トランザクションを使用するこの方法では、コネクターがビジネス・オブジェクトを正常にサブミットしたがキュー内でのトランザクションのコミットに失敗した場合に、ビジネス・オブジェクトが 1 つのコラボレーションに 2 回デリバリーされる可能性があります。この問題を回避するために、コネクターは、すべてのメッセージを進行中キューに移動します。進行中キューでは、メッセージは処理が完了するまで保持されます。処理中にコネクターの予期しないシャットダウンが発生した場合、進行中キュー内のメッセージは、元の入力キューに復元されることなくそのまま保持されます。

注: コネクターは、1) メッセージがビジネス・オブジェクトに変換され、2) ビジネス・オブジェクトが InterChange Server にデリバリーされ、3) 戻り値が受信されるまで、進行中キューからメッセージを除去しません。

リカバリー

コネクターの初期化時には、コネクターのシャットダウンなどが原因で完全に処理されなかったメッセージが進行中キュー内にあるかどうかが検査されます。コネクター構成プロパティー InDoubtEvents を使用すると、このようなメッセージのリカバリー処理のための 4 つのオプション (始動時の失敗、再処理、無視、またはエラー・ログの記録) のうちの 1 つを指定できます。

FailOnStartup

FailOnStartup オプションを使用すると、コネクターの初期化時に進行中キュー内のメッセージが検出された場合、エラー・ログが記録され、コネクターは即時にシャットダウンします。メッセージを調べて適切な処置を行う (これらのメッセージを完全に削除するか、または別のキューに移動する) のは、ユーザーまたはシステム管理者の役割です。

Reprocess

Reprocess オプションを使用すると、コネクターの初期化時に進行中キュー内のメッセージが検出された場合、以降のポーリング中にこれらのメッセージが最初に処理されます。コネクターは、進行中キュー内のメッセージをすべて処理した後で、入力キューからのメッセージの処理を開始します。

無視

Ignore オプションを使用すると、コネクターの初期化時に進行中キュー内のメッセージが検出された場合、これらのメッセージは無視されますが、コネクターはシャットダウンしません。コネクターは、入力キューからのメッセージの処理を開始します。

ログ・エラー

LogError オプションを使用すると、コネクターの初期化時に進行中キュー内のメッセージが検出された場合、エラー・ログが記録されますが、コネクターはシャットダウンしません。コネクターは、入力キューからのメッセージの処理を開始します。

アーカイブ

コネクター・プロパティー `ArchiveQueue` が指定され、有効なキューが識別されている場合、コネクターは、正常に処理されたすべてのメッセージのコピーをアーカイブ・キューに格納します。`ArchiveQueue` が未定義の場合、正常に処理されたメッセージは処理後に破棄されます。アンサブスクライブされたメッセージまたはエラーが発生したメッセージのアーカイブの詳細については、 65 ページの『第 4 章 ビジネス・オブジェクトの開発』に記載されている 72 ページの『エラー処理』を参照してください。

保証付きイベント・デリバリー

保証付きイベント・デリバリー機能を使用すると、コネクター・フレームワークで、コネクターのイベント・ストア、JMS イベント・ストア、および宛先の JMS キューの間でイベントが失われたり、2 回送信されたりしないようにすることができます。JMS に対応させるには、コネクターの `DeliveryTransport` 標準プロパティーを JMS に構成する必要があります。このように構成すると、コネクターは JMS トランスポートを使用するため、コネクターと統合ブローカー間の以降の通信はすべてこのトランスポートを介して行われます。JMS トランスポートは、メッセージが最終的に宛先へ配信されることを保証します。JMS トランスポートの役割は、いったんトランザクションのキュー・セッションが開始されたら、コミットが発行されるまではメッセージをキャッシュしておくことです。障害が発生した場合やロールバックが発行された場合は、メッセージが廃棄されます。

注: 保証付きイベント・デリバリー機能を使用しない場合、コネクターがイベントをパブリッシュする時間 (コネクターが gotApplEvent() メソッドを自身の pollForEvents() メソッド内部で呼び出す時間) と、コネクターがイベント・レコードを削除することによってイベント・ストアを更新する (または「イベント送付済み」状況を使用して更新する) 時間との間のわずかな期間に障害が発生する可能性があります。この期間に障害が発生すると、イベントは送信されますが、イベント・レコードは「進行中」状況でイベント・ストア内に残ります。コネクターは再始動時にイベント・ストア内に残ったイベント・レコードを検出して送信するため、結果的にイベントが 2 回送信されることになります。

JMS イベント・ストアを有する、または有していない JMS 対応コネクターで保証付きイベント・デリバリー機能を構成することができます。保証付きイベント・デ

リバリーを使用するようにコネクターを構成する方法については、30 ページの『保証付きイベント・デリバリーの使用可能化』を参照してください。

コネクター・フレームワークがビジネス・オブジェクトを ICS 統合ブローカーに配信できない場合、オブジェクトは (UnsubscribedQueue および ErrorQueue ではなく) FaultQueue に置かれ、状況表示と問題の説明が生成されます。FaultQueue メッセージは MQRFH2 形式で書き込まれます。

ビジネス・オブジェクト要求

ビジネス・オブジェクト要求は、InterChange Server が Connector にビジネス・オブジェクトを送信するときに処理されます。コネクターは、XML データ・ハンドラーを使用して、ビジネス・オブジェクトを MQ Workflow WfMessage 形式のメッセージに変換し、そのメッセージを MQ Workflow に発行します。

WfMessage には、ビジネス・オブジェクトと処理情報が格納されます。WfMessage は、MQ Message Descriptor (MQMD) ヘッダーとともに MQ Message 内にカプセル化されます。MQMD ヘッダーには、コネクターによって値が格納される UserID フィールドが含まれています。(デフォルトでは、コネクターは、コネクター構成プロパティー ApplicationUserID の値を渡します。) MQ Workflow は、MQ Message の MQMD ヘッダー内の UserID フィールドを使用して、コネクターによって送信されるすべての要求の許可検査を実行します。

注: WfMessage データ構造体に取り込まれる子構成メタオブジェクト内で属性 UserID が定義されている場合、この UserID 属性は、コネクター構成プロパティー内で静的に定義された値をオーバーライドして、MQ Workflow に発行されるビジネス・オブジェクトのインスタンスに対して適用されます。

注: ユーザーが MQ Workflow 内でワークフロー・プロセスを開始する許可を持っている場合でも、それ以外に WebSphere MQ を使用してメッセージを発行する許可が必要です。メッセージを発行する許可をユーザーに割り当てる方法については、WebSphere MQ の資料を参照してください。

動詞処理

コネクターは、ワークフロー内のデータ構造体とビジネス・オブジェクトの間の橋渡しをする役割を果たします。ビジネス・オブジェクト内に設定されている動詞に応じてアクションを実行するのは、MQ Workflow の役割です。コネクターが保証できるのは、ワークフローでの内容の正常な受信のみです。したがって、コネクターには、MQ Workflow 製品内のビジネス・オブジェクトを制御する手段はありません。MQ Workflow の性質上、ビジネス・オブジェクトは後続ワークフローのトリガーとしてのみ機能するため、ビジネス・オブジェクトのために永続的なデータ構造体が存在することはできません。

Business Object Handler (BOHandler) は、すべてのビジネス・オブジェクトを、動詞とは無関係に同じ方法で処理します。コネクターは、DOM パーサーを使用して、WfMessage を構築します。

コネクターは、トップレベル・ビジネス・オブジェクトのアプリケーション固有のテキストを検査して、cw_mo_wfptcfg=*XXX* の形式の名前と値のペアが定義されていることを確認します。*XXX* によって識別される子メタオブジェクトが解析され、値が解釈されます。

実行されるテンプレートは、メタオブジェクト属性 ProcessTemplateName によって識別されます。これらのテンプレートは、MQ Workflow へのすべてのコマンドを指定するとともに結果を格納するために必要な構造体を提供します。属性 ProcessInstanceName が指定されている場合、コネクターは既存のインスタンスを実行します。指定されていない場合、コネクターはテンプレートの新規のインスタンスを作成します。テンプレートは、子メタオブジェクトの属性 UserId で識別されるユーザーの権限に基づいて実行されます。この属性が指定されていない場合は、コネクター構成プロパティー ApplicationUserID の値が代わりに使用されます。テンプレートの詳細については、57 ページの『第 3 章 WebSphere MQ Workflow アプリケーションの変更』に記載されている 35 ページの『トップレベルのビジネス・オブジェクトと内容の構成』を参照してください。

コネクターは、以下のビジネス・オブジェクト動詞をサポートしています。これらの動詞によって、既存のワークフロー制御の要求が表されます。

- Delete
- Suspend
- Terminate
- Restart
- Resume

コネクターは、MQ Workflow からメッセージを受信すると、WfMessage 内の ProgramParameters の値に基づいて、ビジネス・オブジェクトの動詞を識別します。ProgramParameters のテキストには、メッセージに含まれるビジネス・オブジェクト動詞を指定する *name =value* ペアが含まれていなければなりません。例えば、Delete 動詞の指定時には、この要素テキストに *name=value* ペアである verb=Delete が含まれます。

しかし、逆方向の処理では、コネクターは MQ Workflow に対して要求を処理するためにどの動詞を使用するかを指示しません。コネクターは、MQ Workflow にビジネス・オブジェクトを発行するときに、ビジネス・オブジェクトの動詞を無視します。コネクターは、ビジネス・オブジェクトを XML に変換し、その内容を MQ Workflow 用の WfMessage に取り込みます。実行されるアクションは、(コラボレーションによってビジネス・オブジェクトのために指定された動詞ではなく) ビジネス・オブジェクトの発行先のワークフローによって決定されます。

XML API の動詞処理

注: MQ Workflow バージョン 3.3.2 以上では、XML API を動詞処理に使用することをお勧めします。

MQ Workflow コネクター XML API を使用して、コラボレーションはワークフロー・プロセスの状況をモニターおよび制御できます。ワークフロー・オペレーションを正常に制御したうえで、コネクターはプロセス・インスタンス・オブジェクト

(MO_MQWorkflow_ProcessInstance) にプロセスの詳細を取り込みます。コネクターは、app-text が ProcessInstance に等しいオブジェクトのすべてを MO_MQWorkflow_ProcessInstance のインスタンスと見なします。

コネクターは、ビジネス・オブジェクトを XML に変換し、その内容を MQ Workflow の WfMessage に取り込みます。ビジネス・オブジェクトに指定された動詞によって、プロセス・インスタンスに対して実行されるアクションが決まります。

XML API を使用する場合、コネクターは MO_MQWorkflow_ProcessInstance に対して以下の動詞をサポートします。

Delete
コネクターは、指定したプロセス・インスタンスを MQ Workflow から削除します。プロセス・インスタンスは、Ready、Finished、または Terminated のいずれかの状態である必要があります。プロセスが存在しない場合、またはプロセスを削除できない場合、コネクターは BON_FAIL を戻します。それ以外の場合、コネクターは新しい状態が取り込まれた MO_MQWorkflow_ProcessInstance オブジェクトを戻します。

Suspend
プロセスが存在しない場合、またはプロセスを中断できない場合、コネクターはワークフロー・プロセスを中断するための要求を発行し、BON_FAIL を戻します。プロセス・インスタンスは Running の状態である必要があります。deep オプションが true の場合は、すべての非独立サブプロセスも中断されます。プロセスが存在しない場合、またはプロセスを中断できない場合、コネクターは BON_FAIL を戻します。それ以外の場合、コネクターは新しい状態が取り込まれた MO_MQWorkflow_ProcessInstance オブジェクトを戻します。

Terminate
コネクターは、プロセス・インスタンスおよびそのすべての非独立サブプロセスを終了します。すべての実行、チェックアウト、および中断アクティビティーが終了します。プロセスは、Running、Suspended、または Suspending のいずれかの状態である必要があります。プロセスが存在しない場合、またはプロセスを終了できない場合、コネクターは BON_FAIL を戻します。それ以外の場合、コネクターは新しい状態が取り込まれた MO_MQWorkflow_ProcessInstance オブジェクトを戻します。

Restart
コネクターはワークフロー・プロセス・インスタンスを再始動するための要求を発行します。完了または終了したトップレベルのプロセス・インスタンスのみが再始動できます。プロセスが存在しない場合、またはプロセスを再始動できない場合、コネクターは BON_FAIL を戻します。それ以外の場合、コネクターは新しい状態が取り込まれた MO_MQWorkflow_ProcessInstance オブジェクトを戻します。

Resume
コネクターは中断された、または中断しているプロセス・インスタンスの処理を再開するための要求を発行します。deep オプションが true の場合は、すべての非独立サブプロセスも再開されます。プロセスが存在しない場合、またはプロセスを再

開できない場合、コネクターは BON_FAIL を戻します。それ以外の場合、コネクターは新しい状態が取り込まれた MO_MQWorkflow_ProcessInstance オブジェクトを戻します。

非同期要求

構成メタオブジェクトの属性 ResponseTimeout が 0 より小さい場合、コネクターは、MQ Workflow サーバーに要求を発行した後で応答を待機しません。プロセスの実行中にエラーが発生した場合に、コラボレーションに通知する手段はありません。図 4 に、非同期要求のサンプルを示します。

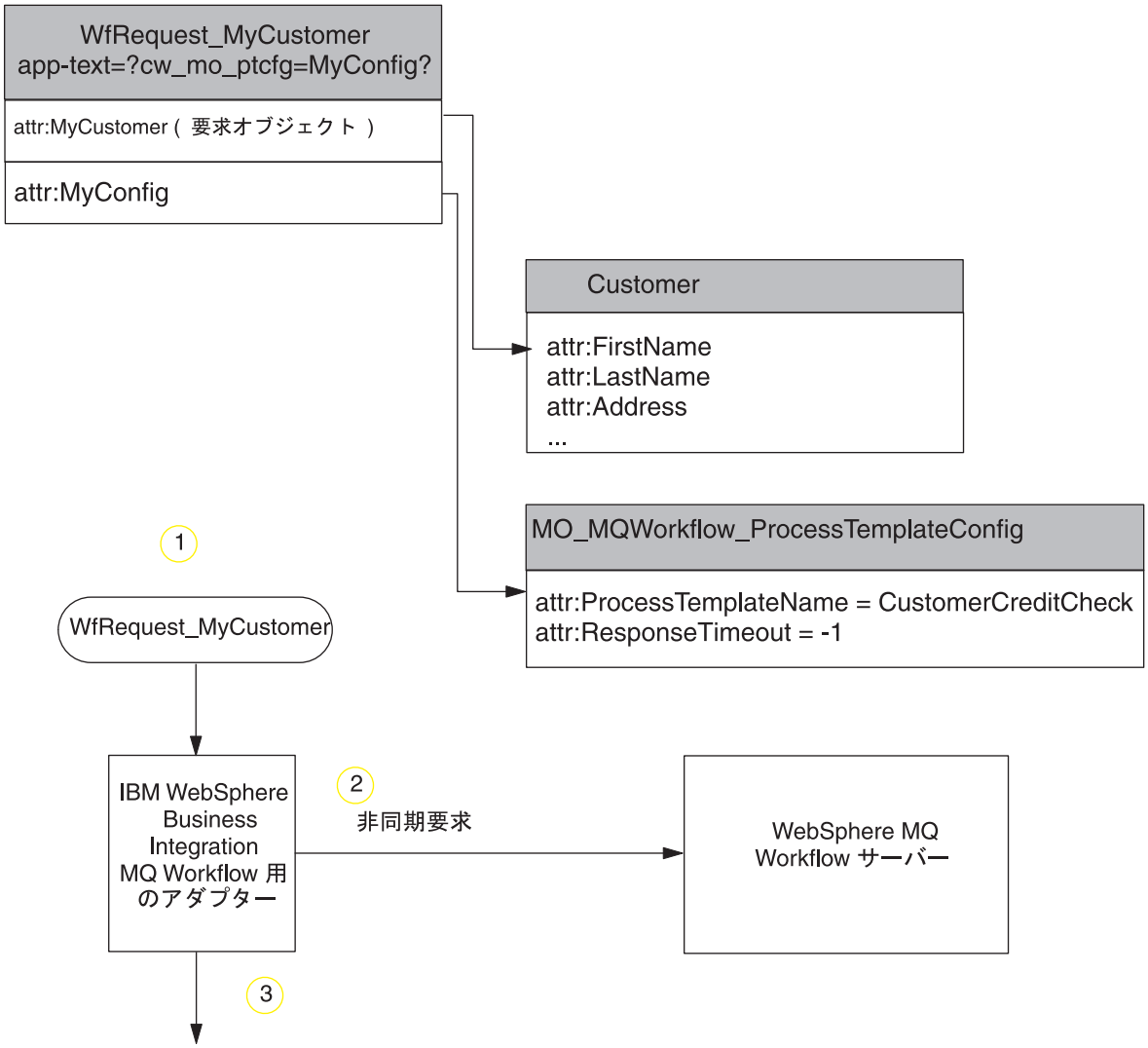

図 4. MQ Workflow への非同期コネクター要求のサンプル

1. コネクターは、負の値が指定された「ResponseTimeout」属性を持つトップレベルのビジネス・オブジェクト WfRequest_MyCustomer を受信します。
2. コネクターは、MQ Workflow サーバーに対して、プロセスに渡すデータ構造体であるオブジェクト「MyCustomer」が格納されている要求を発行します。

3. コネクターは、応答を待機せずに正常に復帰します。コネクターが MQ Workflow サーバーの XML 入力キューへのメッセージの格納に失敗した場合にのみ、エラーが発生します。

プロセス・インスタンス ID に対する非同期要求

子メタオブジェクト内で負以外の `ResponseTimeout` が指定され、属性 `ExecutionMode` が *`Asynchronous`* である場合、コネクターは要求を発行し、コラボレーションにプロセス・インスタンス ID を戻します。プロセス・インスタンス ID が正常に受信されても、対応するワークフロー・プロセスが正常に完了したとは限りません。状況を判別するには、コラボレーションがそのプロセス・インスタンス ID の「Retrieve」を実行する必要があります。この方法は、処理時間が長いトランザクションで役立ちます。図 5 にこの処理を示します。

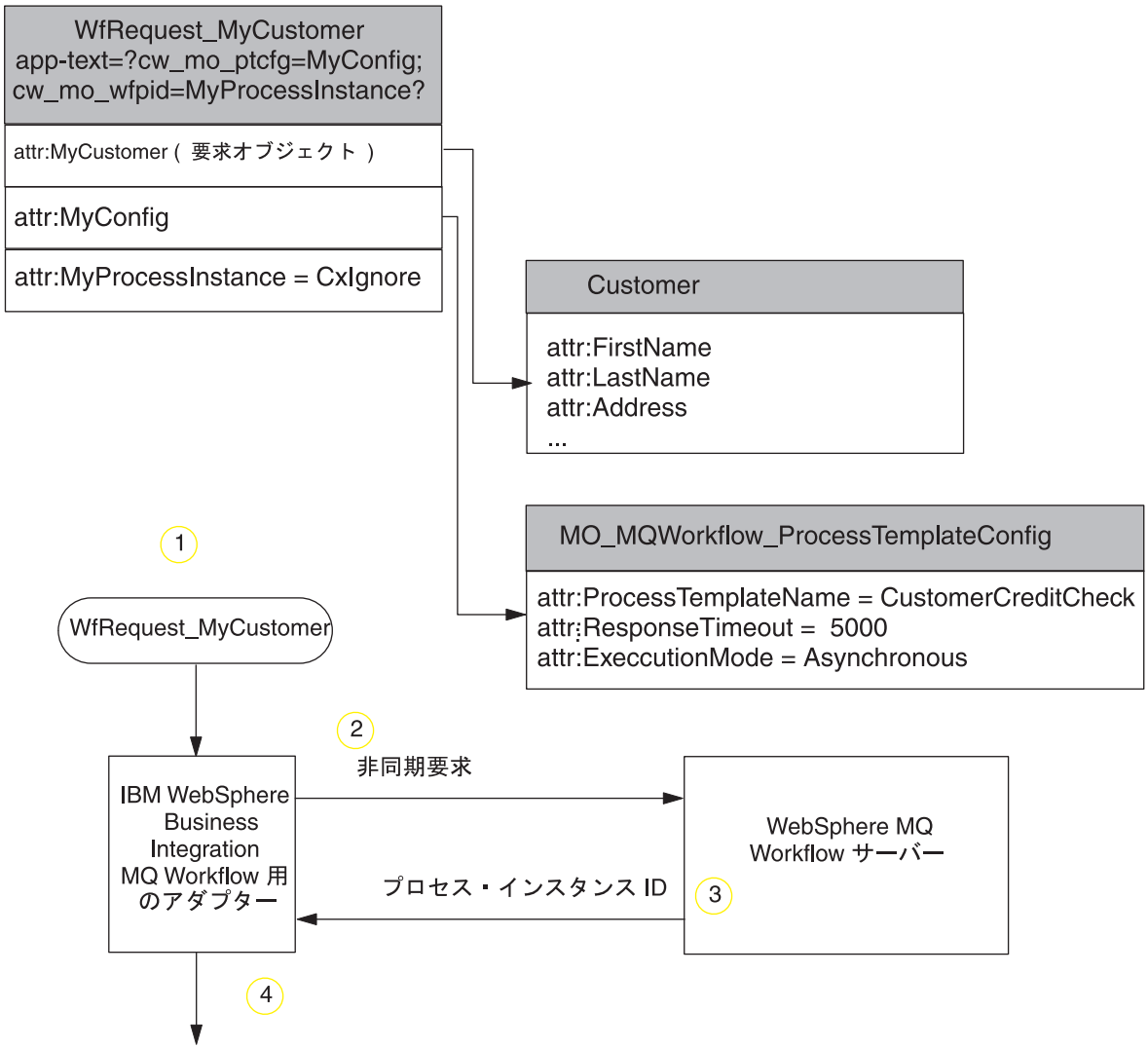

図 5. プロセス・インスタンス ID に対する非同期コネクター要求のサンプル

1. コネクターは、属性 *`ResponseTimeout`* に負以外の値が指定され、属性 `ExecutionMode` に値 *`Asynchronous`* が指定されている、トップレベルのビジネス・オブジェクト `WfRequest_MyCustomer` を受信します。

2. コネクターは、MQ Workflow サーバーに対して、プロセス・テンプレート CustomerCreditCheck の入力データ構造体であるオブジェクト MyCustomer が格納されている要求メッセージを発行します。コネクターは応答を (最大 5000 ミリ秒まで) 待機します。
3. 新規の MQ Workflow プロセス・インスタンス CustomerCreditCheck が生成されます。ワークフロー・プロセスが完了する前に、応答メッセージがコネクターに戻されます。このメッセージには、プロセス・インスタンス ID のみが格納されています。
4. コネクターは、オブジェクト MyProcessInstance にプロセス・インスタンス情報を取り込みます。MQ Workflow がエラー・メッセージを戻す場合、コネクターは BON_FAIL を戻し、ワークフロー・メッセージ内で示されるエラー・メッセージを伝達します。

同期要求

子メタオブジェクトに負以外の値が指定されている ResponseTimeout 属性が含まれ、ExecutionMode 属性に *Synchronous* が指定されている場合には、コネクターは同期要求を発行します。ワークフロー・プロセスが完了するまで、要求は正常に戻されません。同期要求処理では、コラボレーションが開始した MQ Workflow プロセスが成功したか失敗したかを示す情報が、そのコラボレーションに通知されます。処理時間が短いトランザクションの場合、同期処理は即時のフィードバックを生成するための効率的な手段です。図 6 に、同期要求のサンプルを示します。

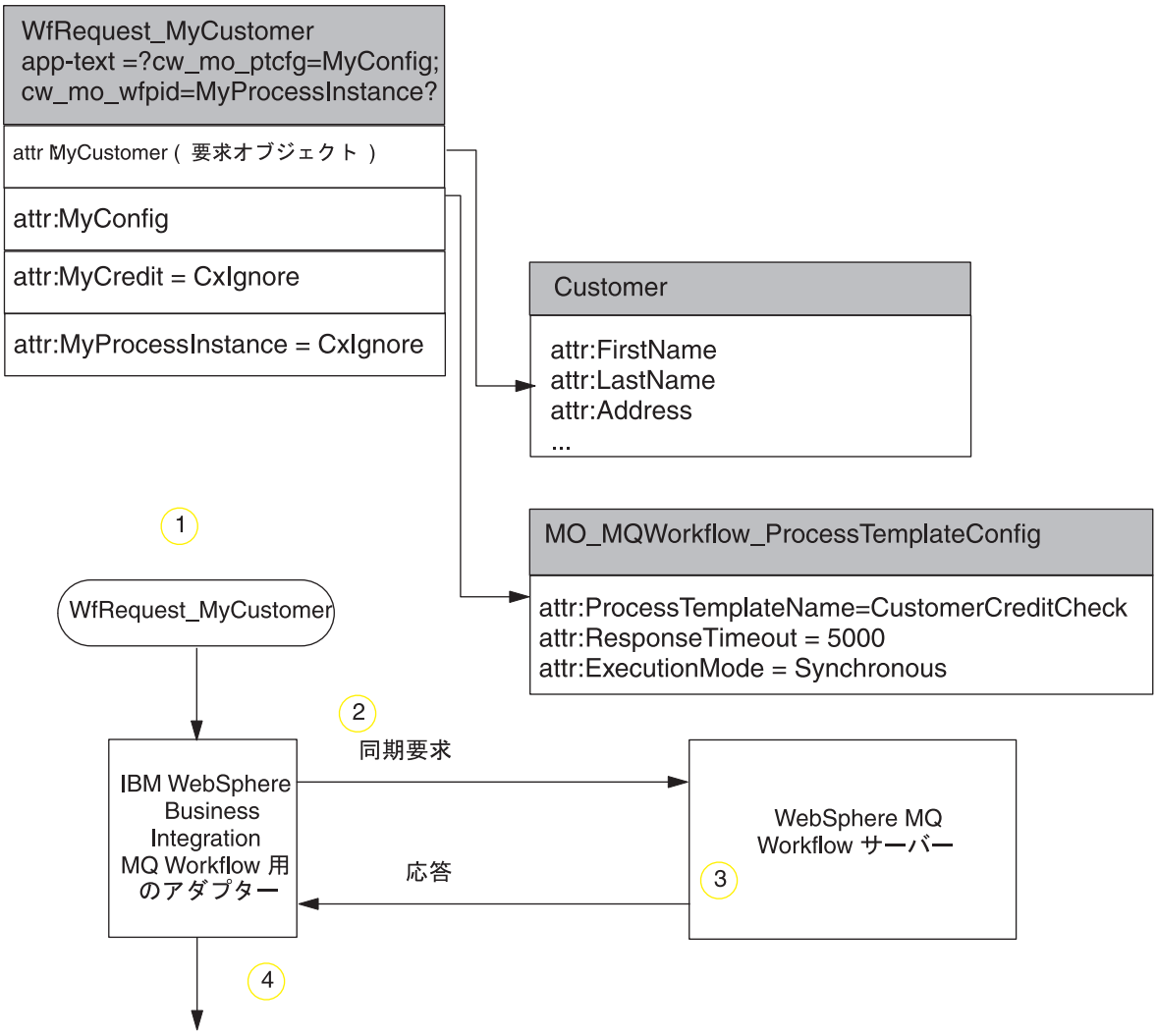

図 6. MQ Workflow への同期コネクター要求のサンプル

1. コネクターは、トップレベルのビジネス・オブジェクト `WfRequest_MyCustomer` を受信します。構成メタオブジェクトでは、属性 `ResponseTimeout` に負以外の値が指定され、属性 `ExecutionMode` に値 *`Synchronous`* が指定されています。
2. コネクターは、MQ Workflow サーバーに対して、プロセス・テンプレート `CustomerCreditCheck` の入力データ構造体であるオブジェクト `MyCustomer` が格納されている要求メッセージを発行します。コネクターは応答を (最大 5000 ミリ秒まで) 待機します。
3. ワーク・プロセス `CustomerCreditCheck` の新規のインスタンスが MQ Workflow 内で生成されます。ワークフロー・プロセスが完了すると、応答メッセージがコネクターに戻されます。

