WebSphere Adapters

Adapter for JDBC ユーザーズ・ガイド

バージョン 6.0

お願い

本書をご使用になる前に、 99 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM WebSphere Adapter for JDBC (5724L77) バージョン 6.0、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： WebSphere Adapters
Adapter for JDBC User Guide
Version 6.0

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 2006.3

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

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

目次

WebSphere Adapter for JDBC バージョン 6.0 ユーザーズ・ガイド

IBM(R) WebSphere(R) Adapter for JDBC は、J2EE アプリケーションとエンタープライズ情報システムの間に双方向の接続性を提供します。

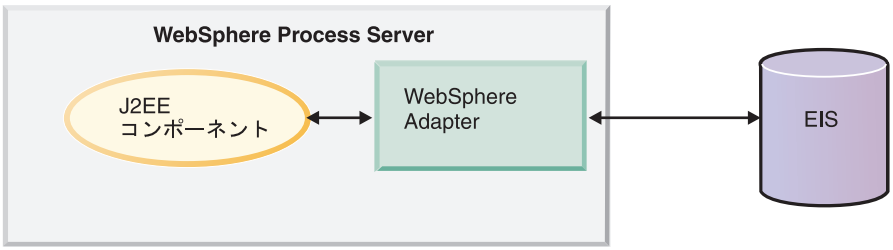
製品の概要

このトピックでは、WebSphere(R) Adapter for JDBC と、そのアーキテクチャーおよび環境要件についての基本概念を紹介します。

IBM WebSphere Adapter

IBM WebSphere Adapter には、Java 2 Enterprise Edition (J2EE) Connector Architecture (JCA) バージョン 1.5 が実装されています。WebSphere Adapter は、リソース・アダプターまたは JCA アダプターとも呼ばれ、WebSphere Process Server によってサポートされるエンタープライズ情報システム (EIS) と J2EE コンポーネント間の双方向の管理接続を可能にします。

WebSphere Adapter



IBM(R) WebSphere (R) Adapter ポートフォリオは、Java 2 Platform Enterprise Edition (J2EE) 標準に基いた新世代のアダプター群です。JCA とは、J2EE アプリケーションとエンタープライズ情報システムを統合するための標準アーキテクチャーです。これらの各システムには、呼び出す関数の識別、その入力データの指定、出力データの処理にそれぞれ固有の API があります。JCA の目的は、このような関数をコーディングするための独立した API を提供すること、データ共用を促進すること、および J2EE アプリケーションと既存およびその他の EIS を統合することです。JCA 標準では、アプリケーション・サーバー内での EIS と J2EE コンポーネント間の対話を制御する一連のコントラクトを定義することにより、上記の接続を実現します。

JCA 標準に完全に準拠している WebSphere Adapter は、WebSphere Process Server 上で稼働するように開発されました。 WebSphere Adapter は以下の処理を行います。

- WebSphere Process Server を統合する。
- WebSphere Process Server 上で稼働するアプリケーションと EIS を接続する。
- アプリケーションと EIS の間のデータ交換を可能にする。

各 WebSphere Adapter は、次の構成要素から成ります。

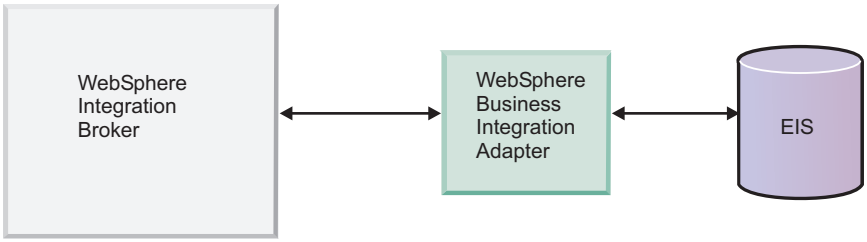
- WebSphere Process Server をサポートする (J2EE) Connector Architecture (JCA) バージョン 1.5 の実装
- 