   注: 処理時間が長い同期トランザクションの場合は、プロセス全体を実行できるように、`ResponseTimeout` の値が十分に大きい値であることを確認します。
4. コネクターは、プロセスによって戻されたデータ構造体を応答オブジェクト `MyCreditCheck` に変換します。また、コネクターは、オブジェクト `MyProcessInstance` にプロセス・インスタンス情報を取り込みます。MQ

Workflow がエラー・メッセージを戻す場合、コネクターは BON_FAIL を戻し、MQ Workflow メッセージ内で示されるエラー・メッセージを伝達します。

ロケール依存型データの処理

コネクターは、2 バイト文字セットをサポートし、指定の言語でメッセージ・テキストを配信できるように国際化されています。コネクターが、1 文字コードを使用する地域から異なるコード・セットを使用する地域にデータを転送する場合は、文字変換を実行し、データの意味を保持します。

Java 仮想マシン (JVM: Java Virtual Machine) 内の Java ランタイム環境では、データが Unicode 文字コード・セットで表現されます。Unicode には、既知の文字コード・セット (単一バイトとマルチバイトの両方) における文字のエンコードが、ほぼ一通り組み込まれています。WebSphere Business Integration システムでは、ほとんどのコンポーネントが Java で書かれています。したがって、たいていの統合コンポーネント間のデータ転送の場合には、文字変換の必要はありません。

エラーおよび通知メッセージを国または地域に応じて適切な言語でログに記録するには、ご使用の環境に応じて `Locale` 標準構成プロパティーを構成します。構成プロパティーの詳細については、81 ページの『付録 A. コネクターの標準構成プロパティー』を参照してください。

第 2 章 コネクターのインストールと構成

この章では、IBM WebSphere Business Integration Adapter for WebSphere MQ Workflow のインストール方法および構成方法について説明します。本章の内容は、次のとおりです。

アダプター環境

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

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 WebSphere MQ Workflow バージョン 2.5 は、以下のアダプター・フレームワークと統合ブローカーでサポートされています。

- **アダプター・フレームワーク**:
  WebSphere Business Integration Adapter Framework バージョン 2.3.1 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

- WebSphere Application Server Enterprise、バージョン 5.0.2 (WebSphere Studio Application Developer Integration Edition バージョン 5.0.1 と併用)

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。

WebSphere Application Server については、「*IBM WebSphere Business Integration Adapters アダプター実装ガイド (WebSphere Application Server)*」を参照してください。

アダプターのプラットフォーム

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

オペレーティング・システム:

以下のアプリケーション・プラットフォームのいずれか:

- AIX 5.1、AIX 5.2
- Solaris 7.0、Solaris 8.0
- HP UX 11i
- Windows 2000

サード・パーティーのソフトウェア:

IBM WebSphere MQ Workflow バージョン 3.4

グローバリゼーション

このアダプターは、DBCS (2 バイト文字セット) に対応しています。

前提条件

クライアントと Windows 2000 サーバーをセットアップする必要があります。これを実行するには、「*WebSphere MQ Quick Beginnings for Windows 2000*」の資料を参照してください。

インストール作業の概要

Connector for WebSphere MQ Workflow をインストールするには、以下の作業を実行する必要があります。

- **統合ブローカーのインストール**: この作業では、WebSphere Business Integration システムのインストールと統合ブローカーの始動を行います。作業の詳細については、使用するブローカーおよびオペレーティング・システムのインストール文書に説明があります。
- **アダプターおよび関連ファイルのインストール**: この作業では、アダプター用のファイルをソフトウェア・パッケージから使用システムにインストールします。『アダプターおよび関連ファイルのインストール』を参照。

アダプターおよび関連ファイルのインストール

WebSphere Business Integration adapter 製品のインストールの詳細については、以下の WebSphere Business Integration Adapters Infocenter のサイトにある「*WebSphere Business Integration Adapters インストール・ガイド*」を参照してください。

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

インストール済みファイルの構造

インストール後の製品のパスおよびファイル名については、以降のセクションで説明します。

Windows のコネクター・ファイル構造

インストーラーは、コネクターに関連付けられた標準ファイルをご使用のシステムにコピーします。

このユーティリティーは、コネクター・エージェントを *ProductDir*¥connectors¥WEBSPHEREMQWORKFLOW ディレクトリーにインストールして、「スタート」メニューにコネクター・エージェントへのショートカットを追加します。

表 1 に、コネクターが使用する Windows ファイル構造が記載されており、インストーラーを使用したコネクターのインストールを選択した際に自動的にインストールされるファイルを示します。サンプルのインストール、構成、および実行の方法については、 99 ページの『付録 B. Connector Configurator』を参照してください。

表 1. コネクター用にインストールされた Windows ファイル構造

***ProductDir* のサブディレクトリー**	**説明**
connectors¥WebSphereMQWorkflow¥CWMQWorkflow.jar	WebSphere MQ Workflow コネクターが使用するクラス・ファイルを格納します。
connectors¥WebSphereMQWorkflow¥utilities¥FdlBorgen.jar	FDLBORGEN ユーティリティーが使用するクラス・ファイルを格納します。
repository¥WebSphereMQWorkflow¥CN_WebSphereMQWorkflow.txt	コネクターのリポジトリー定義。
repository¥WebSphereMQWorkflow¥WebSphereMQWorkflow_MetaObjects.txt	コネクターが使用するメタオブジェクトのリポジトリー定義。
connectors¥messages¥WebSphereMQWorkflowConnector.txt	コネクター・メッセージ・ファイル。
connectors¥WebSphereMQWorkflow¥start_WebSphereMQWorkflow.bat	コネクター (2000) の始動スクリプト
connectors¥WebSphereMQWorkflow¥utilities¥fdlborgen.bat	FDLBORGEN ユーティリティーの始動スクリプト (NT/2000)。
connectors¥WebSphereMQWorkflow¥Samples¥WebSphereMQWorkflow_Samples.fdl	サンプル・ワークフローを格納するワークフロー定義ファイル。
connectors¥WebSphereMQWorkflow¥Samples¥Sample_MQWF_Order_Collaborations.in	サンプル・コラボレーションのリポジトリー定義。

表 1. コネクター用にインストールされた Windows ファイル構造 (続き)

***ProductDir* のサブディレクトリー**	**説明**
connectors¥WebSphereMQWorkflow¥Samples¥Sample_MQWF_Order_Connectors.in	サンプル・オブジェクトのリポジトリー定義。
connectors¥WebSphereMQWorkflow¥Samples¥Sample_MQWF_Order_Objects.in	サンプル・オブジェクトのリポジトリー定義。
connectors¥WebSphereMQWorkflow¥Samples¥collaboration_classes¥SampleItemOrderSync.class	コラボレーション・クラスのサンプル
connectors¥WebSphereMQWorkflow¥Samples¥collaboration_classes¥SampleItemSync.class	コラボレーション・クラスのサンプル
connectors¥WebSphereMQWorkflow¥Samples¥collaboration_classes¥SampleWorkflowProcessControl.class	コラボレーション・クラスのサンプル
connectors¥WebSphereMQWorkflow¥Samples¥collaboration_classes¥SampleItemActivityImpl.class	コラボレーション・クラスのサンプル
connectors¥WebSphereMQWorkflow¥Samples¥map_classes¥MQWF_Sample_GBOtoResponse.class	マップ・クラスのサンプル
connectors¥WebSphereMQWorkflow¥Samples¥map_classes¥MQWF_Sample_RequesttoGBO.class	マップ・クラスのサンプル

注: すべての製品のパス名は、ご使用のシステムで製品がインストールされているディレクトリーを基準とした相対パス名です。

UNIX のコネクター・ファイル構造

インストーラーは、コネクターに関連付けられた標準ファイルをご使用のシステムにコピーします。

このユーティリティーにより、コネクター・エージェントが *ProductDir*/connectors/WEBSPHEREMQWORKFLOW ディレクトリーにインストールされます。

表 2 に、コネクターが使用する UNIX ファイル構造が記載されており、インストーラーを介したコネクターのインストールを選択した際に自動的にインストールされるファイルを示します。サンプルのインストール、構成、および実行の方法については、 99 ページの『付録 B. Connector Configurator』を参照してください。

表 2. コネクター用にインストールされた UNIX ファイル構造

***ProductDir* のサブディレクトリー**	**説明**
connectors/WebSphereMQWorkflow/CWMQWorkflow.jar	WebSphere MQ Workflow コネクターが使用するクラス・ファイルを格納します。
connectors/WebSphereMQWorkflow/utilities/FdlBorgen.jar	FDLBORGEN ユーティリティーが使用するクラス・ファイルを格納します。
repository/WebSphereMQWorkflow/CN_WebSphereMQWorkflow.txt	コネクターのリポジトリー定義。
repository/WebSphereMQWorkflow/WebSphereMQWorkflow_MetaObjects.txt	コネクターが使用するメタオブジェクトのリポジトリー定義。
connectors/messages/WebSphereMQWorkflowConnector.txt	コネクター・メッセージ・ファイル。
connectors/WebSphereMQWorkflow/start_WebSphereMQWorkflow.sh	コネクターのシステム始動スクリプト。このスクリプトは、汎用のコネクター・マネージャー・スクリプトから呼び出されます。 System Manager の「コネクター構成」画面をクリックすると、インストーラーは、このコネクター・マネージャー・スクリプト用にカスタマイズされたラッパーを作成します。コネクターの始動と停止には、このカスタマイズ済みラッパーを使用してください。
connectors/WebSphereMQWorkflow/utilities/fdlborgen.sh	FDLBORGEN ユーティリティーの始動スクリプト。
connectors/WebSphereMQWorkflow/Samples/WebSphereMQWorkflow_Samples.fdl	サンプル・ワークフローを格納するワークフロー定義ファイル。
connectors/WebSphereMQWorkflow/Samples/Sample_MQWF_Order_Collaborations.in	サンプル・コラボレーションのリポジトリー定義。
connectors/WebSphereMQWorkflow/Samples/Sample_MQWF_Order_Connectors.in	サンプル・オブジェクトのリポジトリー定義。

表 2. コネクター用にインストールされた UNIX ファイル構造 (続き)

ProductDir のサブディレクトリー	説明
connectors/WebSphereMQWorkflow/Samples/Sample_MQWF_Order_Objects.in	サンプル・オブジェクトのリポジトリー定義。
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleItemOrderSync.class	コラボレーション・クラスのサンプル
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleItemSync.class	コラボレーション・クラスのサンプル
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleWorkflowProcessControl.class	コラボレーション・クラスのサンプル
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleItemActivityImpl.class	コラボレーション・クラスのサンプル
connectors/WebSphereMQWorkflow/Samples/map_classes/MQWF_Sample_GBOtoResponse.class	マップ・クラスのサンプル
connectors/WebSphereMQWorkflow/Samples/map_classes/MQWF_Sample_RequesttoGBO.class	マップ・クラスのサンプル

注: すべての製品のパス名は、ご使用のシステムで製品がインストールされているディレクトリーを基準とした相対パス名です。

XML API を使用するための WebSphere MQ Workflow のアップグレード

XML API を使用するために WebSphere MQ Workflow をアップグレードするには (XML API について詳しくは、 11 ページの『XML API の動詞処理』を参照)、以下の作業を実行する必要があります。

1. WebSphere MQ Workflow 3.3.2 をインストールします。
2. Suspend、Terminate、および Resume に加えて動詞 Delete および Restart をサポートするため、コラボレーションを作成します。XML API は、上記のすべての動詞の使用をサポートします。
3. コネクター固有プロパティー JavaCorbaApi を追加し、その値を false に設定します。
4. 以下の属性を MO_MQWorkflow_ProcessInstance ビジネス・オブジェクトに追加します。

表 3. コネクター固有の構成プロパティー

名前	型	アプリケーション固有の情報
ProcTemplValidFromDate	ストリング	`ProcTemplValidFromDate; type=pcdata;`
ProcInstSuspensionTime	ストリング	`ProcInstSuspensionTime; type=pcdata;`
ProcInstSuspension ExpirationsTime	ストリング	`ProcInstSuspension ExpirationsTime; type=pcdata;`
ExternalProcessContext	ストリング	`ExternalProcessContext; type=pcdata` **注:** InterChange Server でフロー・モニターを使用している場合は、この属性は定義しないでください (以下の 7 を参照)。

または:
Repository フォルダーから repos_copy MQWorkflowMetaObject.txt および CNMQWorkflow.txt を追加します。

5. 動詞 Suspend および Resume を処理するには、以下のアプリケーション固有の情報を MO_MQWorkflow_ProcessInstance に追加します。

```
Suspend      deep=true;
Resume       deep=true;
```

deep=true の場合、すべての非独立サブプロセスも中断または再開されます。deep=false の場合、これらの同じサブプロセスは無視されます。deep のデフォルトは、false です。

6. ワークフロー・プロセスの状況を制御またはモニターするには、(ProcInstID の代わりに) MQ_MQWorkflow_ProcessInstance オブジェクトの ProcInstName を使用します。
7. オブジェクトの ExternalProcessContext が未定義の場合は、InterChange Server は ObjectEvent ID を介してアダプターにフロー・モニター情報を渡します。

コネクターの構成

コネクターの構成プロパティーには、標準構成プロパティーとアダプター固有の構成プロパティーという 2 つのタイプがあります。アダプターを実行する前に、これらのプロパティーの一部の値を設定する必要があります。

Connector Configurator を使用して、コネクター・プロパティーを構成します。

- Connector Configurator およびステップバイステップの手順については、99 ページの『付録 B. Connector Configurator』を参照してください。
- 標準のコネクター・プロパティーについては、『標準コネクター・プロパティー』および 81 ページの『付録 A. コネクターの標準構成プロパティー』を参照してください。
- コネクター固有のプロパティーについては、『コネクター固有のプロパティー』を参照してください。

標準コネクター・プロパティー

標準構成プロパティーは、すべてのコネクターが使用する情報を提供します。これらのプロパティーの資料については、81 ページの『付録 A. コネクターの標準構成プロパティー』を参照してください。Connector Configurator で構成プロパティーを設定するときは、BrokerType プロパティーを使用してブローカーを指定してください。この設定を終えると、ブローカー関連のプロパティーが「Connector Configurator」ウィンドウに表示されます。詳細について、99 ページの『付録 B. Connector Configurator』を参照してください。

コネクター固有のプロパティー

コネクター固有の構成プロパティーは、Connector for WebSphere MQ Workflow が必要とする情報を提供します。また、コネクター固有の構成プロパティーを使用すると、コネクター・エージェントのコード変更や再ビルドを行わなくても、エージェント内の静的情報またはロジックを変更できます。

表 4 に、コネクターのコネクター固有構成プロパティーを示します。プロパティーの説明については、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ー

名前	指定可能な値	デフォルト値	必須
ApplicationPassword	*ログイン・パスワード*	password	いいえ
ApplicationUserID	*ログイン・ユーザー ID*	ADMIN	いいえ
ArchiveQueue	*正常に処理されたメッセージのコピーが送信されるキュー*	MQWFCONN.ARCHIVE	いいえ
MQSeriesCCSID	キュー・マネージャー*接続用文字セット*	ヌル	いいえ
MQSeriesChannel	*MQ サーバー・コネクター・チャネル*	FMCQM.CL.TCP	はい
DataHandlerClassName	データ・ハンドラー・クラス名	com.crossworlds .DataHandlers. text.xml	いいえ
DataHandlerConfigMO	データ・ハンドラー・メタオブジェクト	MO_DataHandler _Default	はい
DataHandlerMimeType	*ファイルの MIME タイプ*	text/xml	いいえ
ErrorQueue	エラーを格納するメッセージ用のキュー	MQWFCONN.ERROR	いいえ
MQSeriesHostName	*WebSphere MQ Workflow* キュー・マネージャーを格納するマシンの名前	ブランク	はい
InDoubtEvents	*FailOnStartup*、*Reprocess*、*Ignore*、または *LogError*	Reprocess	いいえ
InputQueue	*WebSphere MQ Workflow 要求をポーリングする対象のキュー*	CWLDINPUTQ	はい
InProgressQueue	*進行中イベント・キュー*	MQWFCONN.IN_ PROGRESS	はい
JavaCorbaApi	*値が true の場合は、Java CORBA API が使用可能になる*	false	いいえ
OutputQueue	WebSphere MQ Workflow への要求が発行される先のキュー	FMC.FMCGRP.EXE. XML	はい
PollQuantity	入力キューから検索するメッセージの数	1	いいえ
MQSeriesPort	*WebSphere MQ (MQSeries) リスナーのために確立するポート*	14000	はい
ReplyToQueue	コネクターからの*要求発行時に応答メッセージがデリバリーされるキュー*	MQWFCONN.REPLYTO	いいえ
UnsubscribedQueue	アンサブスクライブされたメッセージが送信されるキュー	MQWFCONN. UNSUBSCRIBE	いいえ
MQSeriesQueueManager	Workflow のキュー・マネージャー	FMC (ブランクの場合は、デフォルトのキュー・マネージャーを使用)	いいえ
BOPrefix	サブスクリプション・デリバリーの場合、このプレフィックスにデータ構造体の名前を付加してトップレベル・ビジネス・オブジェクトの名前を決定	MQWF_	いいえ
WorkflowSystemName	ワークフロー・プロセスを直接制御するためにコネクターがバインドする先の WebSphere MQ Workflow システムの名前	FMCSYS	いいえ
WorkflowSystemGroup	ワークフロー・プロセスを直接制御するために接続が確立される先の WebSphere MQ Workflow システム・グループの名前	FMCGRP	いいえ

表 4. コネクター固有の構成プロパティー (続き)

名前	指定可能な値	デフォルト値	必須
WorkflowAgentLocatorPolicy	WebSphere MQ Workflow サーバーへの接続がローカルかリモートかを指定 LOC: コネクターはローカル・マシン上で実行中のサーバーに接続します。 OSA: コネクターは、WebSphere MQ Workflow サーバーにリモートで接続します。	LOC	いいえ
WorkflowAgentName	WebSphere MQ Workflow CORBA エージェントの名前です。		いいえ

ApplicationPassword

WebSphere MQ へのログイン時に UserID とともに使用されるパスワードです。

デフォルト = パスワード。

ApplicationPassword がブランクの場合または除去された場合、コネクターは、WebSphere MQ Workflow が提供するデフォルトのパスワードを使用します。

ApplicationUserID

(Java 直接バインド API の) 接続を許可するためにコネクターが WebSphere MQ Workflow サーバーに渡すユーザー ID です。このプロパティーは以下の目的でも使用されます。

- WebSphere MQ Workflow を使用するメッセージ・デリバリーおよび WebSphere MQ Workflow プロセス実行を認証します。
- ユーザーが (XML API 用の) メタオブジェクト内で属性 UserId を指定しなかった場合に、コネクターが WebSphere MQ Workflow アプリケーションに対する要求操作で使用します。

ApplicationUserID がブランクの場合または除去された場合、コネクターは、WebSphere MQ Workflow が提供するデフォルトのユーザー ID を使用します。

ユーザー ID は、このプロパティーまたはメタオブジェクト属性 UserID に指定することができます。どちらの場合も ApplicationUserID が次の条件を満たしている必要があります。

- ApplicationUserID が WebSphere MQ Workflow アプリケーションに定義されており、指定のワークフローの実行許可などの必要な許可すべてを持っていること。
- ApplicationUserID がローカル・オペレーティング・システムに定義されていること。
- ApplicationUserID がローカル・マシン上で mqm グループのメンバーとなっており、その権限のもとで、WebSphere MQ メッセージを送信できること。

注: ApplicationUserID は、WebSphere MQ Workflow が使用する WebSphere MQ サーバー接続チャネルの MCA プロパティーには指定されません。WebSphere MQ Workflow のデフォルトでは、このプロパティーにユーザー fmc が指定されます。この指定では、アダプターと WebSphere MQ Workflow アプリケーション間でやり取りされるメッセージはすべて、ユーザー fmc の権限で送信され

ます。WebSphere MQ サーバー接続チャネルのプロパティーに指定されている
この値をクリアすると、このコネクター固有のプロパティーに指定した
ApplicationUserID でメッセージを送信することができます。

デフォルト = ADMIN

ArchiveQueue
正常に処理されたメッセージのコピーが送信されるキューです。

デフォルト = MQWFCONN.ARCHIVE

MQSeriesChannel
コネクターが WebSphere MQ と通信するために経由する WebSphere MQ Workflow サーバー・コネクター・チャネルです。

デフォルト = FMCQM.CL.TCP

Channel がブランクの場合または除去された場合、コネクターは、WebSphere MQ Workflow が提供するデフォルトのサーバー・チャネルを使用します。

DataHandlerClassName
ビジネス・オブジェクトとの間でのメッセージ変換に使用するデータ・ハンドラー・クラスです。

デフォルト = com.crossworlds.DataHandlers.text.xml

DataHandlerConfigMO
構成情報を提供するためにデータ・ハンドラーに渡されるメタオブジェクトです。

デフォルト = MO_DataHandler_Default

DataHandlerMimeType
使用すると、特定の MIME タイプに基づいたデータ・ハンドラーを要求できます。

デフォルト = text/xml

ErrorQueue
処理されなかったメッセージが送信されるキューです。

デフォルト = MQWFCONN.ERROR

MQSeriesHostName
WebSphere MQ Workflow のホストであるサーバーの名前です。

デフォルト = ブランク

InDoubtEvents
コネクターの予期しないシャットダウンのために、処理が完了していない進行中イベントの処理方法を指定します。初期化中に進行中キューにイベントが見つかった場合に実行するアクションを、以下の 4 つから選択してください。

- FailOnStartup: エラー・ログを記録して即時にシャットダウンします。

- Reprocess: 残りのイベントを先に処理してから、入力キューのメッセージを処理します。
- Ignore: 進行中キューのすべてのメッセージを無視します。
- LogError: エラー・ログを記録しますが、シャットダウンはしません。

デフォルト = Reprocess

InputQueue
コネクターが新規のメッセージの有無を確認するためにポーリングするメッセージ・キューです。

デフォルト = CWLDINPUTQ

InProgressQueue
進行中イベント・キューです。

デフォルト = MQWFCONN.IN_PROGRESS

JavaCorbaApi
WebSphere MQ Workflow 3.2.2 と共に使用するには、Java CORBA API を使用可能にする必要があります。false の場合、コネクターは WebSphere MQ Workflow 3.3.2 以上と共に使用できるように XML API をサポートします。WebSphere MQ Workflow 3.4 またはそれ以降のリリースを使用する場合、このプロパティーは false でなければなりません。

デフォルト = false

MQSeriesCCSID
WebSphere MQ Workflow のキュー・マネージャーに接続するために使用される CCSID。この値は WebSphere MQ Workflow のキュー・マネージャーの CCSID プロパティーに一致している必要があります。

デフォルト = ブランク (ブランクのままにすると、819 と見なされます。)

選択した文字をサポートするように、CCSID を変更する必要が生じる場合があります。変更する場合は、WebSphere MQ Workflow キューの CCSID と共に CCSID コネクター固有プロパティーも変更する必要があります。

MQSeries CCSID コネクター・プロパティーの変更:
1. System Manager で「MQWF コネクター (MQWF connector)」をダブルクリックする。「コネクター・デザイナー (Connector Designer)」--「MQWorkflowConnector」が開きます。
2. 「アプリケーション構成プロパティー」タブをクリックする。
3. 新しい値 (“943” など) を MQSeriesCCSID プロパティーに入力する。
4. コネクターを再始動する。
5. ICS を再始動する (推奨)。

MQSeries CCSID キュー・プロパティーの変更:
1. コマンド・プロンプトから RUMMQSC FMCQM を実行する。

2. ALTER QMGR CCSID (*new_value*) と入力して、Enter を押す。
3. END と入力して、Enter を押す。

OutputQueue
MQSeries Workflow への要求が発行される先のキューです。

デフォルト = FMC.FMCGRP.EXE.XML

PollQuantity
入力キューから検索するメッセージの数です。

デフォルト = 1

MQSeriesPort
MQSeries (WebSphere MQ) リスナーのために確立するポートです。

デフォルト = 14000

ReplyToQueue
コネクターからの要求発行時に応答メッセージがデリバリーされるキューです。

注: メッセージを WebSphere MQ Workflow アプリケーションに送信するときに、コネクターは、応答を待つかどうかに関わらず、ReplyToQueue フィールドをアウトバウンド・メッセージのヘッダーに追加します。これにより、無効なビジネス・データが MQ Workflow アプリケーションに送信された場合、問題の識別に役立ちます。

デフォルト = MQWFCONN.REPLYTO

UnsubscribedQueue
アンサブスクライブされたメッセージが送信されるキューです。コネクターは、以下の場合にこのキュー・プロパティーにメッセージをデリバリーします。

- コネクターが MQSTR または FMCXML 以外の形式のメッセージを検索する。
- コネクターによって WfMessage が検索されたが、メッセージ名が ActivityImplInvoke タイプまたは General Error タイプのものではない。
- コネクターがサブスクリプション・デリバリーを処理するときにデータ構造体のトップレベル・ビジネス・オブジェクトを検出できない。

デフォルト = MQWFCONN.UNSUBSCRIBE

MQSeriesQueueManager
WebSphere MQ Workflow のキュー・マネージャーです。

デフォルト = FMC (ブランクの場合は、デフォルトのキュー・マネージャーを使用)

BOPrefix
サブスクリプション・デリバリーの場合に、このプレフィックスにデータ構造体の名前を付加します。データ構造体によって、トランザクションのトップレベル・ビジネス・オブジェクトの名前が決定します。

デフォルト = MQWF_

WorkflowSystemName

コネクターがワークフロー・プロセスを直接制御する必要があるときに、接続が確立される先の WebSphere MQ Workflow システムの名前です。

デフォルト = FMCSYS

WorkflowSystemGroup

コネクターがワークフロー・プロセスを直接制御する必要があるときに、接続が確立される先の WebSphere MQ Workflow システム・グループの名前です。

デフォルト = FMCGRP

WorkflowAgentLocatorPolicy

WorkflowSystemName と *WorkflowSystemGroup* のプロパティーによって識別される WebSphere MQ Workflow サーバーへの接続をコネクターが確立する方法を指定します。指定可能な値を以下に示します。

- LOC。コネクターはローカル・マシン上で実行中の WebSphere MQ Workflow サーバーに接続します。
- OSA。コネクターは IBM Java Object Request Broker (ORB) を使用して WebSphere MQ Workflow サーバーにリモートで接続します。IBM Java ORB クライアントをサポートするように WebSphere MQ Workflow を構成する必要があります。詳細については、「*WebSphere MQ Workflow インストール・ガイド*」および「*WebSphere MQ Workflow プログラミング・ガイド*」を参照してください。

デフォルト = LOC

注: IBM Java ORB とのクライアント接続をサポートするには、`start_MQWorkflow.bat` (Windows) ファイルまたは `start_MQWorkflow.sh` (UNIX) ファイルを変更する必要があります。該当する `start_MQWorkflow` ファイルを開き、ステップ 3 で開始されるコメントが表示されるまでスクロールダウンします。この位置以降の行で、与えられた指示に従ってパスを調整します。これにより、初期化中に正しい IBM Java ORB ライブラリーがロードされ、WebSphere MQ Workflow クライアント・ライブラリーによってこのライブラリーが使用されます。この変更は、InterChange Server との通信には影響を及ぼしません。

WorkflowAgentName

WebSphere MQ Workflow CORBA エージェントの名前です。

デフォルト = なし

保証付きイベント・デリバリーの使用可能化

JMS 対応コネクター用の保証付きイベント・デリバリー機能を、以下のいずれかの方法を使用して構成することができます。

- コネクターが JMS イベント・ストア (JMS ソース・キューとしてインプリメントされたもの) を使用している場合、コネクター・フレームワークで JMS イベン

ト・ストアを管理できます。詳細については、『JMS イベント・ストアを使用するコネクターの保証付きイベント・デリバリー』を参照してください。

- コネクターが JMS 以外のイベント・ストア (例えば、JDBC テーブル、E メールのメールボックス、またはフラット・ファイルとしてインプリメントされたもの) を使用している場合、コネクター・フレームワークは、JMS モニター・キューを使用して重複イベントが発生しないようにすることができます。詳細については、33 ページの『JMS 以外のイベント・ストアを使用するコネクターの保証付きイベント・デリバリー』を参照してください。

JMS イベント・ストアを使用するコネクターの保証付きイベント・デリバリー

JMS 対応のコネクターが JMS キューを使用してイベント・ストアを実装している場合、コネクター・フレームワークは「コンテナー」として機能して、JMS イベント・ストア (JMS ソース・キュー) を管理できます。コネクターは、1 つの JMS トランザクション内でソース・キューからメッセージを取り除き、そのメッセージを宛先のキューに配置できます。このセクションでは、JMS イベント・ストアを有する JMS 対応のコネクターでの、保証付きイベント・デリバリー機能の使用について、以下の内容を説明します。

- 『JMS イベント・ストアを使用するコネクターでの機能の使用可能化』
- 33 ページの『イベント・ポーリングへの影響』

JMS イベント・ストアを使用するコネクターでの機能の使用可能化

JMS イベント・ストアを有する JMS 対応コネクターで保証付きイベント・デリバリー機能を使用可能にするには、表 5 に示されているコネクター構成プロパティーを設定します。

表 5. JMS イベント・ストアを使用するコネクターの、保証付きイベント・デリバリー機能関連のコネクター・プロパティー

コネクター・プロパティー	値
DeliveryTransport	JMS
ContainerManagedEvents	JMS
PollQuantity	イベント・ストアを単一ポーリングする際の処理するイベントの数

表 5. JMS イベント・ストアを使用するコネクターの、保証付きイベント・デリバリー機能関連のコネクター・プロパティー (続き)

コネクター・プロパティー	値
SourceQueue	コネクター・フレームワークがポーリングし、処理するイベントを検索するソースとなる、JMS ソース・キュー (イベント・ストア) の名前 **注:** ソース・キューとその他の JMS キューは、同じキュー・マネージャーの一部でなければなりません。コネクターのアプリケーションが、異なるキュー・マネージャーに格納されるイベントを生成する場合は、リモート・キュー・マネージャーのリモート・キュー定義を定義する必要があります。定義を行うと、WebSphere MQ では、イベントをリモート・キューから、JMS 対応コネクターが統合ブローカーとの伝送で使用するキュー・マネージャーへ転送することが可能になります。リモート・キュー定義の構成方法の詳細については、IBM WebSphere MQ の資料を参照してください。

コネクターを構成する他に、JMS ストア内のイベントとビジネス・オブジェクト間の変換を行うデータ・ハンドラーも構成する必要があります。このデータ・ハンドラー情報は、表 6 に示すコネクター構成プロパティーで構成されます。

表 6. 保証付きイベント・デリバリーのデータ・ハンドラー・プロパティー

データ・ハンドラー・プロパティー	値	必須ですか?
MimeType	データ・ハンドラーが処理する MIME タイプ。この MIME タイプは、呼び出すデータ・ハンドラーを識別します。	はい
DHClass	データ・ハンドラーをインプリメントする Java クラスの完全名	はい
DataHandlerConfigMOName	MIME タイプとそのデータ・ハンドラーを関連付けるトップレベルのメタオブジェクトの名前	オプション

注: データ・ハンドラー構成プロパティーは、その他のコネクター構成プロパティーとともに、コネクター構成ファイル内に入っています。

保証付きイベント・デリバリーが使用されるように、JMS イベント・ストアを有するコネクターを構成する場合、表 5 および 表 6 の説明にしたがって、コネクター・プロパティーを設定する必要があります。これらのコネクター構成プロパティーを設定するには、Connector Configurator ツールを使用します。Connector Configurator を使用すると、表 5 で示されたコネクター・プロパティーが「標準のプロパティー」タブに表示されます。同様に、表 6 に示されたコネクター・プロパティーは「データ・ハンドラー」タブに表示されます。

注: Connector Configurator で「データ・ハンドラー」タブに表示されているフィールドが有効になるのは、DeliveryTransport コネクター構成プロパティーが JMS に設定され、かつ ContainerManagedEvents も JMS に設定されている場合のみです。

Connector Configurator の詳細については、99 ページの『付録 B. Connector Configurator』を参照してください。

イベント・ポーリングへの影響

ContainedManagedEvents を JMS に設定して、コネクターで保証付きイベント・デリバリー機能を使用すると、その機能を使用しない場合と比べて、コネクターの動作が多少変化します。コンテナーでイベントを管理するには、コネクター・フレームワークで以下のステップを実行し、イベント・ストアをポーリングする必要があります。

1. JMS トランザクションを開始する。
2. イベント・ストアから JMS メッセージを読み取る。

   イベント・ストアは JMS ソース・キューとして実装されます。 JMS メッセージには、イベント・レコードが含まれています。 JMS ソース・キューの名前は、SourceQueue コネクター構成プロパティーから取得されます。
3. データ・ハンドラーを呼び出して、イベントをビジネス・オブジェクトに変換する。

   コネクター・フレームワークは、32 ページの表 6 で示されたプロパティーで構成されたデータ・ハンドラーを呼び出します。
4. WebSphere MQ Integrator Broker が統合ブローカーである場合、ビジネス・オブジェクトを、構成済みのワイヤー・フォーマット (XML) に基づいたメッセージに変換する。
5. その結果得られたメッセージを JMS 宛先キューに送信する。
   WebSphere ICS 統合ブローカーを使用する場合、JMS 宛先キューへ送信されるメッセージはビジネス・オブジェクトです。WebSphere MQ Integrator Broker を使用する場合は、JMS 宛先キューへ送信されるメッセージは、(データ・ハンドラーが生成した) XML メッセージです。
6. JMS トランザクションをコミットする。

   JMS トランザクションをコミットすると、同一トランザクション内でメッセージが JMS 宛先キューに書き込まれ、JMS ソース・キューから除去されます。
7. ステップ 1 から 6 を繰り返します。繰り返す回数は、PollQuantity コネクター・プロパティーによって決まります。

重要: ContainerManagedEvents プロパティーに JMS が設定されているコネクターでは、イベント・ポーリングを実行する pollForEvents() メソッドは*呼び出されません*。コネクターの基底クラスに pollForEvents() メソッドが含まれていても、このメソッドは*呼び出されません*。

JMS 以外のイベント・ストアを使用するコネクターの保証付きイベント・デリバリー

コネクターが JMS 以外のソリューションを使用してイベント・ストア (例えば、JDBC イベント表、E メールのメールボックス、またはフラット・ファイル) をインプリメントしている場合、コネクター・フレームワークは、重複イベント回避機

能を使用して重複イベントが発生しないようにすることができます。このセクションでは、JMS 以外のイベント・ストアを有する JMS 対応のコネクターでの、保証付きイベント・デリバリー機能の使用について、以下の内容を説明します。

- 『JMS 以外のイベント・ストアを使用するコネクターでの機能の使用可能化』
- 33 ページの『イベント・ポーリングへの影響』

JMS 以外のイベント・ストアを使用するコネクターでの機能の使用可能化: JMS 以外のイベント・ストアを有する JMS 対応コネクターで保証付きイベント・デリバリー機能を使用可能にするには、コネクター構成プロパティーを、表 7 に示されている値に設定してください。

表 7. JMS 以外のイベント・ストアを使用するコネクターの、保証付きイベント・デリバリー機能関連のコネクター・プロパティー

コネクター・プロパティー	値
DeliveryTransport	JMS
DuplicateEventElimination	true
MonitorQueue	コネクター・フレームワークが、処理済みのビジネス・オブジェクトの ObjectEventId を格納する JMS モニター・キューの名前

保証付きイベント・デリバリーを使用するようにコネクターを構成する場合、表 7 に記載されているコネクター・プロパティーを設定する必要があります。これらのコネクター構成プロパティーを設定するには、Connector Configurator ツールを使用します。このツールを使用すると、これらのコネクター・プロパティーが「標準のプロパティー」タブに表示されます。Connector Configurator の詳細については、99 ページの『付録 B. Connector Configurator』を参照してください。

イベント・ポーリングへの影響: DuplicateEventElimination を true に設定して、コネクターで保証付きイベント・デリバリー機能を使用すると、この機能を使用しない場合と比べて、コネクターの動作が多少変化します。重複イベント回避機能を使用するには、コネクター・フレームワークで JMS モニター・キューを使用してビジネス・オブジェクトを追跡します。JMS モニター・キューの名前は、MonitorQueue コネクター構成プロパティーから取得します。

コネクター・フレームワークは、(pollForEvents() メソッドの gotApplEvent() への呼び出しにより) アプリケーション固有のコンポーネントからビジネス・オブジェクトを受け取った後、(gotApplEvents() から受け取った) 現在のビジネス・オブジェクトが重複したイベントを表しているかどうかを判別する必要があります。この判別を行うために、コネクター・フレームワークは JMS モニター・キューからビジネス・オブジェクトを検索し、その ObjectEventId を現在のビジネス・オブジェクトの ObjectEventId と比較します。

- これら 2 つの ObjectEventId が同じであれば、現在のビジネス・オブジェクトが重複イベントであるということになります。このような場合、コネクター・フレームワークは、現在のビジネス・オブジェクトが表すイベントを無視します。つまり、このイベントを統合ブローカーに*送信しません*。
- これらの ObjectEventId が*同じでない* 場合、ビジネス・オブジェクトは重複イベントではありません。このような場合は、コネクター・フレームワークは現在の

ビジネス・オブジェクトを JMS モニター・キューにコピーし、その後これを JMS デリバリー・キューにデリバリーします。これらの作業は、すべて同じ JMS トランザクションの一部として実行されます。JMS デリバリー・キューの名前は、DeliveryQueue コネクター構成プロパティーから取得されます。gotApplEvent() メソッドを呼び出した後は、制御はコネクターの pollForEvents() メソッドに戻ります。

重複イベント回避機能をサポートする JMS 対応コネクターの場合、コネクターの pollForEvents() メソッドで、必ず以下のステップを行う必要があります。

- JMS 以外のイベント・ストアから検索したイベント・レコードからビジネス・オブジェクトを作成した場合は、イベント・レコードの固有イベント ID をビジネス・オブジェクトの ObjectEventId 属性として保管してください。

  アプリケーションは、イベント・ストアのイベント・レコードを一意に識別するため、このイベント ID を生成します。統合ブローカーへイベントを送信してから、このイベント・レコードの状況が変更可能となる前に、コネクターに障害が発生した場合、このイベント・レコードは「進行中」状況のままイベント・ストアに残されます。コネクターが復旧した際に、「進行中」のイベントをリカバリーする必要があります。コネクターは、ポーリングを再開すると、イベント・ストアに残っているイベント・レコードのビジネス・オブジェクトを生成します。ただし、すでに送信済みのビジネス・オブジェクトと新規ビジネス・オブジェクトの両方がそれらの ObjectEventId として同じイベント・レコードを持っているため、コネクター・フレームワークは、新規ビジネス・オブジェクトを重複オブジェクトと認識し、それを統合ブローカーに送信しません。
- コネクターのリカバリー時には、コネクターが新規イベントのためのポーリングを開始する*前*に、「進行中」のイベントを処理するようにしてください。

  コネクターの開始時に、「進行中」のイベントが「ポーリング可能」状況に変更されない限り、ポーリング・メソッドは再処理のためにイベント・レコードを受信しません。

トップレベルのビジネス・オブジェクトと内容の構成

メタデータは、ビジネス・オブジェクトとともに WebSphere MQ Workflow の WfMessage 構造に組み込まれます。この構造は、コネクターと WebSphere MQ Workflow の間で XML メッセージ API を使用して交換されるすべての要求と応答の基礎になります。図 7 にすべてのメッセージの構造を示します。

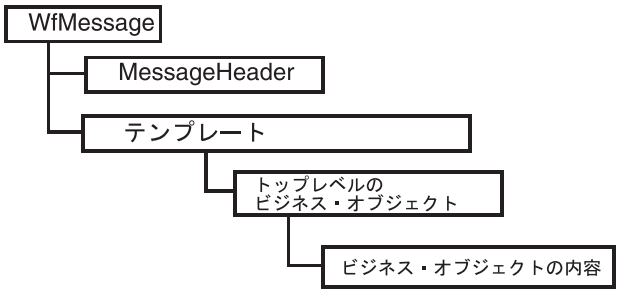

図 7. WebSphere MQ Workflow のメッセージ構造

XML API のコマンドと戻り値は、「テンプレート」に含まれます。これらのテンプレートは、WebSphere MQ Workflow へのすべてのコマンドを指定するとともに結果を格納するために必要な構造体を提供します。テンプレートのタイプは、要求されるアクションに応じて異なります。多くの場合、ビジネス内容はテンプレートの子要素内に格納されます。ビジネス内容を識別するためには、コネクターが各テンプレートを個別の対象として認識する必要があります。テンプレートの名前がそれぞれ異なるのと同様に、子要素の名前もそれぞれ異なります。

コネクターは、3 つのテンプレートとそれに関連付けられた応答構造体を処理できます。

ProcessTemplateExecute
プロセスを同期的または非同期的に実行するために、コネクターが WebSphere MQ Workflow サーバーに送信します。プロセスが非同期的に実行される場合、WebSphere MQ Workflow は応答を発行しません。プロセスが同期的に実行される場合、ワークフロー・プロセスが完了した後で応答が戻されます。ワークフロー入力データ構造体を表すビジネス・オブジェクトは、子要素 ProcInstInputData 内に格納されます。

ProcessTemplateExecuteResponse
コネクターが発行した同期要求への応答として、WebSphere MQ Workflow が送信します。ワークフロー・プロセスからの結果であるビジネス・オブジェクトは、子要素 ProcInstOutputData 内に格納されます。プロセス・インスタンス ID (PID) が戻されますが、この PID はその時点では非アクティブであるため、以降のワークフローを制御するために使用することはできません。

ProcessTemplateCreateAndStartInstance
プロセスを非同期的に実行するために、コネクターが WebSphere MQ Workflow サーバーに送信します。ProcessTemplateExecute テンプレートの場合とは異なり、(ビジネス・オブジェクトではなく) アクティブな PID を含む応答がコネクターに即時に発行されます。この PID は、以降のワークフロー・プロセスを制御するために使用できます。ワークフローを宛先とするデータ構造体を表すビジネス・オブジェクトは、子要素 ProcInstInputData 内に格納されます。

ProcessTemplateCreateAndStartInstanceResponse
コネクターが送信した要求への応答として、WebSphere MQ Workflow が送信します。PID がビジネス・オブジェクトなしで戻されます (ワークフローの非同期実行が想定されているためです)。

ActivityImplInvoke
ビジネス内容を InterChange Server に通知することを要求するために、WebSphere MQ Workflow サーバーがコネクターに送信します。ビジネス・オブジェクトは、子要素 ProgramInputData 内に格納されます。WebSphere MQ Workflow は、同期要求のワークフローに対して戻されるビジネス内容のデフォルト値を格納する追加の子要素 ProgramOutputDataDefault を組み込むことができます。

ActivityImplInvokeResponse
イベント・ポーリング中に処理された同期要求を完了するために、

コネクターが WebSphere MQ Workflow に戻します。コラボレーションが戻すビジネス・オブジェクトは、子要素 ProgramOutputData に追加されます。

処理されたテンプレートの構造に応じて、コネクターは、以下の XML 子要素の 1 つからビジネス・オブジェクトを検索または追加する必要があります。

- ProcInstInputData
- ProcInstOutputData
- ProgramInputData
- ProgramOutputData

メタオブジェクトの構成

ビジネス・オブジェクト本体は、1 つのトップレベルのビジネス・オブジェクトと、データ構造を表す 1 つ以上の子オブジェクトから成り立っています。つまり、各ビジネス・オブジェクトは、メタデータを格納するいくつかの子オブジェクトと、実際のビジネス内容を格納する 1 つ以上の子オブジェクトを保持しています。

トップレベルのビジネス・オブジェクトは、交換されるすべてのオブジェクトのラッパーです。トップレベルのビジネス・オブジェクトは、ラッパーであるため、それ自体ではビジネス・データを保持しません。トップレベルのビジネス・オブジェクトは、交換を開始および完了するために必要なビジネス・オブジェクトの構築に使用されるアプリケーション固有のテキストおよび名前と値のペアを提供します。名前と値のペアは、要求されたビジネス・オブジェクトまたは必須のビジネス・オブジェクトを作成するために必要なメタデータを指定する子属性を指示します。

図 8 に、以下のような場合におけるトップレベルのビジネス・オブジェクトと子オブジェクトの関係を示します。

- コネクターがイベントをポーリングし、WebSphere MQ Workflow から要求を受信する場合
- WebSphere MQ Workflow モードが非同期、または WebSphere MQ Workflow モードが同期でコラボレーション名が WebSphere MQ Workflow コマンド行パラメーターで指定されている場合 (62 ページの図 20 を参照)

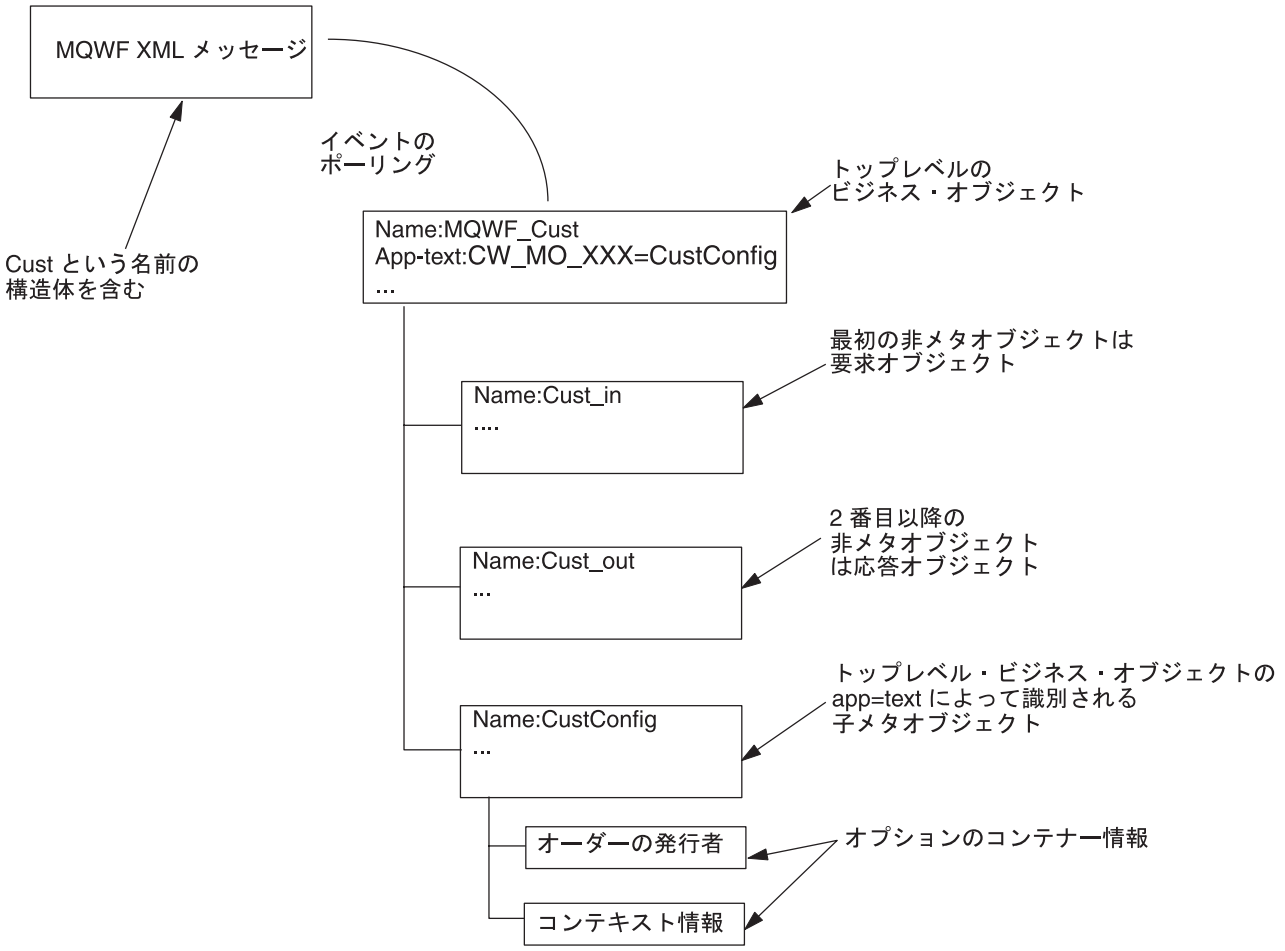

図 8. イベントのポーリング・シナリオでのトップレベル・ビジネス・オブジェクトと子 (ビジネス・オブジェクトとメタオブジェクト)

図 9 に、以下のような場合におけるトップレベルのビジネス・オブジェクトと子オブジェクトの関係を示します。

- コネクターがイベントをポーリングし、WebSphere MQ Workflow から要求を受信する場合
- WebSphere MQ Workflow モードが同期でコラボレーション名が WebSphere MQ Workflow コマンド行パラメーターで指定されていない場合 (62 ページの図 20 を参照)

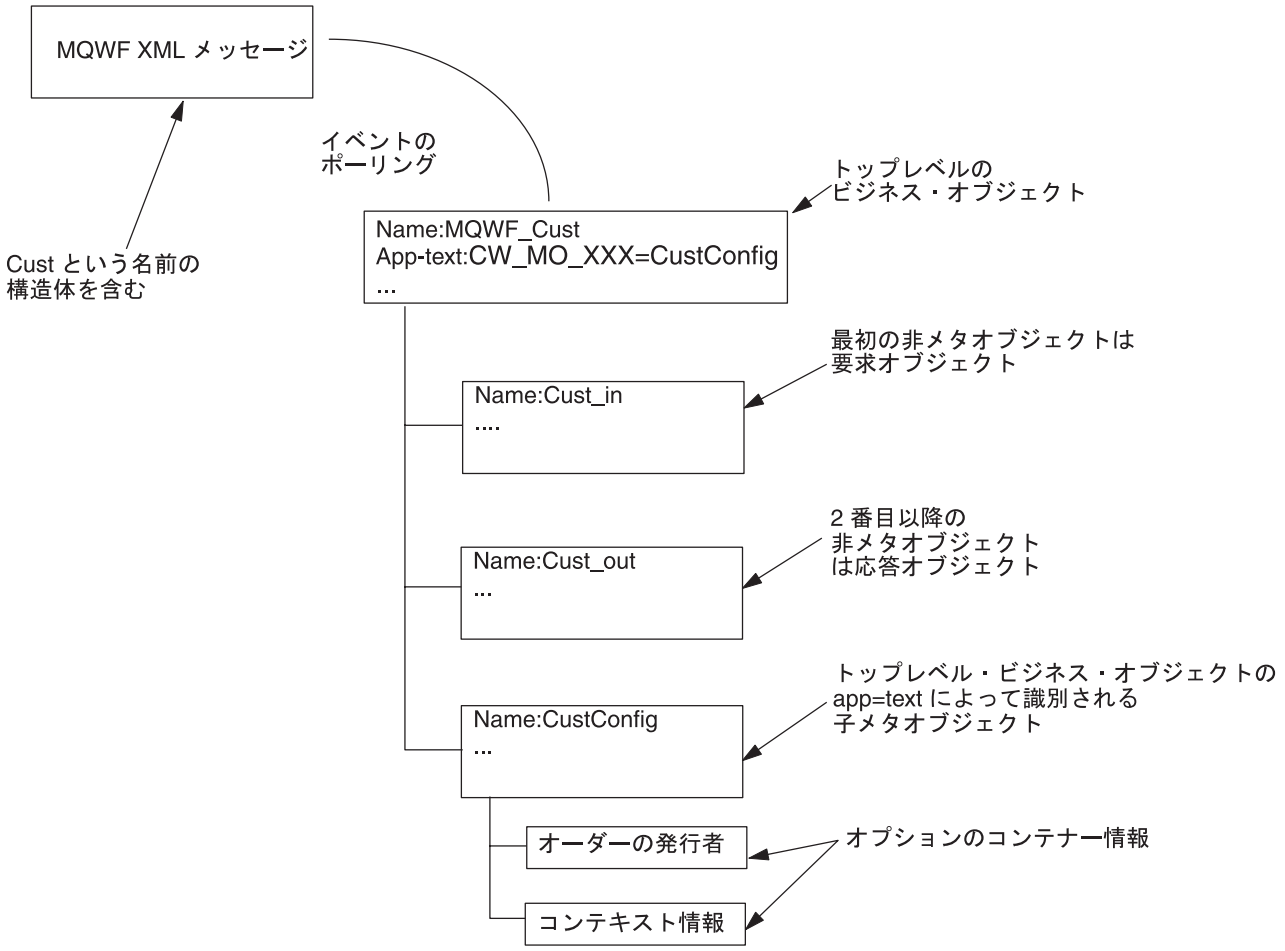

図 9. イベントのポーリング・シナリオでのトップレベル・ビジネス・オブジェクトと子メタオブジェクト

図 10 に、以下のような場合におけるトップレベルのビジネス・オブジェクトと子オブジェクトの関係を示します。

- コネクターがコラボレーションのために WebSphere MQ Workflow に対して要求を発行する場合
- `cw_mo_wfactivityresponse` タグが、トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報 (app-text) に指定されていない場合

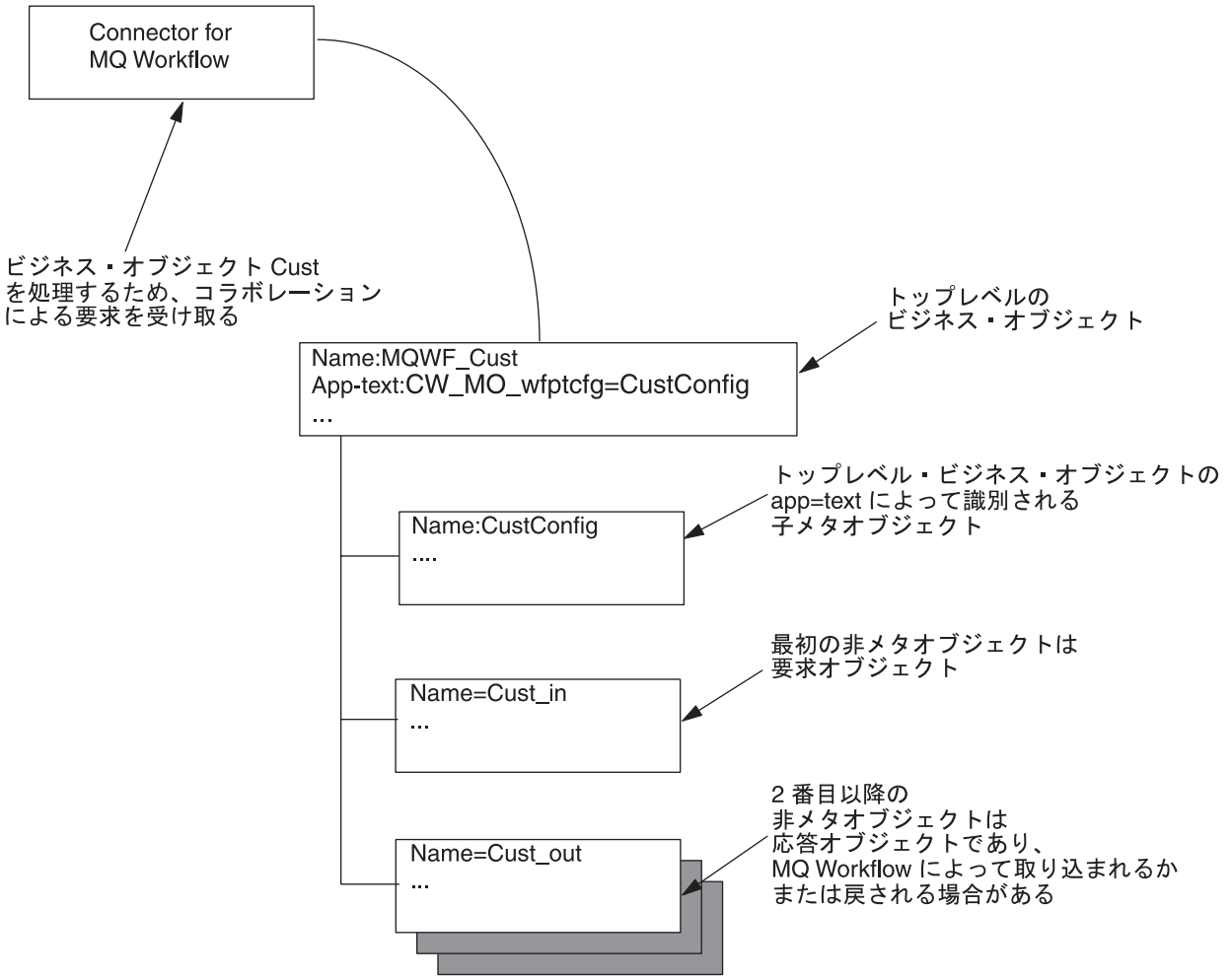

図 10. 要求処理シナリオでのトップレベル・ビジネス・オブジェクトと子オブジェクト (ビジネス・オブジェクトとメタオブジェクト)

図 11 に、以下のような場合におけるトップレベルのビジネス・オブジェクトと子オブジェクトの関係を示します。

- コネクターがコラボレーションのために WebSphere MQ Workflow に対して要求を発行する場合
- `cw_mo_wfactivityresponse` タグが、トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報 (app-text) に指定されている場合

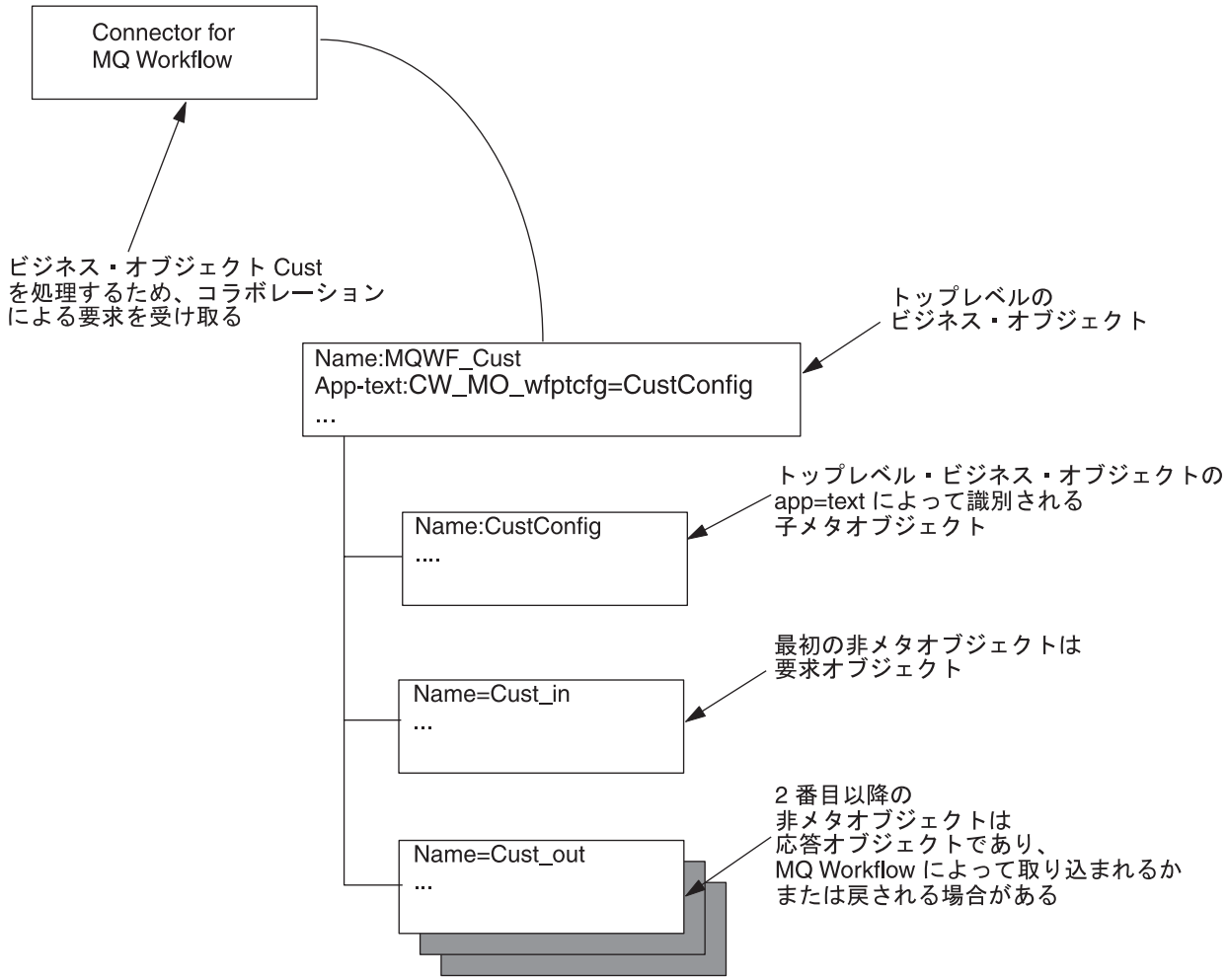

図 11. 要求処理シナリオでのトップレベル・ビジネス・オブジェクトと子オブジェクト (ビジネス・オブジェクトとメタオブジェクト)

WebSphere MQ Workflow プロセスはさまざまな入出力データ構造体を含むことができますが、コラボレーションとコネクターの間のトランザクションでは、各トランザクションでオブジェクト・タイプを 1 つしか使用できません。この制限による不都合が生じないようにするには、1 つの要求オブジェクトと 1 つ以上の応答オブジェクトを子オブジェクトとして持つトップレベルのビジネス・オブジェクトを構成する必要があります。具体的には、*<boprefix>* 構成プロパティーに付加されたデータ構造体の名前により、どのトップレベル・ビジネス・オブジェクトを作成するかを決定します。このトップレベル・ビジネス・オブジェクト内の最初の (非メタデータ関連) 子オブジェクトには、データ構造体が取り込まれます。

子メタオブジェクトとビジネス内容子オブジェクトを区別するために、トップレベルのビジネス・オブジェクトのアプリケーション固有の情報には、以下のタグを組み込む必要があります。

`cw_mo_`*`tag=child meta-object_attribute_name`*

注: メタデータ・タグが複数ある場合は、セミコロンで区切ります。区切り文字の前後の空白文字は無視されます。(例えば、`cw_mo_foo = bar` は `cw_mo_foo=bar` と同じです。)

ここで、tag には以下のいずれかの値を使用します。

- wfptcfg WebSphere MQ Workflow プロセス・テンプレート・メタデータ (『MO_MQWorkflow_ProcessTemplateConfig』を参照)
- wfcontainer WebSphere MQ Workflow コンテナー情報 (43 ページの『MO_MQWorkflow_ContainerInfo』を参照)
- wfpid WebSphere MQ Workflow プロセス・インスタンス情報 (48 ページの『MO_MQWorkflow_ProcessInstance』を参照)
- wfactivityresponse WebSphere MQ Workflow アクティビティー情報 (50 ページの『MO_MQWorkflow_ActivityResponse』を参照)

cw_mo_ で始まるアプリケーション固有の情報は、いずれも構成メタデータ用または動的メタデータ用に予約されています。したがって、コネクター・エージェントは、ビジネス・オブジェクトを受信したときにビジネス・オブジェクト本体のアプリケーション固有の情報を検査するだけで、ビジネス・オブジェクトに実行時メタデータが組み込まれているかどうかを即時に判別できます。同様に、データ・ハンドラーは、ビジネス・オブジェクト・レベルのアプリケーション固有の情報を検査して、直列化または非直列化のプロセスでどの子オブジェクトを追加する (または除外する) かを判別できます。

例えば、構成メタオブジェクトを必要とする WfRequest_MyCustomer という名前のトップレベル・ビジネス・オブジェクトを考えます。MyConfig タイプのオブジェクト構成属性を指定できます。コネクターにメタオブジェクトを認識させ、親オブジェクトを直列化するときにデータ・ハンドラーが「MyConfig」を取り込まないようにするには、アプリケーション固有の情報を WRequest_MyCustomer オブジェクトにタグ cw_mo_wfptcfg=MyConfig の形式で追加します。

WebSphere MQ Workflow の要求または応答を構築するときに、コネクターは、テンプレートを使用してビジネス・オブジェクトを構築します。

MO_MQWorkflow_ProcessTemplateConfig

作成または実行される WebSphere MQ Workflow プロセスについての情報をコネクターに提供するには、トップレベルのビジネス・オブジェクト内にメタオブジェクトを組み込む必要があります。このメタオブジェクトには、使用するプロセス・テンプレートに関する情報、応答が必要かどうかの情報、WebSphere MQ Workflow はプロセスが完了するまで結果を戻さずに待機する必要があるかどうかの情報などが含まれます。これらの情報をメタオブジェクト内に格納することにより、コネクターは、要求された WebSphere MQ Workflow プロセスのアプリケーション固有の情報を動的に構成できます。このメタオブジェクト (または同等のオブジェクト) は、すべてのコラボレーション要求で必須です。

コネクターは、トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報を読み取り、以下の名前と値のペアを検索します。

```
cw_mo_wfptcfg=xxx
```

ここで、xxx は、メタデータを指定する子属性の名前です。表 8 に属性の名前と説明を示します。

表 8. MO_MQWorkflow_ProcessTemplateConfig メタオブジェクトの属性

属性名	説明	許容値
ProcessTemplateName (必須)	実行される WebSphere MQ Workflow テンプレートの名前。 デフォルト = なし	任意
ProcessInstanceName	実行される WebSphere MQ Workflow インスタンスの名前。非同期実行モードでは適用されません。 デフォルト = ブランクの場合は、プロセス・テンプレートの新規のインスタンスを作成。	任意
KeepName	使用後にプロセスを破棄するかどうかを示すフラグ。 デフォルト = false	true または false
UserID	プロセスを実行する許可を持つユーザーを識別します。コネクター固有のプロパティー ApplicationUserID と同じです。この属性の重要な制限については、 26 ページの『ApplicationUserID』を参照してください。 デフォルト = コネクター構成プロパティー ApplicationUserID の値	任意
ResponseTimeout	WebSphere MQ Workflow からの応答を待機する時間の長さ (単位: ミリ秒)。正の値を指定すると、コネクターは応答を待機します。負の値を指定すると、コネクターは WebSphere MQ Workflow 入力キューに要求を発行した後で正常に復帰します。 デフォルト = -1	整数
TimeoutFatal	WebSphere MQ Workflow が応答を受信しない場合、コネクターは InterChange Server に BON_APPRESPONSETIMEOUT を戻し、コネクター・エージェントを終了します。 デフォルト = false (ResponseTimeout が負の値である場合には適用されません)	true または false
ExecutionMode	プロセスがコラボレーションに対して非同期的または同期的のどちらで実行されるかを決定します。このモードが Asynchronous の場合は、プロセス・テンプレートの新規のインスタンスが作成および実行されます。追跡に使用するために、PID がコラボレーションに戻されます。このモードが Synchronous の場合は、プロセス・テンプレートの (既存のまたは新規の) インスタンスが実行されます。ワークフロー・プロセスが完了すると、結果のビジネス・オブジェクトがコラボレーションに戻されます。 デフォルト = Synchronous	Asynchronous または Synchronous

MO_MQWorkflow_ContainerInfo

WebSphere MQ Workflow が発行するアクティビティー呼び出し (ActivityImplInvoke) メッセージでは、ビジネス・オブジェクトに加えて、オプションとしてコンテナー情報も保持できます。このコンテナー情報は、子メタオブジェクト MO_MQWorkflow_ContainerInfo (定義されている場合) にマップしてから、サブスクライブしているコラボレーションにパブリッシュすることができます。このコンテナー情報には、プロセスが発生した条件と環境に関して WebSphere MQ Workflow が提供する情報が含まれます。

注: MO_MQWorkflow_ContainerInfo メタデータは、情報提供のみを目的としています。コラボレーションは、プロセス・モデルまたは要求されたプロセスを開始したユーザーの役割に基づいて、このメタデータを無視したり、さまざまなアクションを実行したりできます。

コネクターは、トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報を読み取り、以下の名前と値のペアを検索します。

```
cw_mo_wfcontainer=XXX
```

ここで、XXX は、トップレベルのビジネス・オブジェクト情報を取り込む子属性の名前です。この情報は、コネクターに対しては意味を持たず、ビジネス・オブジェクトの処理では使用されません。この情報をビジネス・オブジェクトの一部としてコネクターに渡しても無効です。

MO_MQWorkflow_ContainerInfo の定義例を以下に示します。

```
[ReposCopy]

     Version = 3.1.0
  [End]
  [BusinessObjectDefinition]
  Name = MO_MQWorkflow_ProcessInfo
  Version = 1.0.0


    [Attribute]
    Name = Role
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = true
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = Organization
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = ProcessAdministrator
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = Duration
    Type = String
    Cardinality = 1
    MaxLength = 255
    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]
[BusinessObjectDefinition]
Name = MO_MQWorkflow_ActivityInfo
Version = 1.0.0

    [Attribute]
    Name = Priority
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = true
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = MembersOfRoles
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = CoordinatorOfRole
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = Organization
    Type = String
    Cardinality = 1
    MaxLength = 255
```

```
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = OrganizationType
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = LowerLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = UpperLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = People
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PersonToNotify
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration2
```

```
    Type = String
    Cardinality = 1
    MaxLength = 255
    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]
[BusinessObjectDefinition]
Name = MO_MQWorkflow_ContainerInfo
Version = 1.0.0

    [Attribute]
    Name = PROCESS_INFO
    Type = MO_MQWorkflow_ProcessInfo
    ContainedObjectVersion = 1.0.0
    Relationship = Containment
    Cardinality = 1
    MaxLength = 1
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = ACTIVITY_INFO
    Type = MO_MQWorkflow_ActivityInfo
    ContainedObjectVersion = 1.0.0
    Relationship = Containment
    Cardinality = 1
    MaxLength = 1
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = ACTIVITY
    Type = String
    Cardinality = 1
    MaxLength = 1
```

```
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = PROCESS
    Type = String
    Cardinality = 1
    MaxLength = 1
    IsKey = true
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
    [End]
    [Attribute]
    Name = PROCESS_MODEL
    Type = String
    Cardinality = 1
    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]
```

MO_MQWorkflow_ProcessInstance

オプションとして、WebSphere MQ Workflow は、プロセス実行への応答としてプロセス・インスタンス ID (PID) を戻します。ワークフロー・プロセスが正常に作成または実行されると、コネクターは、このオブジェクトにプロセスの詳細を取り込みます。プロセスとそのプロセスを開始したコラボレーションが並行して実行されている場合、コラボレーションは、PID を使用してプロセス・インスタンスを制御できます (プロセスがコラボレーションに対して非同期的に実行された場合)。

コネクターは、親ビジネス・オブジェクトのアプリケーション固有の情報を読み取り、以下の名前と値のペアを検索します。

cw_mo_wfpid

=XXX

ここで、XXX は、プロセス・インスタンス・メタデータを格納する子属性の名前です。オブジェクトは、XML データ・ハンドラーが規定する要件に準拠する必要があります。また、子オブジェクト内の属性の名前には、コネクターに対する意味構造の役割があります。オブジェクトのアプリケーション固有の情報には、「ProcessInstance」を組み込む必要があります。

すべてのカスタム・オブジェクトをこのサンプルから派生させることを強くお勧めします。

表 9. MO_MQWorkflow_ProcessTemplateInstance メタオブジェクトの属性

属性名	説明	許容値
ProcInstID	プロセス・インスタンスを識別する基本キー。	任意
ProcessInstName	WebSphere MQ Workflow のプログラミング・ガイドを参照。	
ProcInstParentName	WebSphere MQ Workflow のプログラミング・ガイドを参照。	
ProcInstTopLevelName	WebSphere MQ Workflow のプログラミング・ガイドを参照。	任意
ProcInstDescription	WebSphere MQ Workflow のプログラミング・ガイドを参照。	整数
ProcInstState	プロセスの状態。	Suspended、Resumed、または Terminated
LastStateChangeTime	WebSphere MQ Workflow のプログラミング・ガイドを参照。	Asynchronous または Synchronous
LastModificationTime	WebSphere MQ Workflow のプログラミング・ガイドを参照。	
ProcTempID	WebSphere MQ Workflow のプログラミング・ガイドを参照。	
ProcTempIName	WebSphere MQ Workflow のプログラミング・ガイドを参照。	
Icon	WebSphere MQ Workflow のプログラミング・ガイドを参照。	
Category	WebSphere MQ Workflow のプログラミング・ガイドを参照。	

MO_MQWorkflow_ActivityRequest

WebSphere MQ Workflow プロセス・インスタンスがアクティビティー呼び出し (ActivityImplInvoke) メッセージを発行し、以下の条件が満たされる場合:

- WebSphere MQ Workflow モードが同期である (64 ページの図 22 を参照)
- コラボレーション名が WebSphere MQ Workflow コマンド行パラメーターで指定されていない (62 ページの図 20 を参照)

コネクターは ActivityImplInvoke メッセージからアクティビティー情報を抽出します。このアクティビティー情報は子メタオブジェクト MO_MQWorkflow_ActivityRequest にマップされ、その後サブスクライブしているコラボレーションにパブリッシュされます。

コネクターは、トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報を読み取り、以下の名前と値のペアを検索します。

cw_mo_wfactivityrequest

=XXX

ここで、XXX は、トップレベルのビジネス・オブジェクト情報を取り込む子属性の名前です。この情報は、コネクターに対しては意味を持たず、ビジネス・オブジェクトの処理では使用されません。表 10 に属性の名前と説明を示します。

表 10. MO_MQWorkflow_ActivityRequest 属性

属性名	説明	許容値
ActImplCorrelID	プロセス・インスタンスが発行した ActivityImplInvoke メッセージを、ActivityImplInvokeResponse メッセージと相関させる ID。コラボレーションは、ActImplCorrelID を使用して、コネクターに応答を送信します。	任意
Starter	プロセス・インスタンスを開始したユーザー ID。	任意
ProcTemplID	プロセス・テンプレートの ID。	任意
ProgramName	プロセス・インスタンスによって呼び出されたプログラム名。	任意
ResponseRequired	プロセス (またはインスタンス) が ActivityImplInvokeResponse メッセージを予期するかどうかを示します。	Yes または No または IfError
ExternalProcessContext	プロセス・インスタンスのプロセス・コンテキスト。	任意

MO_MQWorkflow_ActivityResponse

応答を要求と相関させるため、コネクターでは、適切な情報を含むメタオブジェクトをトップレベルのビジネス・オブジェクトに組み込むことが必要です。このメタオブジェクトには、関連した ActivityImplInvoke メッセージの ActImplCorrelID に関する情報、プロセス・インスタンスに関連したユーザー ID (Starter)、およびプロセス・インスタンスに関連付けられた戻りコードが組み込まれます。このメタオブジェクトが必要となるのは、コラボレーションが ActivityImplInvokeResponse メッセージを ActivityImplInvoke メッセージと関連付ける場合です。

コネクターは、トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報を読み取り、以下の名前と値のペアを検索します。

cw_mo_wfactivityrequest

=XXX

ここで、XXX は、メタデータを指定する子属性の名前です。表 11 に属性の名前と説明を示します。

表 11. MO_MQWorkflow_ActivityResponse 属性

属性名	説明	許容値
ActImplCorrelID	プロセス・インスタンスが、関連した ActivityImplInvoke メッセージによって呼び出された要求と応答とを相関させるために使用する ID。	任意
Starter	ActivityImplInvoke メッセージに関連したプロセス・インスタンスを開始したユーザー ID。	任意
ReturnCode	プロセス・インスタンスに対して発行された戻りコード。	整数

アプリケーション固有の情報

親ビジネス・オブジェクト・レベルのアプリケーション固有の情報は、名前と値のペアで構成され、それらはセミコロンで区切られています。空白文字は無視されます。例えば、次のようにします。

```
cw_mo_wfptcfg=CUST_Config;cw_mo_pid=CUST_IN_Nieman
```

始動ファイルの構成

Connector for WebSphere MQ Workflow を始動する前に、始動ファイルを構成する必要があります。

Windows

Windows プラットフォーム用のコネクターの構成を完成させるには、start_WebSphereMQWorkflow.bat ファイルを変更する必要があります。

1. start_WebSphereMQWorkflow.bat ファイルを開きます。
2. ステップ 1 で開始されるセクションまでスクロールし、使用する WebSphere MQ Java クライアント・ライブラリーの位置を指定します。
3. ステップ 2 で開始されるセクションまでスクロールし、使用する Workflow Java クライアント・ライブラリーの位置を指定します。
4. IBM Java ORB を使用して WebSphere MQ Workflow に接続する場合は、ステップ 3 で開始されるセクションまでスクロールし、使用する IBM Java ORB ライブラリーの位置を指定してから、これらの行のコメントを外します。

UNIX

UNIX プラットフォーム用のコネクターの構成を完成させるには、start_WebSphereMQWorkflow.sh ファイルを変更する必要があります。

1. start_WebSphereMQWorkflow.sh ファイルを開きます。
2. ステップ 1 で開始されるセクションまでスクロールし、使用する WebSphere MQ Java クライアント・ライブラリーの位置を指定します。
3. ステップ 2 で開始されるセクションまでスクロールし、使用する Workflow Java クライアント・ライブラリーの位置を指定します。

4. IBM Java Object Request Broker (ORB) を使用して WebSphere MQ Workflow に接続する場合は、ステップ 3 で開始されるセクションまでスクロールし、使用する IBM Java ORB ライブラリーの位置を指定してから、これらの行のコメントを外します。

コネクターの複数インスタンスの作成

コネクターの複数インスタンスの作成は、多くの点でカスタム・コネクターの作成と似ています。以下に示すステップを実行することによって、コネクターの複数のインスタンスを作成して実行するように、ご使用のシステムを設定することができます。それには、以下の作業を行う必要があります。

- コネクター・インスタンスの新規ディレクトリーを作成する
- 必要なビジネス・オブジェクト定義が存在することを確認する
- 新規コネクター定義ファイルを作成する
- 新規始動スクリプトを作成する

新規ディレクトリーの作成

コネクター・インスタンスごとにコネクター・ディレクトリーを作成する必要があります。このコネクター・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 によりコネクター・インスタンスを一意に識別できます。

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

```
ProductDir¥repository¥connectorInstance
```

ビジネス・オブジェクト定義の作成

プロジェクト内にコネクター・インスタンスごとのビジネス・オブジェクト定義が存在しない場合は、ビジネス・オブジェクト定義を作成する必要があります。

1. 初期コネクター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、適切なファイルをコピーし、Business Object Designer を使用してファイルをインポートします。初期コネクターには任意のファイルをコピーできます。ファイルに変更を加えたら、名前変更してください。
2. 初期コネクターのファイルは、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
   ```

   追加作成したファイルは、ProductDir¥repository の適切な connectorInstance サブディレクトリーに保管する必要があります。

コネクター定義の作成

Connector Configurator のコネクター・インスタンス用構成ファイル (コネクター定義) 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. 初期コネクターの構成ファイル (コネクター定義) をコピーし、ファイルを名前変更する。
2. 各コネクター・インスタンスがサポートしているビジネス・オブジェクト (および関連メタオブジェクト) が正しくリストされていることを確認する。
3. 適宜コネクター・プロパティーをカスタマイズする。

始動スクリプトの作成

始動スクリプトを作成するには、次の作業を実行します。

1. 初期コネクターの始動スクリプトをコピーし、コネクター・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 52 ページの『新規ディレクトリーの作成』で作成したコネクター・ディレクトリーに置く。
3. 始動スクリプトのショートカットを作成する (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て、コマンド行から新規コネクター・インスタンスに合うように初期コネクター名を変更する。

これにより、Integration Server で両方のコネクター・インスタンスを同時に実行できます。

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。

コネクターの始動

コネクターは、**コネクター始動スクリプト**を使用して明示的に開始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・ディレクトリーに存在していなければなりません。

ProductDir¥connectors¥*connName*

このディレクトリー名の *connName* はコネクターを表しています。表 12 が示すとおり、始動スクリプトの名前はオペレーティング・システム・プラットフォームにより異なります。

表 12. 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 系システム	connector_manager_*connName*
Windows	start_*connName*.bat

以下のいずれかの方法で、コネクター始動スクリプトを起動できます。

- 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。または、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。

- コマンド行から起動する。
  - Windows システム:

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

  - UNIX 系システム:

    ```
    connector_manager_connName -start
    ```

  ここで、*connName* はコネクター名であり、*brokerName* はご使用の統合ブローカーを表しています。以下に例を示します。

  - WebSphere InterChange Server では、*brokerName* に ICS インスタンスの名前を指定する
  - WebSphere Message Brokers (WebSphere MQ Integrator、WebSphere MQ Integrator Broker、または WebSphere Business Integration Message Broker)、または WebSphere Application Server では、*brokerName* にブローカーを識別するストリングを指定する

  注: Windows システム上の WebSphere Message Broker または WebSphere Application Server では、-c オプションの後にコネクター構成ファイルの名前を指定する必要があります。ICS では -c はオプションです。

- Adapter Monitor から起動する (WebSphere Business Integration Adapters 製品のみ)。System Manager を始動すると起動されます。

  このツールを使用して、コネクターをロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除できます。
- System Monitor から起動する (WebSphere InterChange Server 製品のみ)

  このツールを使用して、コネクターをロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除できます。
- Windows システムでは、コネクターを Windows のサービスとして開始されるように構成できる。この場合、自動サービスでは Windows システムがブートするとき、手動サービスでは「Windows サービス」ウィンドウからサービスを開始するときに、コネクターが開始されます。

コマンド行始動オプションを含むコネクターの開始方法についての詳細は、以下の資料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。

コネクターの停止

コネクターの停止方法は、以下のように、コネクターを始動した方法により異なります。

- コネクターの始動スクリプトを使用してコネクターをコマンド行から始動した場合:
  - Windows システムでは、始動スクリプトを起動すると、そのコネクター用の個別の「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、コネクターが停止します。
  - UNIX 系システムではコネクターはバックグラウンドで稼働するため、個別のウィンドウはありません。その代わりに、以下のコマンドを実行してコネクターを停止します。

    ```
    connector_manager_connName -stop
    ```

    ここで、*connName* はコネクター名です。
- System Manager を始動したときに起動された Adapter Monitor から起動した場合 (WebSphere Business Integration Adapters 製品のみ):

  このツールを使用して、コネクターをロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除できます。
- System Monitor から起動した場合 (WebSphere InterChange Server 製品のみ):

  このツールを使用して、コネクターをロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除できます。
- Windows システムでは、コネクターを Windows のサービスとして始動するように構成できます。この場合、Windows システムがシャットダウンされるとコネクターは停止します。

第 3 章 WebSphere MQ Workflow アプリケーションの変更

WebSphere MQ Workflow アプリケーションでは、プロセス内で作業を実行するためのステップとして、ノードが使用されます。各ノードは、1 つの役割 (例えば、部門長) に対応する特定のアクティビティー (例えば、オーダー承認) に関連付けられています。

ノードは、他のノードまたは WebSphere MQ Workflow にとって外部にあるアプリケーションに対して、要求と応答を発行できます。ノードが Connector for WebSphere MQ Workflow と通信できるようにするには、最初に User-Defined Program Execution Server (UPES) を指定する必要があります。

本章では、Connector for WebSphere MQ Workflow と対話できる User-Defined Program Execution Server (UPES) を作成する方法について説明します。本章の内容は次のとおりです。

- 『UPES の構成』

UPES の構成

本章では、User-defined Program Execution Server (UPES) を定義および構成する方法について説明します。UPES を使用すると、ワークフロー・ノードは Connector for WebSphere MQ Workflow に要求を発行できます。

注: UPES を構成する前に、システム上に WebSphere MQ Workflow Buildtime 環境をインストールする必要があります。

1. MQ Workflow Buildtime アプリケーションを始動し、「Network」タブをクリックします。

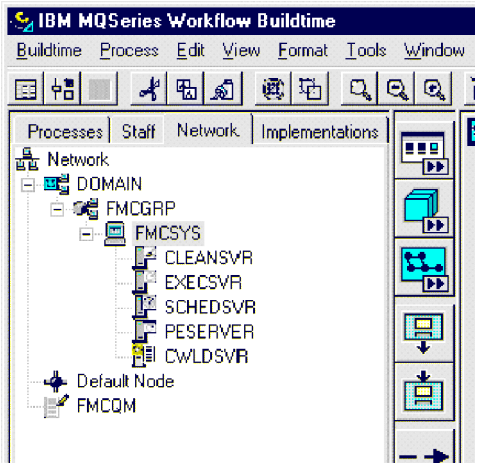

図 12. WebSphere MQ Workflow Buildtime:「Network」ビュー

2. メニュー・バーで「システム」->「New User-Defined Program Execution Server」を選択します。

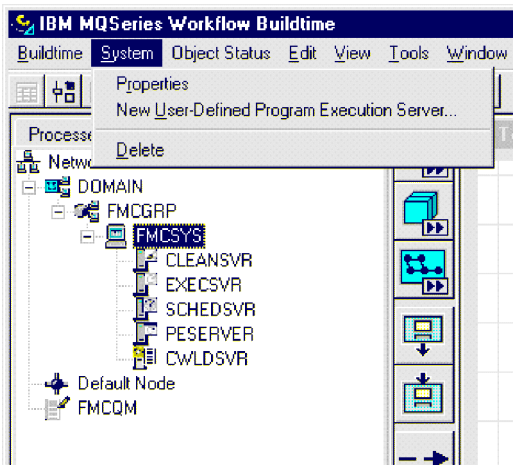

図 13. WebSphere MQ Workflow Buildtime: 新規の UPES の選択

3. ダイアログ・ボックスで UPES の固有の名前 (例えば、CWLDSVR) を入力します。

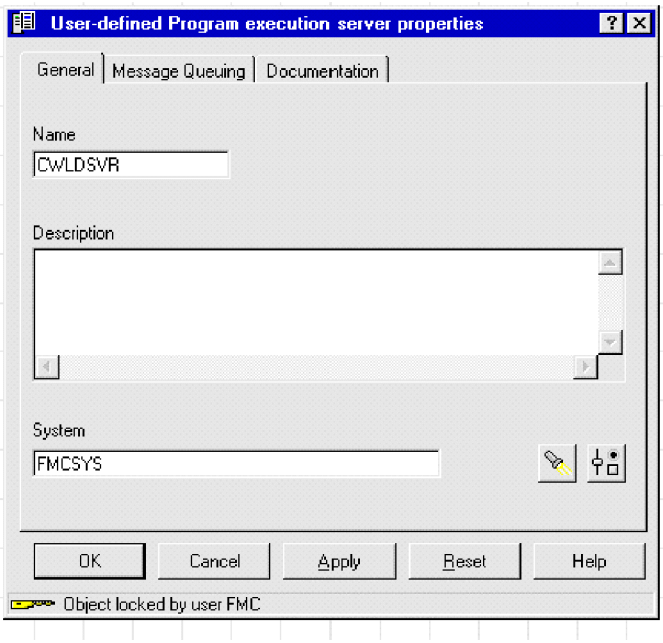

図 14. WebSphere MQ Workflow Buildtime: 新規の UPES の命名

4. 「Message Queuing」タブをクリックし、コネクターの入力キューとキュー・マネージャーの名前を入力します。

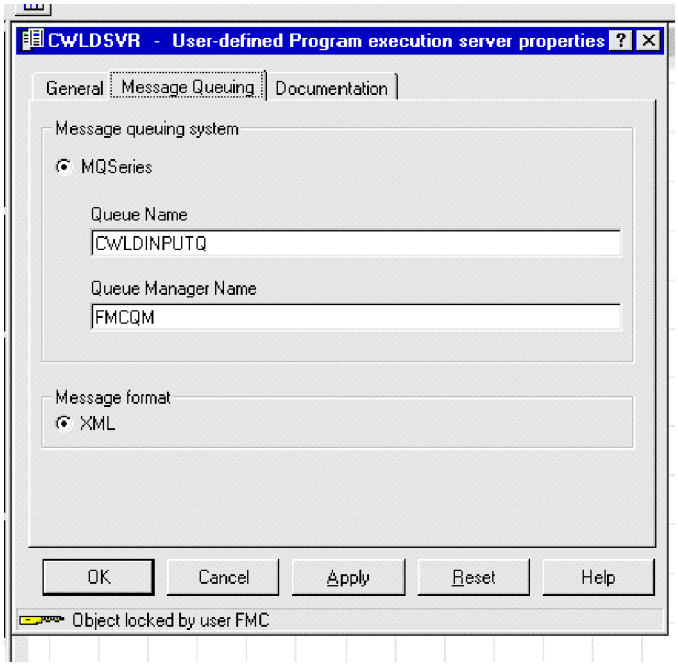

図 15. WebSphere MQ Workflow Buildtime: メッセージ・キューの構成

5. 「Implementations」タブをクリックします。

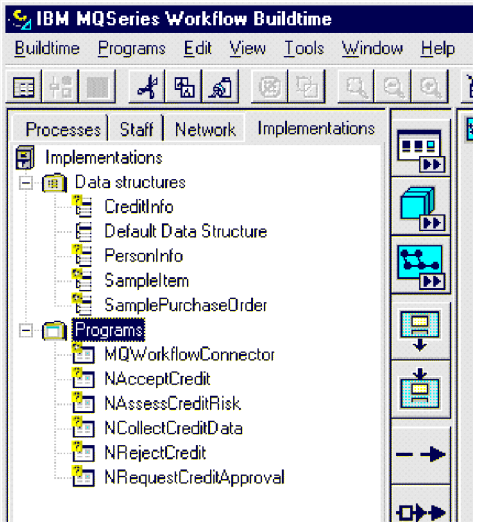

図 16. WebSphere MQ Workflow Buildtime:「Implementations」ビュー

6. メニュー・バーから、「Programs」->「New Program」を選択します。

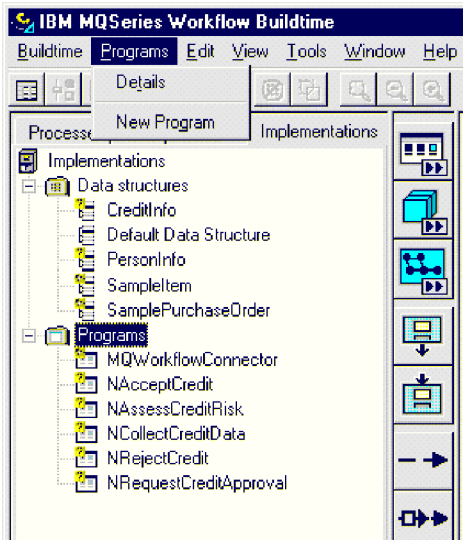

図 17. WebSphere MQ Workflow Buildtime: 新規のプログラムの選択

7. プログラムの名前を指定します。ノード対コラボレーションの個々の関係ごとに 1 つの UPES プログラムを定義する必要があるため、コラボレーションとして同じ名前を使用すると便利です。

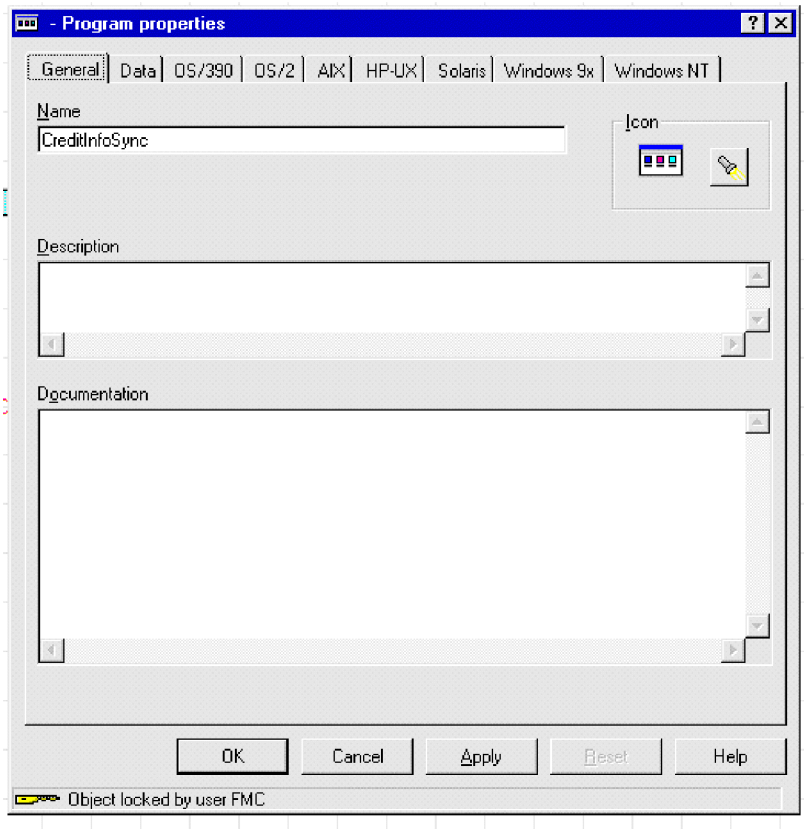

図 18. WebSphere MQ Workflow Buildtime: 新規のプログラムの命名

8. 「データ」タブをクリックし、プログラムまたはコラボレーションが受け入れるデータ構造体を指定します。「Program can run unattended」ボックスにチェックマークが付いていることを確認します。

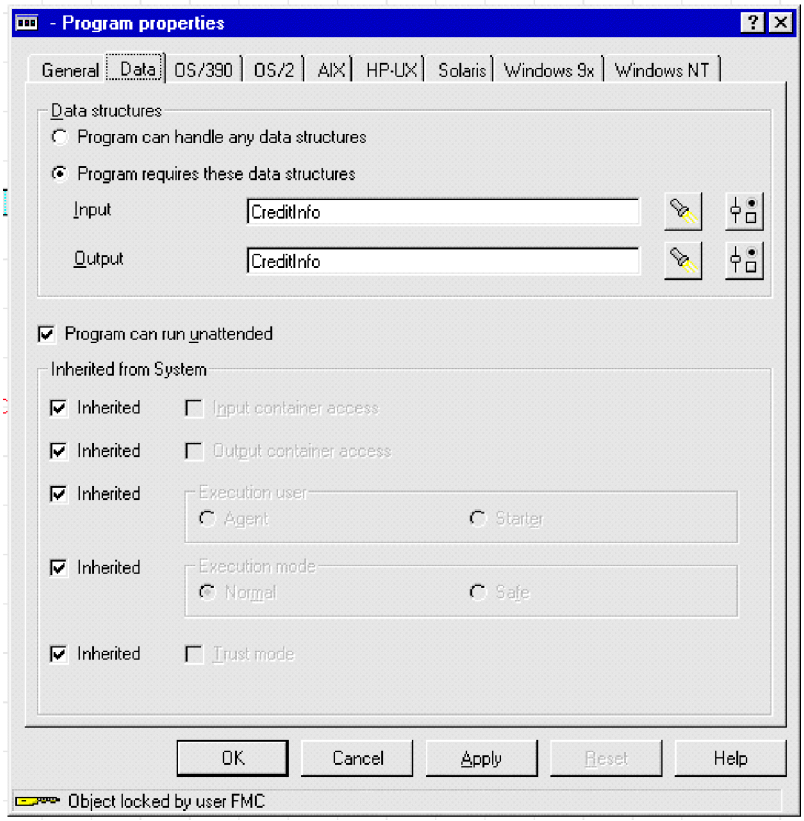

図 19. WebSphere MQ Workflow Buildtime: データ構造体の指定

9. 「Windows」タブをクリックし、実行する既存のプログラムを入力します。

   注: 指定するプログラムは実行されませんが、WebSphere MQ Workflow はこの定義を必要とします。

ICS にデータ構造体を送付するときにどの動詞とコラボレーションを使用するかを指示するために、ワークフロー設計時に 2 つのコマンド行パラメーターを指定する必要があります。これらのパラメーターは、名前と値の形式に従って指定し、名前と値のペアが複数ある場合はセミコロンで区切る必要があります。現在のところ、`verb` と *`collab`* の 2 つの値を指定できます。例えば、ワークフロー・データ構造体をコネクターに発行してから、コラボレーション `CreditInfoSync` 内で `Update` 動詞を使用して処理するには、プログラム・パラメーターを `verb=Update; collab=CreditInfoSync` に設定する必要があります。コラボレーション名が指定されていない場合 (`verb=Update` プログラム・パラメーター)、データ構造体はサブスクライブしているすべてのコラボレーションに通知されます。

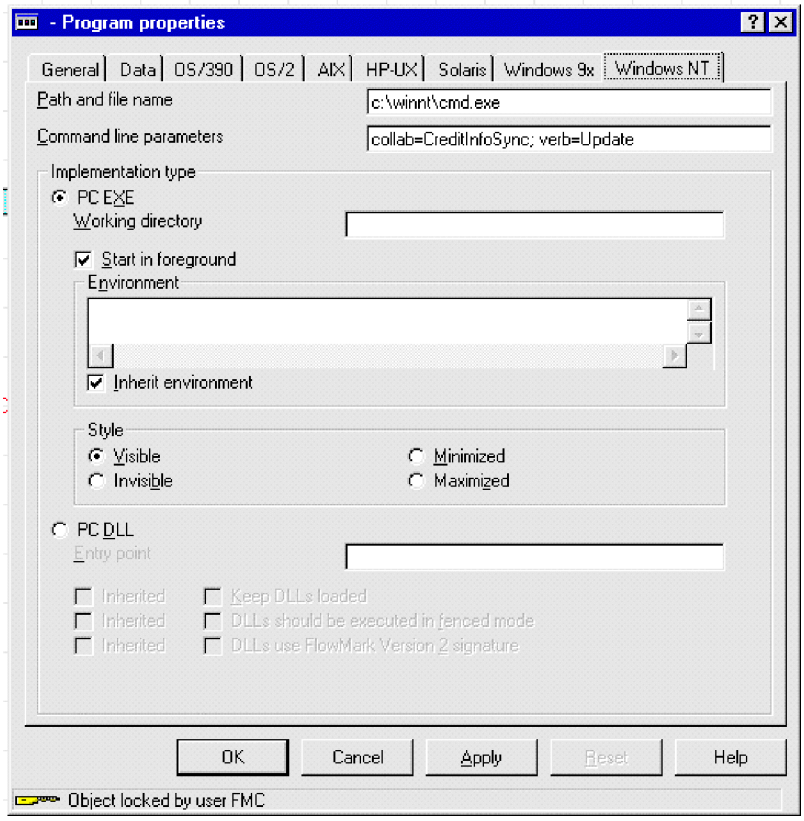

図 20. WebSphere MQ Workflow Buildtime: コマンド行パラメーターの指定

10. プログラム・ノードが WebSphere MQ Workflow コネクターに要求を発行するようにするには、新規のプログラム・ノードを作成し、(ステップ 7 で定義したように) プログラムの名前を指定します。

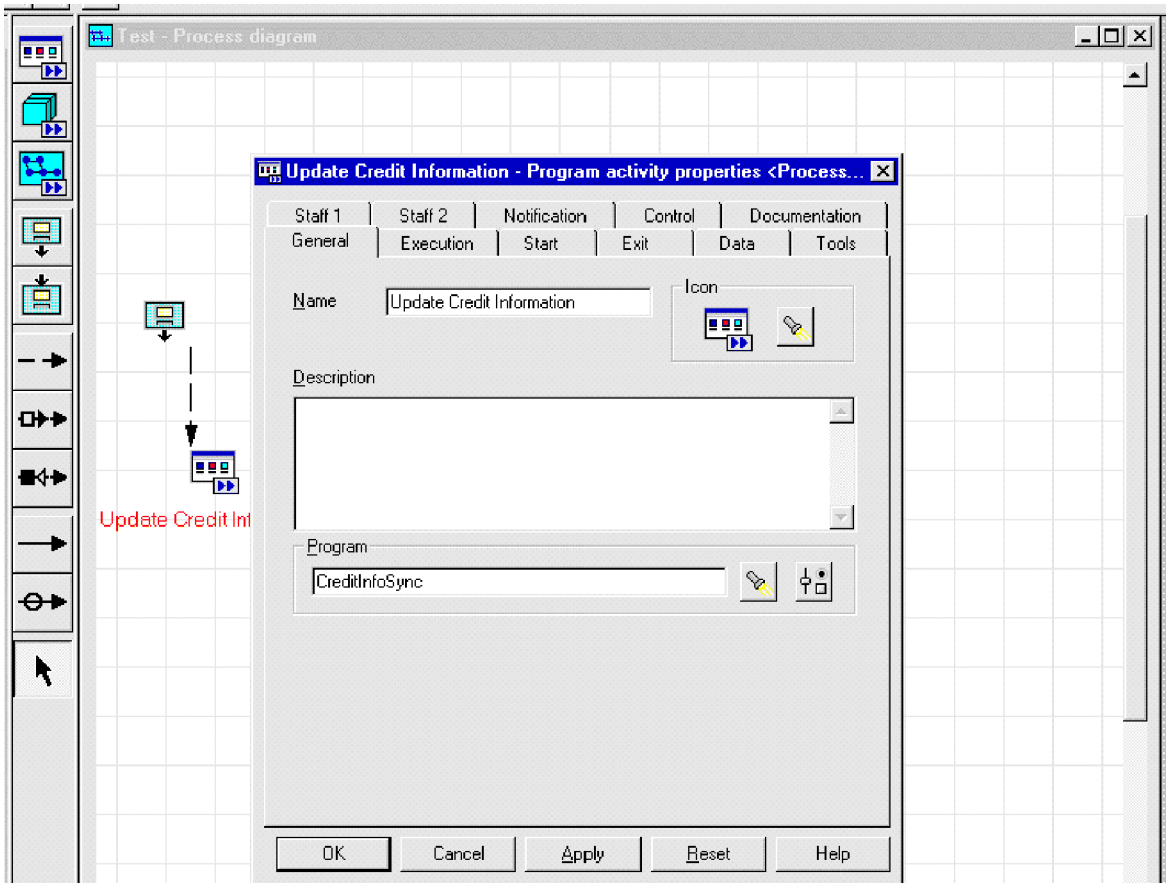

図 21. WebSphere MQ Workflow Buildtime: 新規のプログラム・ノードの作成

11. プログラム実行サーバー (CWLDSVR.FMCSYS.FMCGRP) を定義し、要求のタイプとして同期または非同期を選択します。

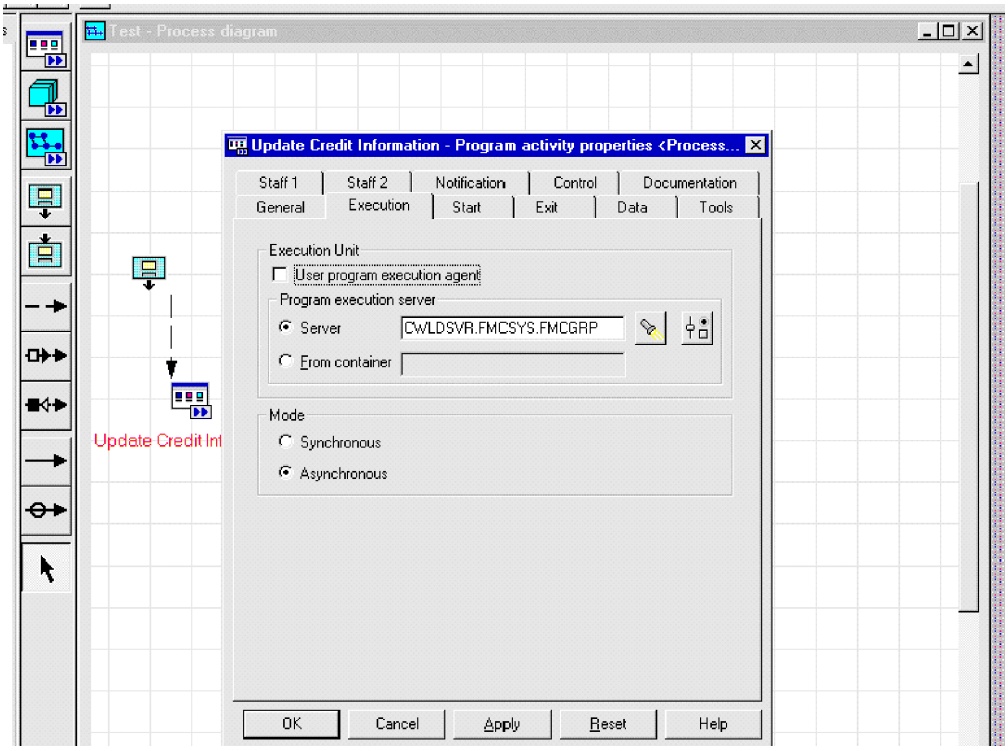

図 22. WebSphere MQ Workflow Buildtime: プログラム・サーバーの定義

WebSphere MQ Workflow 内で処理がこのノードに到達すると、ワークフロー・データ構造体を格納するメッセージがコネクターに発行されます。コネクターはビジネス内容を処理し、変更またはエラーがあれば、内容とともに戻します。

第 4 章 ビジネス・オブジェクトの開発

Connector for WebSphere MQ Workflow のインストールと構成が完了したら、ビジネス・オブジェクトを作成する必要があります。そのためには、以下の作業を実行します。

- 本書に付属しているサンプル・ビジネス・オブジェクトを変更します。
- WebSphere MQ Workflow エクスポート・ファイルからビジネス・オブジェクト定義ファイルを抽出します。

本章では、新規のビジネス・オブジェクトを実装するための手引きとして役立つ情報を提供します。また、ユーザーが変更できるサンプル・ビジネス・オブジェクトも示します。WebSphere MQ Workflow からビジネス・オブジェクト定義ファイルを抽出するには、75 ページの『第 5 章 FDLBORGEN ユーティリティーを使用したビジネス・オブジェクト定義の作成』を参照してください。

コネクターのビジネス・オブジェクトの構造

コネクターは、メタデータ主導型コネクターです。ビジネス・オブジェクトでは、メタデータとはアプリケーションに関するデータであり、このデータはビジネス・オブジェクト定義に保管されて、コネクターとアプリケーションとの対話を支援します。メタデータ主導型コネクターは、コネクターにハードコーディングされている命令ではなく、ビジネス・オブジェクト定義にエンコード されているメタデータに基づいて、サポートする各ビジネス・オブジェクトを処理します。

ビジネス・オブジェクトのメタデータには、ビジネス・オブジェクトの構造、その属性プロパティーの設定、およびアプリケーション固有情報の内容が含まれます。コネクターはメタデータ主導型のため、コネクターのコーディングを変更しなくても、新規ビジネス・オブジェクトや変更されたビジネス・オブジェクトを処理できます。ただし、構成されたコネクターのデータ・ハンドラーは、そのビジネス・オブジェクトの構造について想定を行います。したがって、WebSphere MQ Workflow 向けビジネス・オブジェクトを作成または変更する場合は、コネクターがそれに従うように設計されている規則に準拠して変更を行う必要があります。そうしないと、コネクターは新規のまたは変更されたビジネス・オブジェクトを適切に処理できません。

ビジネス・オブジェクトの構造に関しては、XML データ・ハンドラーが規定する要件以外にも要件があります。詳細については、37 ページの『メタオブジェクトの構成』を参照してください。コネクターが処理するビジネス・オブジェクトには、InterChange Server で許容される任意の名前を付けることができます。

コネクターはキューからメッセージを検索し、ビジネス・オブジェクト (トップレベルのビジネス・オブジェクトとメタデータによって定義されたもの) にメッセージの内容を取り込もうとします。厳密にいうと、コネクターがビジネス・オブジェクト構造を制御したり、この構造に影響を及ぼしたりすることはありません。ビジネス・オブジェクト構造を規定する機能を果たすのは、メタオブジェクト定義とコネクターのデータ・ハンドラーの要件です。ビジネス・オブジェクトを検索して渡すときのコネクターの主要な役割は、メッセージからビジネス・オブジェクトへの (およびその逆方向の) プロセスでエラーをモニターすることです。

ビジネス・オブジェクト定義のサンプル

ここでは、ビジネス・オブジェクト定義のサンプルを示します。Cardinality や IsKey などのビジネス・オブジェクト属性に関する個別情報については、「*Java コネクター開発ガイド*」を参照してください。

```
[BusinessObjectDefinition]
Name = MQWF_SampleItem
Version = 1.0.0
AppSpecificInfo = cw_mo_wfcontainer=ContainerInfo

[Attribute]
Name = Input_Item
Type = MQWF_Structure_SampleItem
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = SampleItem;type=pcdata;
IsRequiredServerBound = false
[End]
[Attribute]
Name = ContainerInfo
Type = MO_MQWorkflow_ContainerInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Output_Item
Type = MQWF_Structure_SampleItem
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = SampleItem;type=pcdata;
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
```

```
IsKey = true
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 = MQWF_Structure_SampleItem
Version = 1.0.0
AppSpecificInfo = SampleItem

[Attribute]
Name = Name
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = Name;type=pcdata;
DefaultValue =
[End]

[Attribute]
Name = Price
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = Price;type=pcdata;
DefaultValue =
[End]

[Attribute]
Name = Stock
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = Stock;type=pcdata;
DefaultValue =
[End]

[Attribute]
Name = ObjectEventId
Type = String
```

```
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
DefaultValue =
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]

[Verb]
Name = Delete
[End]
[End]

[BusinessObjectDefinition]
Name = MO_MQWorkflow_ProcessInfo
Version = 1.0.0

[Attribute]
Name = Role
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Organization
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ProcessAdministrator
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration
Type = String
Cardinality = 1
MaxLength = 255
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]

[BusinessObjectDefinition]
Name = MO_MQWorkflow_ActivityInfo
Version = 1.0.0

[Attribute]
Name = Priority
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = MembersOfRoles
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = CoordinatorOfRole
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Organization
Type = String
```

```
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = OrganizationType
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = LowerLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = UpperLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = People
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PersonToNotify
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = Duration2
Type = String
Cardinality = 1
MaxLength = 255
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]

[BusinessObjectDefinition]
Name = MO_MQWorkflow_ContainerInfo
Version = 1.0.0

[Attribute]
Name = PROCESS_INFO
Type = MO_MQWorkflow_ProcessInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ACTIVITY_INFO
Type = MO_MQWorkflow_ActivityInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ACTIVITY
```

```
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PROCESS
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PROCESS_MODEL
Type = String
Cardinality = 1
MaxLength = 255
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]
```

エラー処理

コネクターによって生成されたすべてのエラー・メッセージは、メッセージ・ファイルに格納されます。(ファイルの名前は、LogFileName 標準コネクター構成プロパティーによって決定されます。) 各エラーには、エラー番号とそれに続くエラー・メッセージが含まれます。

```
Message number
Message text
```

コネクターは、個別のエラーを以下の各セクションで説明するように処理します。

アプリケーション・タイムアウト

以下の場合に、エラー・メッセージ「BON_APPRESPONSETIMEOUT」が戻されます。

- コネクターがメッセージの検索中に WebSphere MQ Workflow キュー・マネージャーへの接続を確立できない。
- コネクターはビジネス・オブジェクトを正常にメッセージに変換したが、接続が切断されたためにメッセージを出力キューにデリバリーできない。
- コネクターはメッセージを発行したが、変換プロパティー `TimeoutFatal` が `true` に設定されたビジネス・オブジェクトの応答を待機していてタイムアウトになった。
- コネクターが同期 API を使用して WebSphere MQ Workflow と通信しているときにタイムアウトになった。

アンサブスクライブされたメッセージ

コネクターは、以下の場合に、`UnsubscribedQueue` プロパティーで指定されたキューにメッセージをデリバリーします。

- コネクターが MQSTR または FMXML 以外の形式のメッセージを検索する。
- コネクターによって `WfMessage` が検索されたが、メッセージ名が `ActivityImplInvoke` タイプまたは `General Error` タイプのものではない。
- コネクターがサブスクリプション・デリバリーを処理するときにデータ構造体のトップレベルのビジネス・オブジェクトを検出できない。

注: `UnsubscribedQueue` が定義されていない場合、アンサブスクライブされたメッセージは破棄されます。

XML 構造エラー

コネクターは、以下の場合に、`ErrorQueue` プロパティーで指定されたキューにメッセージをデリバリーします。

- XML 文書が正しい形式ではない。
- `WfMessage` の構造が完全ではない。

データ・ハンドラーによる変換

コネクターは、以下の場合に、`ErrorQueue` プロパティーで指定されたキューにメッセージをデリバリーします。

- データ・ハンドラーがビジネス・オブジェクトを XML に (またはその逆方向に) 変換できない。

エラーと WfMessage 文書

コネクターは、コネクターが報告した標準エラーに応答するだけでなく WebSphere MQ Workflow 自体が発行したエラーにも応答します。コネクターが発行したメッセージをコネクターが同期処理しているときにエラーが発生した場合、WebSphere MQ Workflow は、Exception 要素を格納する WfMessage を戻します。この要素のテキ

ストは、WebSphere MQ Workflow アプリケーションに対する要求操作が失敗したときにコネクターが InterChange Server に戻すエラー・メッセージを指定します。

WebSphere MQ Workflow が発行した WfMessage の同期処理にコネクターが (上記のなんらかの理由により) 失敗した場合、コネクターは、Exception 要素を格納する応答 WfMessage を WebSphere MQ Workflow に送信しようとします。コネクターは、失敗の原因であるイベントについての詳細な説明をこの要素に取り込みます。WebSphere MQ Workflow は、このメッセージを使用して失敗したワークフローの状態を更新することにより、適切なユーザー介入を実行できるようにします。

コネクターは、WebSphere MQ Workflow サーバーが最初に発行した要求への応答を発行した後には、確認通知を待機しません。WebSphere MQ Workflow はこのような場合にのみエラーを報告するため、コネクターが確認通知を受信する手段はありません。コネクターは、この問題による不都合が生じないようにするために、応答の結果であるすべてのエラー・メッセージがコネクターの入力キューに発行されるように応答メッセージ内で指定します。これにより、エラーが発生した場合、コネクターは最終的に `pollForEvents` の間に通知を受信します。エラー・ログが記録されますが、それ以上のアクションは実行されません。このようなエラーは、応答ビジネス・オブジェクトが生成したデータ構造体が不完全または不正であると WebSphere MQ Workflow が判断した結果だと考えられます。

トレース

トレースは、コネクターの動作を詳細に追跡するために使用できるオプショナル・デバッグ機能です。トレース・メッセージは、デフォルトでは STDOUT に書き込まれます。トレース・メッセージの構成の詳細については、57 ページの『第 3 章 WebSphere MQ Workflow アプリケーションの変更』に記載されている『コネクター構成プロパティー』を参照してください。トレースを使用可能にして設定する方法などのトレースの詳細については、「*コネクター開発ガイド*」を参照してください。

コネクター・トレース・メッセージでは、以下のレベル・オプションを使用できます。

レベル 0　コネクターは、そのバージョンのみをリストします。

レベル 1　コネクターは、ポーリング中にメッセージが検索されたり要求処理中にメッセージが通知されたりするごとに、ログを記録します。

レベル 2　コネクターは、`gotApplEvent()` または `executeCollaboration()` から InterChange Server にビジネス・オブジェクトが送付されるたびにログを記録します。

レベル 3　コネクターは、このプロセスで解析されているメッセージおよび実行されたロジックについてのより詳細な情報を提供します。

レベル 4　コネクターは、コネクターが開始または終了する各メソッドをトレースします。

レベル 5　コネクターは、コネクターの初期化ログを記録し、処理されたすべてのビジネス・オブジェクトをダンプし、読み取り/書き込みが行われているすべての XML 文書をリストします。

第 5 章 FDLBORGEN ユーティリティーを使用したビジネス・オブジェクト定義の作成

本章では、WebSphere MQ Workflow エクスポート・ファイルからのビジネス・オブジェクト定義ファイルの抽出を自動化するスタンドアロン・コマンド行ユーティリティーである FDLBORGEN について説明します。抽出されたビジネス・オブジェクト定義ファイルは、XML データ・ハンドラーと互換性があります。

FDL (WebSphere MQ Workflow Definition Language の頭字語) は、FDLBORGEN が入力として使用する WebSphere MQ Workflow エクスポート・ファイルのサフィックスです。

FDLBORGEN ユーティリティーについて

FDLBORGEN ユーティリティーは、以下の操作を実行します。

- FDL 構造定義に基づいてビジネス・オブジェクト定義を作成し、そのビジネス・オブジェクト定義をビジネス・オブジェクト定義ファイルに書き込みます。このファイルは、`repos_copy` ユーティリティーを使用して InterChange Server リポジトリーにロードできます。
- XML データ・ハンドラーの要件に準拠するビジネス・オブジェクト定義を作成します。これらのビジネス・オブジェクト定義は、編集せずにそのまま使用できます。
- 必須の `ObjectEventId` 属性をすべてのビジネス・オブジェクト定義に追加します。また、リポジトリー・バージョン番号を指定する場合には、このバージョン番号をビジネス・オブジェクト・ファイルの先頭に追加します。

FDLBORGEN は、`connectors¥MQWorkflow¥utilities¥fdlborgen` ディレクトリーの一部としてインストールされる次の 2 つのファイルから構成されます (製品の完全なインストール・ディレクトリーについては、21 ページの『インストール済みファイルの構造』を参照)。

- FDLBORGEN ユーティリティーを起動する実行可能ファイル
  - Windows 環境では、この実行可能ファイルは `fdlborgen.bat` です。
  - UNIX 環境では、この実行可能ファイルは `fdlborgen` です。
- FDLBORGEN の操作に必要なすべての Java クラスを格納する Java アーカイブ・ファイル `fdlborgen.jar`

FDLBORGEN ユーティリティーを使用するための前提条件

FDLBORGEN ユーティリティーを使用する前に、1 つ以上の入力ファイルを作成する必要があります。そのためには、WebSphere MQ Workflow で .fdl タイプのファイルを生成し、そのファイルをエクスポートします。

1. WebSphere MQ Workflow Buildtime を開きます。
2. 「Buildtime」メニューから「エクスポート」を選択します。
3. ダイアログ・ボックスでファイル名 (例えば、MyExport.fdl) を選択し、「OK」をクリックします。

FDLBORGEN の構文

FDLBORGEN コマンド行ユーティリティーを実行するには、以下の手順を実行します。

1. DOS コマンド・プロンプト・ウィンドウ (Windows) またはシェル・ウィンドウ (UNIX) を開きます。
2. FDLBORGEN のコマンドを入力します。

FDLBORGEN コマンドの形式は、以下のように定義されます。

```
fdlborgen -i[input-file] -o[output-file] -p[prefix]
 {-n[object-name]} {-r[repos_version]} {-v[verblist]} {-V}
```

は、次のように説明されます。

-i[*input-file*]
.fdl ファイルの名前とパスを指定します (fdl ファイルが現行ディレクトリーにない場合に使用します)。

-o[*output-file*]
生成されたビジネス・オブジェクト定義の格納先の名前と場所を指定します。

-n[*object-name*]
ビジネス・オブジェクト定義に変換される fdl ファイルに含まれるトップレベルのデータ構造体の名前を指定します。

-p[*prefix*]
生成されたビジネス・オブジェクト定義の名前の前に挿入する接頭部を指定します。これは、fdl オブジェクト名とビジネス・オブジェクト名を区別するときに役立ちます。prefix オプションは必須です。

-r[*repos_version*]
生成されたビジネス・オブジェクト定義ファイルの先頭に ReposCopy ヘッダーを追加します。例えば、-r1.0.2 と指定すると、以下の行がファイルの先頭に追加されます。

```
[ReposCopy]
Version = 1.0.2
[End]
```

ビジネス・オブジェクト定義ファイルの既存のバージョンを上書きする場合は、このパラメーターを使用してバージョン情報を保持します。

-v[*verblist*]
各ビジネス・オブジェクトに組み込む動詞のリスト

を指定します。Create、Retrieve、Update、および Delete の各動詞がサポートされます。各動詞をコンマで区切り、スペースは入力しないでください。

このパラメーターを指定しない場合は、標準の Create、Retrieve、Update、および Retrieve の各動詞が追加されます。

`-V` プログラムを冗長モードに切り替え、検出されたすべてのエントリー、属性、要素、およびコメントを出力します。

`-[mapping-file]` マッピング・ファイルの名前を指定します (ファイルが現行ディレクトリーにない場合はパスも指定します)。マッピング・ファイルについては、以下の『変換についての注意事項』を参照してください。

注: FDLBORGEN から検出されたすべてのエラーは、`stderr` に送信されます。

FDLBORGEN の使用例

WebSphere MQ Workflow データ構造体 `MyCustomer` をビジネス・オブジェクト `Wf_MyCustomer` に変換するには、最初に WebSphere MQ Workflow Buildtime 内で入力ファイルを生成する必要があります。現行ディレクトリー内の `MyExport.fdl` を使用して、FDLBORGEN を以下のように実行できます。

```
fdlborgen -iMyExport.FDL -oMyBO.txt -nMyCustomer -pWf_ -vCreate,Update -r2.0.0
```

これにより、FDL ファイル `MyExport.FDL` に含まれるデータ構造体 `MyCustomer` に基づいて、ビジネス・オブジェクト `Wf_MyCustomer` が生成されます。この定義ファイルには、必要な子データ構造体もすべて組み込まれます。

変換についての注意事項

データ構造体に含まれる `STRING`、`LONG`、または `FLOAT` タイプのメンバーは、いずれも同じ名前を持つ `STRING` タイプの属性としてマップされます。

データ構造体内のすべてのオブジェクト・メンバーは、同じ名前を持つ `<boprefix><data structure name>` タイプの子オブジェクトに変換されます。

ビジネス・オブジェクト名と属性名には、最大 80 文字の英数字と下線を使用できます。FDL でトップレベルのデータ構造体名またはメンバー名を命名するときは、この規則に従ってください。この規則に従わない場合は、マッピング・テーブルを作成し、-m コマンド・オプションを使用してマッピング・テーブル・ファイルを指定してください。

マッピング・テーブル・ファイルは以下の CSV 形式である必要があります。

- `<name1 in FDL>,<mapped name1>`
- `<name2 in FDL>,<mapped name2>`

-m オプションおよびマッピング・ファイル名とパス (必要な場合) を指定すると、

```
<name1 in FDL>
```

は、生成されたビジネス・オブジェクト定義ファイルで

```
<mapped name1>
```

に置換されます。

第 6 章 トラブルシューティング

本章では、Connector for WebSphere MQ Workflow の始動時および実行時に発生する可能性がある問題について説明します。

始動時の問題

問題	可能性のある解決方法/説明
コネクターが初期化中に予期せずシャットダウンし、次のメッセージが通知されました: `java.lang.NoSuchMethodError at com.ibm.workflow.api.Agent$OsaLocator.locateController (Agent.java:219) at com.ibm.workflow.api.Agent$OrbLocator.locate (Agent.java:173) at com.ibm.workflow.api.Agent.setName(Agent.java:401).`	このエラーは、デフォルトの IBM Java ORB ライブラリーと、WebSphere MQ Workflow API が IBM Java ORB エージェントを使用してリモート Workflow サーバーと通信するために必要なライブラリーとが非互換である可能性があることを示しています。IBM の要件に従って、これらのライブラリーをブートストラップにして、デフォルトの IBM Java ORB ライブラリーよりも優先されるようにする必要があります。そのためには、`start_connector.bat` (または `start_connector.sh`) を開き、「ステップ 3」で開始されるセクションまでスクロールダウンし、IBM が必要とする正しい IBM Java ORB ライブラリーが指定されていることを確認します。 **注:** WebSphere MQ Workflow API には、製品付属のライブラリーとは異なる IBM Java ORB ライブラリーが必要となることがあります。詳細については、「*IBM WebSphere MQ Workflow プログラミング・ガイド*」を参照してください。
コネクターがメッセージを処理中に終了し、次のようなエラーが報告されました: `Exception thrown: com.crossworlds.connectors.mqworkflow.exceptions. FatalProcessingException: [Type: Fatal Error ] [MsgID: 40007] [Mesg: Failed to read message content.An IO error occurred: java.io.UnsupportedEncodingException Cp437.] ]`	使用している JVM バージョンには、メッセージ文字セットをサポートするために必要なライブラリーがありません。この問題の最も簡単な解決方法は、Sun Microsystems のサイトから JDK の最新のバージョンをダウンロードし、この新規の JVM 内でコネクターを実行することです。`start_connector.bat` (または `start_connector.sh`) ファイルを開き、`%CROSSWORLDS%¥java` と記述されている箇所のすべてを新規の JVM のパスで置換します。
初期化中にコネクターが予期しないエラーでシャットダウンし、次の例外が報告されました: `java.lang.UnsatisfiedLinkError: no mqjbnd01 in shared library path`	コネクターは、IBM WebSphere MQ Java クライアント・ライブラリーから必要なランタイム・ライブラリー (`mqjbnd01.dll` [Windows] または `libmqjbnd01.so` [UNIX]) を見つけることができません。パスにライブラリー・フォルダーが組み込まれていることを確認します。
初期化中にコネクターが予期しないエラーでシャットダウンし、次のメッセージが報告されました: `Exception in thread "main" java.lang.NoClassDefFoundError: com/ibm/workflow/api/FmcException`	`start_WebSphereMQWorkflow.bat` (Windows) または `start_WebSphereMQWorkflow.sh` (UNIX) 内で WebSphereMQWorkflow クライアント・ライブラリーの正しいパスが指定されていることを確認します。詳細については、51 ページの『始動ファイルの構成』を参照してください。

問題	可能性のある解決方法/説明
初期化中にコネクターが予期しないエラーでシャットダウンし、次の例外が報告されました: `java.lang.UnsatisfiedLinkError: no fmcojprf (libfmcojpfr.a or .so) in java.library.path`	コネクターは、必要なランタイム・ライブラリーを見つけることができません。システム上で、指定されたライブラリー (例えば `libfmcojprf.a`) を検索し、パスにこのファイルの親ディレクトリーが入っていることを確認します。ライブラリーが見つからない場合は、前提条件となっている必要なソフトウェア (WebSphere MQ Workflow アプリケーションおよび WebSphere MQ クライアント・ライブラリー) がすべてインストールされていることを確認してください。ライブラリーが見つかり、パスに追加した場合、`start_connector` スクリプトを変更し、続いてこのパスを、アダプターを始動するコマンド行に渡される `java.library.path` オプションに追加することが必要になる場合があります。これを行うためには、始動スクリプトの末尾までスクロールして、アダプターを始動するコマンドを探します (このコマンドは *ProductDir*¥bin¥java などで始まります)。コマンド行には、オプションが指定されています。`_Djava.library.path=` で始まるオプションを見つけます。当該ライブラリーの親ディレクトリーを、`_Djava.library.path` オプションに指定されている他のディレクトリーのリストの後に追加します。例えば、`c:program¥files¥webspheremq workflow¥bin` に `libfmcojprf.a.` があったとすると、変更後のコマンド行は `_Djava.library.path=`*ProductDir*`¥bin; %CONNDIR%;c:¥program files¥webspheremqworkflow¥bin` となります。

イベント処理

問題	可能性のある解決方法/説明
コネクターが、コネクター構成の指定内容とは無関係に、すべてのメッセージを「fmc」のユーザー ID 付きで送信します。	コネクターのために指定された WebSphere MQ サーバー・チャネルのプロパティーを調べます。MCA User ID プロパティーに値が指定されていないことを確認します。

付録 A. コネクターの標準構成プロパティー

この付録では、WebSphere Business Integration アダプターのコネクター・コンポーネントの標準構成プロパティーについて説明しています。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ー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、そのブローカーで稼働するアダプターのために構成が必要な標準プロパティーのリストが表示されます。

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

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

新規プロパティーと削除されたプロパティー

本リリースには、次の標準プロパティーが追加されました。

新規プロパティー

- XMLNameSpaceFormat

削除されたプロパティー

- RestartCount
- RHF2MessageDomain

標準コネクター・プロパティーの構成

アダプター・コネクターには 2 種類の構成プロパティーがあります。

- 標準構成プロパティー
- コネクター固有の構成プロパティー

このセクションでは、標準構成プロパティーについて説明します。コネクター固有の構成プロパティーの詳細は、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 からコネクター・プロパティーを構成します。Connector Configurator には、System Manager からアクセスします。Connector Configurator の使用方法の詳細は、Connector Configurator に関する付録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。コネクターが UNIX システム上で動作している場合は、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ーのコネクター・プロパティーを設定する場合は、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、コネクター用の Connector Configurator を開く必要があります。

プロパティー値の設定と更新

プロパティーのフィールド長のデフォルトは 255 文字です。

コネクターは、以下の順序に従ってプロパティーの値を決定します (最も番号の大きい項目は、他の項目の値をオーバーライドします)。

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

コネクターは、始動時に構成値を取得します。実行時セッション中に 1 つ以上のコネクター・プロパティーの値を変更する場合は、プロパティーの**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。標準コネクター・プロパティーには、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。コネクターがスタンドアロン・モードで (System Manager から独立して) 作動している場合 (例えば、WebSphere Message Brokers の 1 つで作動している場合)、構成ファイルによるプロパティーの変更のみが可能です。この場合、動的更新は実行できません。
- コンポーネント再始動
  System Manager でコネクターを停止してから再始動しなければ、変更が有効になりません。アプリケーション固有コンポーネントまたは統合ブローカーを停止、再始動する必要はありません。
- サーバー再始動
  アプリケーション固有のコンポーネントおよび統合ブローカーを停止して再始動しなければ、変更が有効になりません。
- エージェント再始動 (ICS のみ)
  アプリケーション固有のコンポーネントを停止して再始動しなければ、変更が有効になりません。

特定のプロパティー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、または次に示すプロパティーの要約の表の**「更新メソッド」**列を参照してください。

標準プロパティーの要約

表 13 は、標準コネクター構成プロパティーの早見表です。すべてのコネクターがこれらのすべてのプロパティーを使用しているわけではありません。標準プロパティーの依存関係は RepositoryDirectory に基づいているため、プロパティーの設定は統合ブローカーごとに異なります。

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

表 13. 標準構成プロパティーの要約

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

表 13. 標準構成プロパティーの要約 (続き)

プロパティー名	指定可能な値	デフォルト値	更新メソッド	注
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 localhost:2506 を使用。	crossworlds.queue.manager	コンポーネント再始動	JMS トランスポートのみ
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	JMS トランスポートのみ
jms.Password	任意の有効なパスワード		コンポーネント再始動	JMS トランスポートのみ
jms.UserName	任意の有効な名前		コンポーネント再始動	JMS トランスポートのみ
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	Repository Directory は <REMOTE>
ListenerConcurrency	1 から 100	1	コンポーネント再始動	Delivery Transport は MQ でなければならない
Locale	en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR **注:** これは、サポートされるロケールの一部です。	en_US	コンポーネント再始動	
LogAtInterchangeEnd	True または False	False	コンポーネント再始動	Repository Directory は <REMOTE> でなければならない
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE> でなければならない
MessageFileName	パスまたはファイル名	InterchangeSystem.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	*CONNECTORNAME*/MONITORQUEUE	コンポーネント再始動	JMS トランスポートのみ: DuplicateEvent Elimination は True でなければならない

表 13. 標準構成プロパティーの要約 (続き)

プロパティー名	指定可能な値	デフォルト値	更新メソッド	注
OADAutoRestartAgent	True または False	False	動的	Repository Directory は <REMOTE> でなければならない
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE> でなければならない
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE> でなければならない
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 (コネクターのコマンド・プロンプト・ウィンドウで文字 p が入力された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 が指定されている
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・リポジトリーの場所		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: C:¥crossworlds¥ リポジトリーに設定
RequestQueue	有効な JMS キュー名	*CONNECTORNAME*/REQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
ResponseQueue	有効な JMS キュー名	*CONNECTORNAME*/RESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS: Repository Directory が <REMOTE> のときのみ必要
RestartRetryCount	0 から 99	3	動的	
RestartRetryInterval	適切な正数 (単位: 分): 1 から 2147483547	1	動的	
SourceQueue	有効な WebSphere MQ 名	*CONNECTORNAME*/SOURCEQUEUE	エージェント再始動	Delivery Transport が JMS であり、かつ Container Managed Events が指定されている場合のみ
SynchronousRequestQueue		*CONNECTORNAME*/SYNCHRONOUSREQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS

表 13. 標準構成プロパティーの要約 (続き)

プロパティー名	指定可能な値	デフォルト値	更新メソッド	注
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
WsifSynchronousRequest Timeout	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ー

このセクションでは、各標準コネクター構成プロパティーの定義を示します。

AdminInQueue

統合ブローカーからコネクタ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/ADMININQUEUE です。

AdminOutQueue

コネクター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> の場合のみ適用できます。

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

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

AgentTraceLevel

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

ApplicationName

コネクターのアプリケーションを一意的に特定する名前です。この名前は、システム管理者が WebSphere Business Integration システム環境をモニターするために使用されます。コネクターを実行する前に、このプロパティー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 ベースのコネクターでは、このプロパティーは使用しません。C++ ベースのコネクターでは、現在、このプロパティーに ascii7 という値が使用されています。

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

ConcurrentEventTriggeredFlows

RepositoryDirectory が <REMOTE> の場合のみ適用できます。

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

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

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

- Maximum number of concurrent events プロパティーの値を増加して、コラボレーションが複数のスレッドを使用できるように構成します。
- 宛先アプリケーションのアプリケーション固有コンポーネントが複数の要求を並行して実行できることを確認します。つまり、このコンポーネントがマルチスレッド化されているか、またはコネクター・エージェント並列処理を使用でき、複

数プロセスに対応するよう構成されている必要があります。Parallel Process Degree 構成プロパティーに、1 より大きい値を設定します。

ConcurrentEventTriggeredFlows プロパティーは、順次に実行される単一スレッド処理であるコネクターのポーリングでは無効です。

ContainerManagedEvents

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

デフォルト値は No value です。

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

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

また、MimeType、DHClass、および DataHandlerConfigMOName (オプショナル) プロパティーを設定したデータ・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ーの値を設定するには、Connector Configurator の**「データ・ハンドラー」**タブを使用します。「データ・ハンドラー」タブの値のフィールドは、ContainerManagedEvents を JMS に設定した場合にのみ表示されます。

注: ContainerManagedEvents を JMS に設定した場合、コネクターはその 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。

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

ControllerStoreAndForwardMode

RepositoryDirectory が <REMOTE> の場合のみ適用できます。

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

このプロパティーを 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、コネクター・コントローラーはそのアプリケーション固有のコンポーネントへの要求をブロックします。アプリケーション固有のコンポーネントが作動可能になると、コネクター・コントローラーはアプリケーション固有のコンポーネントにその要求を転送します。

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

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

デフォルト値は true です。

ControllerTraceLevel

RepositoryDirectory が <REMOTE> の場合のみ適用できます。

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

DeliveryQueue

DeliveryTransport が JMS の場合のみ適用できます。

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

デフォルト値は CONNECTORNAME/DELIVERYQUEUE です。

DeliveryTransport

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

- ICS がブローカー・タイプの場合は、DeliveryTransport プロパティーの指定可能な値は 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、指定可能な値は JMS のみです。

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

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ー・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ー・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、コネクターのポーリング・スレッドは、イベントを選出した後、コネクター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、コネクターのポーリング・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ー・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、コネクターとクライアント・コネクター・フレームワークとの間の通信を可能にします。

JMS をデリバリー・トランスポートとして選択した場合は、`jms.MessageBrokerName`、`jms.FactoryClassName`、`jms.Password`、`jms.UserName` などの追加の JMS プロパティーが 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ーです。

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

- 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> の場合のみ適用できます。

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> の場合のみ適用できます。

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)。このプロパティーは、RepositoryDirectory の値が <REMOTE> の場合のみ適用できます。

デフォルト値は 1M です。

jms.FactoryClassName

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

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

jms.MessageBrokerName

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

デフォルト値は crossworlds.queue.manager です。

jms.NumConcurrentRequests

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

デフォルト値は 10 です。

jms.Password

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

デフォルトはありません。

ListenerConcurrency

このプロパティーは、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。このプロパティーにより、データベースへの複数イベントの書き込み操作をバッチ処理できるので、システム・パフォーマンスが向上します。デフォルト値は 1 です。

このプロパティーは、MQ トランスポートを使用するコネクターにのみ適用されます。`DeliveryTransport` プロパティーには `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。このプロパティーの値は、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。

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

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

デフォルト値は en_US です。コネクターがグローバル化に対応していない場合、このプロパティーの有効な値は en_US のみです。特定のコネクター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ーのバージョン・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/wicserver/infocenter

LogAtInterchangeEnd

RepositoryDirectory が <REMOTE> の場合のみ適用できます。

統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。ブローカーのログ宛先にログを記録すると、電子メール通知もオンになります。これにより、エラーまたは致命的エラーが発生すると、InterchangeSystem.cfg ファイルに指定された MESSAGE_RECIPIENT に対する電子メール・メッセージが生成されます。

例えば、LogAtInterChangeEnd を true に設定した場合にコネクター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・バッファー内のイベントの最大数。このプロパティーはフロー制御が使用し、RepositoryDirectory プロパティーの値が <REMOTE> の場合のみ適用できます。

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

MessageFileName

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

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

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

MonitorQueue

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

コネクターの使用する再始動機能が自動かリモートかを指定します。この機能は、MQ によりトリガーされる Object Activation Daemon (OAD) を使用して、異常シャットダウン後のコネクターの再始動や System Monitor からのリモート・コネクターの始動を行います。

自動およびリモートの再始動機能を使用可能にするには、このプロパティーを true に設定する必要があります。MQ によりトリガーされる 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 によりトリガーされる OAD がコネクターの再始動を自動的に試行する回数の最大数を指定します。OADAutoRestartAgent プロパティーを有効にするには、値を 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 によりトリガーされる OAD の再試行間隔の分数を指定します。コネクター・エージェントがこの再試行間隔の間に再始動しないと、コネクター・コントローラーが OAD にコネクター・エージェントの再始動を再度要求します。OAD はこの再試行処理を OADMaxNumRetry プロパティーで指定されている回数だけ繰り返します。OADAutoRestartAgent プロパティーを有効にするには、値を 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。

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

PollFrequency

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

- ポーリング・アクション間のミリ秒数。

- ワード key。コネクターは、コネクターのコマンド・プロンプト・ウィンドウで文字 p が入力されたときにのみポーリングを実行します。このワードは小文字で入力します。
- ワード no。コネクターはポーリングを実行しません。このワードは小文字で入力します。

デフォルト値は 10000 です。

重要: 一部のコネクターでは、このプロパティーの使用が制限されています。このプロパティーが使用されるかどうかを特定のコネクターについて判別するには、該当するアダプター・ガイドのインストールと構成についての章を参照してください。

PollQuantity

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

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。

このプロパティー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ー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

コネクターが XML スキーマ文書を読み取るリポジトリーの場所です。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、コネクターが InterChange Server リポジトリーからこの情報を取得するためです。

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

ResponseQueue

DeliveryTransport が JMS の場合のみ適用でき、RepositoryDirectory が <REMOTE> の場合のみ必要です。

JMS 応答キューを指定します。JMS 応答キューは、応答メッセージをコネクター・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

コネクターによるコネクター自体の再始動の試行回数を指定します。このプロパティーを並列コネクターに対して使用する場合、コネクター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

コネクターによるコネクター自体の再始動の試行間隔を分単位で指定します。このプロパティーを並列コネクターに対して使用する場合、コネクター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。指定できる値の範囲は 1 から 2147483647 です。

デフォルト値は 1 です。

SourceQueue

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

JMS イベント・ストアを使用する JMS 対応コネクター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ー・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、88 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 の場合のみ適用できます

同期応答を要求する要求メッセージを、コネクター・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、コネクター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、コネクター・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。コネクター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。

デフォルト値は CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE です。

SynchronousResponseQueue

DeliveryTransport が JMS の場合のみ適用できます。

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ー・フレームワークに配信します。このキューは、コネクターが同期実行を使用する場合にのみ必要です。

デフォルト値は CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE です。

SynchronousRequestTimeout

DeliveryTransport が JMS の場合のみ適用できます。

コネクター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。コネクターは、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、設定は CwXML です。
- RepositoryDirectory の値が <REMOTE> の場合、設定は CwBO です。

WsifSynchronousRequest Timeout

WAS 統合ブローカーでのみ使用されます。

コネクター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。コネクターは、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強力なプロパティーです。

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ー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。

- コネクターを構成するためのコネクター固有のプロパティー・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ーを設定する

注:

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

この付録では、次のトピックについて説明します。

Connector Configurator の概要

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ー・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。

- コネクターを構成するための**コネクター固有のプロパティー・テンプレート**を作成する。
- **コネクター構成ファイル**を作成します。インストールするコネクター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ーを設定する。
  場合によっては、コネクター・テンプレートでプロパティー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る場合があります。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 (100 ページの『スタンドアロン・モードでの Configurator の実行』を参照)。

コネクター構成プロパティーには、標準の構成プロパティー (すべてのコネクターがもつプロパティー) と、コネクター固有のプロパティー (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ーで必要なプロパティー) とが含まれます。

標準プロパティーはすべてのコネクターにより使用されるので、標準プロパティー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ー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ーの値を設定する必要があります。

標準プロパティーの範囲は、ブローカーと構成によって異なる可能性があります。特定のプロパティー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ーがあります。Connector Configurator の「標準のプロパティー」ウィンドウには、特定の構成で設定可能なプロパティーが表示されます。

ただし**コネクター固有プロパティー**の場合は、最初にプロパティーを定義し、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ー固有プロパティー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、101 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 環境でコネクター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、コネクター構成ファイルを編集できます。

これを行うには、以下のステップを実行します。

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

- **「システム接続: Integration Broker」**の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、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 (106 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、**「コネクター」**を強調表示します。
3. System Manager メニュー・バーから、**「ツール」>「Connector Configurator」**をクリックします。「Connector Configurator」ウィンドウが開き、**「新規コネクター」**ダイアログ・ボックスが表示されます。
4. **「システム接続: Integration Broker」**の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、ICS、WebSphere Message Brokers、または WAS を選択します。

既存の構成ファイルを編集するには、以下のステップを実行します。

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

コネクター固有のプロパティー・テンプレートの作成

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

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

- テンプレートの新規作成については、101 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、新しい名前でこのテンプレートを保管します。

新規テンプレートの作成

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

テンプレートは以下のように作成します。

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

     このテンプレートが使用されるコネクター (またはコネクターのタイプ) を表す固有の名前を入力します。テンプレートから新規構成ファイルを作成するためのダイアログ・ボックスを開くと、この名前が再度表示されます。
   - **「旧テンプレート」、「変更する既存のテンプレートを選択してください」**

     「テンプレート名」表示に、現在使用可能なすべてのテンプレートの名前が表示されます。
   - テンプレートに含まれているコネクター固有のプロパティー定義を調べるには、**「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ー定義のリストが**「テンプレートのプレビュー」**表示に表示されます。テンプレート作成の取り掛かりとして、コネクターに必要なプロパティー定義に似たプロパティー定義を持つ既存のテンプレートを使用できます。
3. **「テンプレート名」**表示からテンプレートを選択し、その名前を**「名前の検索 (Find Name)」**フィールドに入力し (または**「テンプレート名」**で自分の選択項目を強調表示し)、**「次へ」**をクリックします。

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

一般特性の指定

「次へ」をクリックしてテンプレートを選択すると、**「プロパティー: コネクター固有プロパティー・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ーの「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:**
  プロパティー・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

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

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

   「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、**「プロパティー値」**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ー・テンプレート」**ダイアログ・ボックスが表示されます。

従属プロパティーは、別のプロパティー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ーです。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。プロパティーを従属プロパティーとして指定し、従属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ー」**表示で、従属プロパティーとして指定するプロパティーを選択します。
2. **「プロパティーを選択」**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ーを選択します。
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、従属プロパティー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ー」**表示で強調表示された従属プロパティーで、矢印をクリックし、**「従属プロパティー」**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、最初に統合ブローカーを選択します。選択したブローカーによって、構成ファイルに記述されるプロパティーが決まります。

ブローカーを選択するには、以下のステップを実行します。

- Connector Configurator のホーム・メニューで、**「ファイル」>「新規」>「コネクター構成」**をクリックします。**「新規コネクター」**ダイアログ・ボックスが表示されます。
- **「Integration Broker」**フィールドで、ICS 接続、WebSphere Message Brokers 接続、WAS 接続のいずれかを選択します。
- この章で後述する説明に従って**「新規コネクター」**ウィンドウの残りのフィールドを入力します。

また、以下の作業も実行できます。

- 「System Manager」ウィンドウで**「コネクター」**フォルダーを右マウス・ボタン・クリックし、**「新規コネクターの作成」**を選択します。Connector Configurator が開き、**「新規コネクター」**ダイアログ・ボックスが表示されます。

コネクター固有のテンプレートからの構成ファイルの作成

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

1. **「ファイル」>「新規」>「コネクター構成」**をクリックします。
2. 以下のフィールドを含む**「新規コネクター」**ダイアログ・ボックスが表示されます。
   - **名前**

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

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

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

- **「コネクター固有プロパティー・テンプレート」を選択します。**

  ご使用のコネクター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、**「プロパティー・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ー固有のプロパティーが表示されます。

  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ー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ー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・コネクターを保管」**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、コネクターの始動ファイルで指定するコネクター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、コネクター定義を完了します。

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。

- コネクター定義ファイル。
  コネクター定義ファイルは、特定のコネクターのプロパティーと、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。コネクターの配布パッケージの `¥repository` ディレクトリー内には、このようなファイルが格納されていることがあります (通常、このファイルの拡張子は `.txt` です。例えば、XML コネクターの場合は CN_XML.txt です)。
- ICS リポジトリー・ファイル。
  コネクターの以前の ICS インプリメンテーションで使用した定義は、そのコネクター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。
- コネクター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、コネクターのコネクター固有プロパティーのほとんど、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、コネクター構成ファイルは、ファイルを開いて、プロパティー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ーを構成するには、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」**とクリックします。
2. **「ファイル・コネクターを開く」**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 環境でのコネクター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ー定義が含まれている場合は、ファイルを開くとすべての定義が表示されます。
   - すべてのファイル (*.*)

     コネクターのアダプター・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、適切なコネクター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ー構成を開くには、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」**とクリックします。

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. **「標準のプロパティー」**タブで、BrokerType プロパティー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ーが「標準のプロパティー」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 109 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**「ファイルにすべて保管」**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**「プロジェクトにすべて保管」**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ーすべてに値が設定されているかどうかが確認されます。必須の標準プロパティー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、そのプロパティーの値を指定する必要があります。

構成ファイル・プロパティーの設定

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ーに値が設定されている必要があります。

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

注: JMS メッセージングを使用するコネクターの場合は、データをビジネス・オブジェクトに変換するデータ・ハンドラーの構成に関して追加のカテゴリ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ーの場合、以下のプロパティーの値も設定されている必要があります。

- 関連マップ
- 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ー値も設定可能です。ただし、標準のプロパティーおよびコネクター固有プロパティー、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。

標準プロパティーとコネクター固有プロパティーの違いは、以下のとおりです。

- コネクターの標準プロパティーは、コネクターのアプリケーション固有のコンポーネントとブローカー・コンポーネントの両方によって共用されます。すべてのコネクターが同じ標準プロパティーのセットを使用します。これらのプロパティーの説明は、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プロパティーは、コネクターのアプリケーション固有コンポーネント (アプリケーションと直接対話するコンポーネント) のみに適用されます。各コネクターには、そのコネクターのアプリケーションだけで使用されるアプリケーション固有のプロパティーがあります。これらのプロパティーには、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ーおよび推奨値が記述されています。

「標準のプロパティー」と**「コネクター固有プロパティー」**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、標準のプロパティーを表します。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ー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ーを表します。これらのプロパティーは、アプリケーションまたはコネクター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ーの除去も可能です。
- 「値」フィールドは構成できます。
- **「更新メソッド」**フィールドは通知を行うもので、構成できません。このフィールドでは、値が変更されたプロパティーをアクティブにするために必要なアクションを指定します。

標準コネクター・プロパティーの設定

標準のプロパティー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る場合にはメニューから値を選択します。
3. 標準のプロパティーの値をすべて入力すると、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。**「標準のプロパティー」** (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きに、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ーの設定

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

1. グリッドの左上端の部分で右マウス・ボタン・クリックします。ポップアップ・メニュー・バーが表示されます。プロパティーを追加するときは**「追加」**をクリックします。子プロパティーを追加するには、親の行番号で右マウス・ボタンをクリックし、**「子を追加」**をクリックします。
2. プロパティーまたは子プロパティーの値を入力します。
3. プロパティーを暗号化するには、**「暗号化」**ボックスを選択します。
4. 108 ページの『標準コネクター・プロパティーの設定』で説明したように、変更内容を保管するかまたは破棄するかを選択します。

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

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

コネクター・プロパティーの暗号化

アプリケーション固有のプロパティーは、**「プロパティーを編集」**ウィンドウの**「暗号化」**チェック・ボックスにチェックマークを付けると、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、暗号化が解除された値が表示されます。

各プロパティー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・ガイドにあります。

プロパティーに複数の値がある場合には、プロパティーの最初の値に**「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、そのプロパティーのすべての値が暗号化されます。プロパティーの複数の値を暗号化解除するには、そのプロパティーの最初の値の**「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、すべての複数値が暗号化解除されます。

更新メソッド

付録『コネクターの*標準構成プロパティー*』の 82 ページの『プロパティー値の設定と更新』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

Connector Configurator の**「サポートされているビジネス・オブジェクト」**タブで、コネクターが使用するビジネス・オブジェクトを指定します。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」**リストの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、**「プロジェクトに保管」**をクリックします。

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

エージェント・サポート: ビジネス・オブジェクトにエージェント・サポートがある場合、システムは、コネクター・エージェントを介してアプリケーションにデータをデリバリーする際にそのビジネス・オブジェクトの使用を試みます。

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

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

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

ほとんどのコネクターの場合、選択可能な項目は「最大限の努力」のみです。

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

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

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。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 を選択する場合、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、サポートされているビジネス・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できるビジネス・オブジェクトのリストが示されます。このリストから目的のビジネス・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マップのリストをサポートします。このリストは、**「関連付けられたマップ」**タブを選択すると表示されます。

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

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

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

「関連付けられたマップ」タブには以下のフィールドが表示されます。

- **ビジネス・オブジェクト名**

  これらは、**「サポートされているビジネス・オブジェクト」**タブで指定した、このコネクタ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、サポートされるビジネス・オブジェクトを追加指定した場合、それらの内容は、「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 (Explicit)**

  場合によっては、関連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、コネクターごとに、サポートされる各ビジネス・オブジェクトにマップを自動的にバインドしようとします。複数のマップでその入力データとして同一のビジネス・オブジェクトが使用されている場合、サーバーは、他のマップのスーパーセットである 1 つのマップを見付けて、バインドしようとします。

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

  以下の手順を実行して、マップを明示的にバインドします。

1. **「明示的 (Explicit)」**列で、バインドするマップのチェック・ボックスにチェックマークを付けます。
2. ビジネス・オブジェクトに関連付けるマップを選択します。
3. 「Connector Configurator」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 にデプロイします。
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することができます。
すべてのコネクターでこの機能がサポートされ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・プロパティーは、`DeliveryTransport` 標準プロパティーの値として 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ーは、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

コネクター構成ファイルまたはコネクター定義ファイルを開くと、Connector Configurator 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ーの**「トレース/ログ・ファイル」**タブでのみ使用できます。
   - ファイルに:
     ログ・メッセージまたはトレース・メッセージを指定されたファイルに書き込みます。ファイルを指定するには、ディレクトリー・ボタン (省略符号) をクリックし、指定する格納場所に移動し、ファイル名を指定し、**「保管」**をクリックします。ログ・メッセージまたはトレース・メッセージは、指定した場所の指定したファイルに書き込まれます。

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

データ・ハンドラー

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

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

構成ファイルの保管

コネクターの構成が完了したら、コネクター構成ファイルを保管します。Connector Configurator では、構成中に選択したブローカー・モードで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。

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

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

構成ファイルの変更

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

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

既存の構成ファイルでのブローカーの選択を変更するには、以下の手順を実行します (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**「ブローカー・タイプ」**フィールドで、ご使用のブローカーに適した値を選択します。
  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変更され、選択した新規ブローカーに適したタブとフィールドのみが表示されます。

構成の完了

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

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

グローバル化環境における Connector Configurator 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ーのドロップ・リストに表示されるのは、サポートされる値の一部のみです。ドロップ・リストに、サポートされる他の値を追加するには、製品ディレクトリーの ¥Data¥Std¥stdConnProps.xml ファイルを手動で変更する必要があります。

例えば 「Locale」 プロパティーの値のリストにロケール en_GB を追加するには、stdConnProps.xml ファイルを開き、以下に太字で示される行を追加してください。

```
<Property name="Locale"
isRequired="true"
updateMethod="component restart">
                <ValidType>String</ValidType>
            <ValidValues>
                                <Value>ja_JP</Value>
                                <Value>ko_KR</Value>
                                <Value>zh_CN</Value>
```

```
                                <Value>zh_TW</Value>
                                <Value>fr_FR</Value>
                                <Value>de_DE</Value>
                                <Value>it_IT</Value>
                                <Value>es_ES</Value>
                                <Value>pt_BR</Value>
                                <Value>en_US</Value>
                                <Value>en_GB</Value>
                        <DefaultValue>en_US</DefaultValue>
                </ValidValues>
        </Property>
```

付録 C. チュートリアル

この付録では、アダプターに付属しているサンプルのインストール、構成、および使用方法について説明します。このチュートリアルは、InterChange Server (ICS) 統合ブローカーを使用することを前提として設計されています。表記規則のガイドについては、この文書のまえがきを参照してください。

前提条件

このチュートリアルを使用する前に、IBM WebSphere Business Integration 製品をインストールして使い慣れておく必要があります。以下のタスクを完了してから、サンプルをインストールしてください。

1. WebSphere MQ Workflow のアダプターのインストール。このチュートリアルのサンプルは、WebSphere MQ Workflow バージョン 3.2.2、3.3.2 および 3.4.x 用に設計されています。他のバージョンに対しては、サポートが異なる場合があります。
2. ポート・コネクターのインストール。 (実際のエージェントはポート・コネクターに関連付けられていませんが、コネクター定義はリポジトリー内に必ずなければなりません。)
3. WebSphere MQ Workflow Adapter の定義およびポート・コネクターの定義がリポジトリー内にない場合は、Connector Configurator を使用してそれらの定義をロードしてください。
   a. 「ファイル」メニュー項目から「ファイル」>「開く」を選択します。
   b. ¥connectors¥WebSphereMQWorkflow¥samples フォルダーにあるリポジトリー・ファイル Sample_MQWorkflow_Order_Connectors.in をロードします。
   c. MQWorkflowConnector 定義および PortConnector 定義がロードされたことを確認します。
4. Java 用 IBM WebSphere MQ クライアント・ライブラリーをインストールします。

プリインストール・チェックリスト

サンプルをインストールする前に以下の情報を収集してください。

- IBM WebSphere InterChange Server の名前 (デフォルト: LocalHost)
  ICS 名 = _____
- WebSphere MQ Workflow キュー・マネージャー (デフォルト: FMCQM)
  キュー・マネージャー名 = _____
- キュー・マネージャーの CCSID (デフォルト: 819)
  キュー・マネージャーの CCSID = _____
- キュー・マネージャーのリスナー用に設定するポート (デフォルト: 5010)
  キュー・マネージャーのポート = _____
- キュー・マネージャーのホスト (デフォルト: LocalHost)
  キュー・マネージャーのホスト = _____

- キュー・マネージャーで使用するサーバー接続チャネル (デフォルト: FMCQM.CL.TCP)
  キュー・マネージャー・チャネル = _____
- WebSphere MQ Workflow システム名 (デフォルト: FMCSYS)
  WebSphere MQ Workflow システム名 = _____
- WebSphere MQ Workflow システム・グループ (デフォルト: FMCGRP)
  WebSphere MQ Workflow システム・グループ = _____
- WebSphere MQ Workflow アカウント・ユーザー名 (デフォルト: ADMIN)
  WebSphere MQ Workflow アカウント・ユーザー名 = _____

  注: このアカウントは WebSphere MQ Workflow 内に必ず存在していて、システム・アカウント内の MQM のグループの一部となっていなければなりません。このことが保証されないと、アダプターは WebSphere MQ を介してメッセージを発行することも WebSphere MQ Workflow で WebSphere MQ Workflow プロセスを実行することもできません。
- WebSphere MQ Workflow 構成名 (デフォルト: FMC)
  WebSphere MQ Workflow 構成名 = _____

環境のセットアップ

このセクションでは、サンプルを使用して作業できる環境の準備の仕方について説明します。後出の *sample_folder* は、サンプルがあるフォルダーを指します。*WBI_folder* は、現在インストールされている IBM WebSphere Business Integration が格納されているフォルダーを指します。

1. **キューを作成します。** このチュートリアルでは、WebSphere MQ Workflow キュー・マネージャーで 6 つのローカル・キューを定義しておく必要があります。これらのローカル・キューは、WebSphere MQ Explorer アプリケーションを介して作成するか、コマンド行で「RUNMQSC FMCQM」と入力して以下のコマンドを発行して作成します。
   - DEFINE QL('MQWFCONN.ERROR')
   - DEFINE QL('MQWFCONN.ARCHIVE')
   - DEFINE QL('MQWFCONN.IN_PROGRESS')
   - DEFINE QL('MQWFCONN.REPLYTO')
   - DEFINE QL('MQWFCONN.UNSUBSCRIBED')
   - DEFINE QL('CWLDINPUTQ')
2. **ビジネス・オブジェクト定義を作成します (省略可能)。** WebSphere MQ Workflow データ構造用のビジネス・オブジェクト定義ファイルはすでに存在していますが、このステップを実行することで、FDLBORGEN ユーティリティーを使用して既存のデータ構造をビジネス・オブジェクトに実際に変換してみることができます。このユーティリティーを使用するには、以下の手順を実行します。
   a. コマンド行で、ディレクトリーを *WBI_folder*/connectors/WebSphereMQWorkflow/utilities に変更します。
   b. 以下のように入力します。

```
FdlBorgen -isample_folder/WebSphereMQWorkflow_Samples.fdl -
osample_folder/SampleItem.in -nSampleItem -pMQWF_Structure_ -r3.1.0

FdlBorgen -isample_folder/WebSphereMQWorkflow_Samples.fdl -
osample_folder/SampleItemOrder.in -nSampleItemOrder -pMQWF_Structure_
-r3.1.0
```

3. **サンプル・ビジネス・オブジェクトをリポジトリーにロードします。**IBM WebSphere ICS を起動し、Business Object Designer を使用して、「ファイル」>「ファイルから開く」を選択します。Sample_MQWF_Order_Objects.in という名前のリポジトリー・ファイル (*WBI_folder*/connectors/WebSphereMQWorkflow/Samples フォルダーにあります) をロードします。サンプル・ビジネス・オブジェクトがロードされたことを確認します。
4. **サンプル・コラボレーション・テンプレート、およびコラボレーション・オブジェクトをリポジトリーにロードします。**WebSphere Business Integration System Manager を使用して、「ファイル」>「ファイルから開く」を選択します。sample_MQWF_Order_Collaborations.in という名前のリポジトリー・ファイル (*WBI_folder*/connectors/WebSphereMQWorkflow/Samples フォルダーにあります) をロードします。
5. **コラボレーション・テンプレートをコンパイルします。** WebSphere Business Integration System Manager を使用して、「コラボレーション・テンプレート」というラベルが付いたフォルダーを右マウス・ボタンでクリックし、ドロップダウン・リストから「すべてコンパイル」を選択します。
6. **IBM WebSphere ICS を再始動します。** 変更をすべて有効にするために、Interchange Server をリブートします。System Monitor を使用して、コラボレーション・オブジェクトおよびコネクター・コントローラーがすべて正常であることを確認します。

サンプル、テンプレート、アダプター、およびマップの構成

このセクションでは、サンプルについて説明したうえで、サンプルおよびアダプターの構成方法についても説明します。

サンプルのコンテンツについて

以下のサンプルがあります。

- MQWF_DataStructure_SampleItemOrder WebSphere MQ Workflow 内の SampleItemOrder という名前のデータ構造を表すビジネス・オブジェクトです。
- MQWF_DataStructure_SampleItemOrder_Item WebSphere MQ Workflow 内のデータ構造 SampleItemOrder の中に含まれている子データ構造 Item を表すビジネス・オブジェクトです。
- MQWF_DataStructure_SampleItem WebSphere MQ Workflow 内のデータ構造 SampleItem を表すビジネス・オブジェクトです。
- MQWF_SampleItemOrder MQWF_DataStructure_SampleItemOrder 用のコンテナー・オブジェクトです。入出力データ構造と、アダプターが使用する各種メタオブジェクトを保持しています。このオブジェクトは、データ構造 SampleItemOrder を処理する際に、アダプターとコラボレーション間で受け渡されます。

- MQWF_SampleItem MQWF_DataStructure_SampleItem 用のコンテナー・オブジェクトです。入出力データ構造と、アダプターが使用する各種メタオブジェクトを保持しています。このオブジェクトは、データ構造 SampleItem を処理する際に、アダプターとコラボレーション間で受け渡されます。
- MO_MQWorkflow_ProcessInstance WebSphere MQ Workflow プロセスの追跡と制御に使用されるオブジェクトです。
- MQWF_SampleItemRequest MQWF_DataStructure_SampleItemRequest 用のコンテナー・オブジェクトです。WebSphere MQ Workflow が InterChange Server に要求を送信するときに使用されます。入力データ構造と、アダプターが使用する各種メタオブジェクトを保持しています。このオブジェクトは、データ構造 SampleItemRequest を処理する際に、アダプターからコラボレーションに渡されます。
- MQWF_SampleItemResponse MQWF_DataStructure_SampleItemRequest 用のコンテナー・オブジェクトです。InterChange Server が WebSphere MQ Workflow に応答を返すときに使用されます。出力データ構造と、アダプターが使用する各種メタオブジェクトを保持しています。このオブジェクトは、データ構造 SampleItemRequest を処理する際に、コラボレーションからアダプターに渡されます。
- MQWF_SampleItem MQWF_DataStructure_SampleItem 用のコンテナー・オブジェクトです。入出力データ構造を保持し、コラボレーション内の架空の汎用ビジネス・オブジェクトを表しています。
- SampleItemOrderSync_MQWF_to_Port および SampleItemOrderSync_Port_to_MQWF アダプターとポート・コネクターの間で、ビジネス・オブジェクト MQWF_SampleItemOrder のやり取りに使用されるコラボレーション・オブジェクトです。
- SampleItemSync_MQWF_to_Port および SampleItemSync_Port_to_MQWF アダプターとポート・コネクターの間で、ビジネス・オブジェクト MQWF_SampleItem のやり取りに使用されるコラボレーション・オブジェクトです。
- SampleWorkflowProcessControl_Port_to_MQWF ビジネス・オブジェクト MO_MQWorkflow_ProcessInstance の検索および制御に使用されるコラボレーション・オブジェクトです。
- SampleItemActivity_MQWF_to_MQWF MQWF_GBO_SampleItem をアダプターから受け取ってポート・コネクター・エージェントに渡し、アダプターに返送するために使用されるコラボレーション・オブジェクトです。
- MQWF_Sample_RequesttoGBO 要求ビジネス・オブジェクト MQWF_SampleItemRequest からのデータを架空の汎用ビジネス・オブジェクト MQWF_GBO_SampleItem に転送するマップです。
- MQWF_Sample_GBOtoResponse 架空の汎用ビジネス・オブジェクト MQWF_GBO_SampleItem からのデータを応答ビジネス・オブジェクト MQWF_SampleItemResponse に転送するマップです。

テンプレート・クラス・ファイルのコピー

コラボレーション・テンプレートを機能させるには、関連クラス・ファイルをインクルードする必要があります。

- *sample_folder*/classes 内のファイルを *WBI_folder*/collaborations/classes/UserCollaborations/classes にコピーします。

注: サンプル・コラボレーションは、CollaborationFoundation テンプレート (別個に使用可能) をベースにしているため、コラボレーションの変更に必要な .CLM コンポーネントや .java コンポーネントは IBM から提供されていません。

コネクターの構成

コネクターのプロパティーを構成するには、次のようにします。

- AgentTraceLevel=3
- ApplicationPassword=(*username* 用のパスワード)
- ApplicationUserName=*username*
- ArchiveQueue=MQWFCONN.ARCHIVE
- BOPrefix=MQWF_
- DeliveryTransport=IDL (オプション)
- ErrorQueue=MQWFCONN.ERROR
- InputQueue=CWLDINPUTQ
- MQSeriesCCSID=*CCSID*
- MQSeriesChannel=*CHANNEL*
- MQSeriesHostname=*HOST*
- MQSeriesPort=*PORT*
- MQSeriesQueueManager=*queue manager*
- OutputQueue=FMC.FMCGRP.EXE.XML
- ReplyToQueue=MQWFCONN.REPLYTO
- UnsubscribedQueue=MQWFCONN.UNSUBSCRIBED

サンプル・ビジネス・オブジェクトのサポート

サンプル・ビジネス・オブジェクトを使用して作業するには、アダプターがそのサンプル・ビジネス・オブジェクトをサポートしていることを確認しておく必要があります。

1. System Manager で、WBI Adapter for the WebSphere MQ Workflow およびポート・コネクター定義を開きます。
2. 「サポートされているビジネス・オブジェクト」タブを選択し、次のビジネス・オブジェクトを追加します。
   - MO_MQWorkflow_ProcessInstance
   - MO_MQWorkflow_ProcessTemplateConfig
   - MO_MQWorkflow_ContainerInfo
   - MO_MQWorkflow_ProcessInfo
   - MO_MQWorkflow_ActivityInfo
   - MO_MQWorkflow_ActivityRequest
   - MO_MQWorkflow_ActivityResponse

- MO_DataHandler_Default
- MO_DataHandler_DefaultXMLConfig
- MQWF_SampleItem
- MQWF_Structure_SampleItem
- MQWF_SampleItemOrder
- MQWF_Structure_SampleItemOrder
- MQWF_Structure_SampleItemOrder_Item
- MQWF_SampleItemRequest
- MQWF_SampleItemResponse
- MQWF_Structure_SampleItemRequest

マップのバインディング

マップによるデータ転送を可能にするには、マップをアダプターに関連付ける必要があります。

1. System Manager で、アダプター定義を開きます。
2. 「関連付けられたマップ」タブをクリックします。
3. 次のマップについて「明示的バインド」を確認します。
   - MQWF_Sample_RequesttoGBO
   - MQWF_Sample_GBOtoResponse

最終の構成ステップ

1. 変更をすべて有効にするために、Interchange Server をリブートします。
2. *sample_folder*/WebSphereMQWorkflow_Samples.fdl を IBM WebSphere MQ Workflow 実行時サーバーにインポートします。

   注: 実行時サーバーへの FDL ファイルのインポートについては、「*IBM WebSphere MQ Workflow: 定義機能の開始*」を参照してください。既存の実行時データを上書きしても差し支えない場合は、コマンド行ウィンドウを開いて次のように入力すれば、サンプル・ワークフローをすばやくロードすることができます (警告: この操作を実行すると実行時データがすべて上書きされます)。

   ```
   fmcibie /i=WebSphereMQWorkflow_Samples.fdl /u=ADMIN /y= /t
   /o (password: password)
   ```

シナリオの実行

シナリオを実行するには、以下の手順に従ってください。

1. InterChange Server がまだ稼働していない場合は始動します。
2. WBI Adapter for WebSphere MQ Workflow がまだ稼働していない場合は始動します。この場合、-fkey オプションを (自動ポーリングを無効にするため) 指定します。
3. Visual Test Connector がまだ稼働していない場合は始動します。

Visual Test Connector を始動し、PortConnector 用のプロファイルを定義してエージェントをバインドすることにより、ポート・コネクターをシミュレートします。

同期要求

このシナリオでは、ビジネス・データを定義済みワークフロー・プロセスに渡し、最終結果を取得します。これは、同期呼び出しです。アダプターは WebSphere MQ Workflow への要求発行後、開始されたワークフロー・プロセスが完了するまでブロックするためです。

このシナリオでは、ワークフロー・プロセス Lookup_Order_Status にオーダー・キーを渡して、架空のオーダーの状況を取得します。このワークフローの唯一のアクションは、オーダー情報を取得するために IBM WebSphere Business Integration Server に対して Retrieve を発行することだけです。このアクションは、同期要求がアダプターから WebSphere MQ Workflow に発行される仕組みと、WebSphere MQ Workflow からアダプターに発行される仕組みを例示しています。

1. **サンプル・オーダーを作成します。** Visual Test Connector を使用して、動詞 Create を設定したビジネス・オブジェクト MQWF_SampleItemOrder のインスタンスを新規に作成し、このインスタンスに次のようなデータを取り込みます (未定義の値は CxIgnore とします)。
   MQWF_SampleItemOrder
   - Input_ItemOrder
     - TrackingNumber = ABC123
   - MO_Config
     - ProcessTemplateName = Lookup_Order_Status
     - KeepName = false
     - UserId = UserName
     - ExecutionMode = Synchronous
     - ResponseTimeout = 600000
     - TimeoutFatal = false
2. **このオブジェクトをアダプターに送信します。** アダプターによってこのオブジェクトは要求メッセージに変換されて WebSphere MQ Workflow サーバーに発行されます。アダプターはすぐには戻らずに、WebSphere MQ Workflow からの応答を待機し始めます。
   WebSphere MQ Workflow はアダプターから要求を受け取り、オブジェクト MQWF_SampleItemOrder.Input_ItemOrder からのデータを使用して、プロセス・テンプレート Lookup_Order_Status を同期的に作成して起動します。このワークフロー・プロセスにおいて最初で唯一のステップは、トラッキング番号をキーとして使用して IBM WebSphere Business Integration Server からデータ構造 SampleItemOrder を取得するステップです。このステップを実行するために、MQ Workflow は要求メッセージをアダプターの入力キューに発行し、応答自体を待機し始めます。これは、WebSphere MQ Workflow クライアント・アプリケーションをチェックすることにより確認できます。
3. アダプター・エージェント・ウィンドウで p キーを押してイベントをポーリングします。アダプターは、WebSphere MQ Workflow が (当初の要求によって起動されて) 発行した要求を見つけ、要求メッセージを、動詞 Retrieve を設定し

たオブジェクト MQWF_SampleItemOrder に変換して、コラボレーション SampleItemOrderSync_MQWF_to_Port に通知します。

4. Visual Test Connector を介して要求が受け付けられます。オブジェクト MQWF_SampleItemOrder.Input_ItemOrder の属性 TrackingNumber が ABC123 (当初の要求の属性値と同じ) であることを確認してください。オブジェクト MQWF_SampleItemOrder.Output_ItemOrder に次のようにデータを取り込み、「応答の完了」を選択して要求を完了します。

   • Output_ItemOrder
     – TrackingNumber = ABC123
     – Approved = YES

   アダプターは応答を WebSphere MQ Workflow に戻して、MQWF_SampleItemOrder.Output_ItemOrder 内に格納されたビジネス・データを渡します。WebSphere MQ Workflow はアダプターから応答を受け取り、当初の要求に対する応答メッセージにそのデータを組み込み、アダプターの ReplyTo キューに発行して戻します。アダプターは応答メッセージを取得し、変更内容またはエラーをコラボレーションに戻します。これで同期ワークフロー要求は完了です。コラボレーションに戻されたオブジェクトには、次のようにデータが取り込まれます。

   • MQWF_SampleItemOrder
     – Input_ItemOrder
       - TrackingNumber = ABC123
     – MO_Config
       - ProcessTemplateName = Lookup_Order_Status
       - KeepName = false
       - UserId = UserName
       - ExecutionMode = Synchronous
       - ResponseTimeout = 600000
       - TimeoutFatal = false
     – Output_ItemOrder
       - TrackingNumber = ABC123
       - Approved = YES
     – ProcessInstance これにはデータが取り込まれますが、値は予測できません。プロセスの状態は TERMINATED でなければなりません。

   注: このプロセスを 10 分 (メタオブジェクト内での構成値は 600000 ミリ秒) 以内に完了できなかった場合、アダプターは WebSphere MQ Workflow からの応答の受信に失敗したことを報告します。

非同期要求

このシナリオでは、アダプターはビジネス・データを定義済みワークフロー・プロセスに渡しますが、このプロセスの完了を待機しません。これは非同期呼び出しです。なぜなら、アダプターは WebSphere MQ Workflow への要求発行後にプロセス ID を取得し、取得した ID を使用してプロセスを追跡しながらプロセスを並列実行するためです。このシナリオでは、アダプターがワークフロー・プロセス

Approve_Order にオーダーを発行して、オーダー承認のタスクを開始します (承認が完了したかどうかについては後で確認します)。ワークフロー・プロセスでは、注文した品目についての情報を検索し、該当品目の在庫が十分な量だけあるかどうかを基準に、オーダー承認を更新します。このシナリオでは、アダプターがワークフロー・プロセスの開始を非同期式に起動し、並列実行されているワークフロー・プロセスの状況をモニターする仕組みを例示します。

1. Visual Test Connector を使用して、動詞 Create を設定したビジネス・オブジェクト MQWF_SampleItemOrder のインスタンスを新規に作成し、このインスタンスに次のようなデータを取り込みます (未定義の値は CxIgnore とします)。
   - MQWF_SampleItemOrder
     - Input_ItemOrder
       - TrackingNumber = ABC123
       - Customer = Billy Bob
     - Item
       - Name = Hammer
       - Quantity = 1
     - MO_Config
       - ProcessTemplateName = Approve_Order
       - KeepName = false
       - UserId = UserName
       - ExecutionMode = Asynchronous
       - ResponseTimeout = 500
       - TimeoutFatal = false
2. MQWF_SampleItemOrder をアダプターに送信します。すると、アダプターはこのオブジェクトを要求メッセージに変換して WebSphere MQ Workflow サーバーに発行します。その後、アダプターはプロセス・インスタンス ID を含む応答を待機します。 WebSphere MQ Workflow はアダプターから要求を受け取り、オブジェクト MQWF_SampleItemOrder.Input_ItemOrder からのデータを使用して、プロセス・テンプレート Approve_Order を非同期的に作成して起動します。プロセスが開始すると、WebSphere MQ Workflow は、開始されたワークフロー・プロセスの ID を含む応答を即時にアダプターに戻します。また、ワークフロー・プロセスは、最初のステップとして、IBM WebSphere Business Integration Server から name = Hammer が設定されたデータ構造 SampleItem を検索します。アダプターの ReplyTo キューへ応答が発行されると同時に、アダプターの入力キューへ要求が発行されますが、この 2 つは別々のアクションです。アダプターは WebSphere MQ Workflow から応答を受け取って、呼び出し側コラボレーションにビジネス・オブジェクトを戻します。このオブジェクトは、以下のものと似ています。
   - MQWF_SampleItemOrder
     - Input_ItemOrder
       - TrackingNumber = ABC123
       - Customer = Billy Bob
     - Item

- Name = Hammer
- Quantity = 1

– MO_Config
- ProcessTemplateName = Approve_Order
- KeepName = false
- UserId = UserName
- ExecutionMode = Asynchronous
- ResponseTimeout = 5000
- TimeoutFatal = false

– ProcessInstance これにはデータが取り込まれますが、値は予測できません。

注: 属性 ProcessInstanceID の値は忘れないようにしてください。後でこのチュートリアルで使用します。

この時点で、ワークフロー・プロセスはコラボレーション処理と並列実行されています。ワークフロー・プロセスの追跡または制御は、オブジェクト MQWF_SampleItemOrder.ProcessInstance 内に戻される ProcessInstanceID を介してのみ可能です。

3. Visual Test Connector を使用して、動詞 Terminate を設定したビジネス・オブジェクト MO_MQWorkflow_ProcessInstance のインスタンスを新規に作成します。XML API の使用は、WebSphere MQ Workflow 3.4 では必須、WebSphere MQ Workflow 3.3.2 では推奨です。XML API を使用するには、「アダプター構成プロパティー (Adapter Configuration Property)」を JavaCorbaApi = False に設定し、ワークフロー・プロセス (MO_MQWorkflow_ProcessInstance) をモニターするには、ProcessInstanceName = *ProcInstName* (前のステップで戻された *ProcInstName*) を設定します。

   注: XML API 用にサポートされている動詞は、Restart および Delete のみです。

4. MO_MQWorkflow_ProcessInstance をアダプターに送信します。すると、アダプターからワークフロー・プロセスの状況が戻されます。属性 ProcInstState は RUNNING と等しくなければなりません。
5. このシナリオで開始したワークフロー・プロセスを再開するには、アダプター・エージェント・ウィンドウで p キーを押してイベントをポーリングします。アダプターは、WebSphere MQ Workflow が (当初の要求によって起動されて) 発行した要求を見つけ、この要求メッセージを、動詞 Retrieve を設定したオブジェクト MQWF_SampleItem に変換して、コラボレーション SampleItemSync_MQWF_to_Port に通知します。
6. Visual Test Connector を介して要求が受け付けられます。オブジェクト MQWF_SampleItem.Input_Item の属性 Name が Hammer であることを確認します。オブジェクト MQWF_SampleItem.Output_Item に次のようにデータを取り込み、「応答の完了」を選択して要求を完了します。

   • Output_Item
   – Name = Hammer
   – Price = 14.99

– Stock = 20

アダプターは応答を WebSphere MQ Workflow に戻して、MQWF_SampleItem.Output_Item 内に格納されたビジネス・データを渡します。WebSphere MQ Workflow は、アダプターから応答を受け取り、Stock の値が当初のオーダーの Quantity の値よりも大きいかどうかをチェックします。大きい場合、ハンマーについてはオーダーを完了するための在庫が十分あることになるため、オーダーは承認されます。ワークフロー・プロセスは、InterChange Server でオーダーを更新し、SampleItemOrder データ構造をアダプターの入力キューに発行して、最終ステップを実行します。この際に使用するキーは当初のオーダーと同じですが、この時点では属性 Approve が Y に等しくなります。

7. この最終の要求を WebSphere MQ Workflow から処理するには、アダプター・エージェント・ウィンドウで p キーを押してイベントをポーリングします。アダプターは、WebSphere MQ Workflow が発行した要求を見つけ、動詞 Update を設定したオブジェクト MQWF_SampleItemOrder を作成して、そのオブジェクトをコラボレーション SampleItemOrderSync_MQWF_to_Port に通知します。この要求は受け入れるだけでかまいません (このシナリオでは、レコードはいっさい更新されません)。アダプターが応答を発行し終えると、IBM WebSphere Business Integration Server からの要求、およびワークフロー・プロセスの両方が完了します。

ワークフロー・プロセスの制御

このシナリオでは、進行中のプロセスを終了してワークフロー・プロセスを制御する仕組みを例示します。

1. 124 ページの『非同期要求』の説明に従ってシナリオ 2 を開始します。ただし、動詞 Retrieve を設定したビジネス・オブジェクト MO_MQWorkflow_ProcessInstance を発行する変わりに、その動詞を Suspend に変更します。このオブジェクトをアダプターに送信し、WebSphere MQ Workflow クライアント・アプリケーションを介してプロセスが中断していることを確認します。

   注: プロセスは InterChange Server に対する最初の要求を完了するまで SUSPENDING の状態のままになります。これは、アダプターではなく WebSphere MQ Workflow の機能が反映されたためです。

2. 動詞を Resume に変更して、オブジェクトを再送します。ワークフロー・プロセスの状態が元の RUNNING に変わります。
3. 動詞を Terminate に変更して、オブジェクトを再送します。ワークフロー・プロセスの状態が元の TERMINATED に変わります。この状態変化は、動詞 Retrieve を設定したオブジェクトを発行すれば確認できます。このような方法で、ワークフロー・プロセスの状態は、ICS を介して正常に制御およびモニターされました。

   注: 要求メッセージは、この要求を生成したプロセスが終了しても、アダプターの入力キュー内に残ったままになる可能性がありますが、これは正常です。この要求はアダプターによって処理されますが、WebSphere MQ Workflow は生成された応答をすべて無視します。

WebSphere MQ Workflow からの同期要求

このシナリオでは、WebSphere MQ Workflow から ICS への同期要求、およびその応答をシミュレートします。シナリオ 1 との違いは、アダプターがコラボレーションを非同期に呼び出している点です (これに対し、シナリオ 1 では常に同期呼び出しです)。このシナリオは、ワークフロー・プロセスの完了を別のワークフロー・プロセスが待つ必要がないため、シナリオ 1 よりも実際的です。

1. 更新要求を作成します。WebSphere MQ Workflow クライアントで、ワークフロー・プロセスのインスタンスを作成して開始してから、次のように入力データ構造 SampleItemRequest を設定します。
   - SampleItemRequest
     - Name = Hammer
     - Price = 14.99
     - Stock = 20

   WebSphere MQ Workflow は、この要求をアダプターの入力キューに同期的に発行して、応答を待ちます。
2. アダプター・エージェント・ウィンドウで p キーを押してイベントをポーリングします。アダプターは、WebSphere MQ Workflow が発行した要求を見つけ、この要求メッセージを、動詞 Update を設定したオブジェクト MQWF_SampleItemRequest に変換します。さらにアダプターは、マップ MQWF_Sample_RequesttoGBO を使用してビジネス・オブジェクトを汎用ビジネス・オブジェクト MQWF_GBO_SampleItem に変換し、この汎用ビジネス・オブジェクトを非同期的にパブリッシュします。この時点で、アダプターは応答を待機せずに、別の要求を受信できるようになります。
3. Visual Test Connector を介して要求が受け付けられます。汎用ビジネス・オブジェクトは、サブスクライブ元のコラボレーション SampleItemRequest_MQWF_to_MQWF によって受信され、Visual Test Connector によって受け入れられます。オブジェクト MQWF_GBO_SampleItem.InputItem および MQWF_GBO_SampleItem.OutputItem の属性 Name が Hammer (当初の要求の属性値と同じ) であることを確認してください。次のようにオブジェクト MQWF_GBO_SampleItem.OutputItem の空の属性を設定し、「応答の完了」を選択して、汎用ビジネス・オブジェクトをアダプターに送信します。
   - MQWF_GBO_SampleItemOrder
     - ContainerInfo
       - ReturnCode 以外の属性をすべて設定します。
       - (ActImplCorrelID は、MQWF では ID として使用されるため、変更しないでください)
     - InputItem
       - Name = Hammer
       - Price = 14.99
       - Stock = 20
     - OutputItem
       - Name = Hammer
       - Price = 11.25

- Stock = 8

アダプターは、汎用ビジネス・オブジェクトを受信し、マップ MQWF_Sample_GBOtoResponse を使用してオブジェクト MQWF_SampleItemResponse に変換します。アダプターは、ビジネス・データを MQWF_SampleItemResponse.Output_Item に格納して応答を WebSphere MQ Workflow に戻します。WebSphere MQ Workflow は、アダプターから応答を受け取り、ActImplCorrelID の値をチェックします。ActImplCorrelID の値に一致するワークフロー・プロセスが存在する場合、プロセスは完了します。WebSphere MQ Workflow クライアント内にある対応するプロセス・インスタンスは消滅します。(必要に応じてウィンドウを最新表示してください。)

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で提供し、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すが、有償の場合もあります。

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

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、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にお願いします。

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、単に目標を示しているものです。

プログラミング・インターフェース情報

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

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

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

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

商標

以下は、IBM Corporation の商標です。

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

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

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

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

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

IBM WebSphere Business Integration Adapter Framework V2.4.0

Printed in Japan

日本アイ・ビー・エム株式会社
〒106-8711　東京都港区六本木3-2-12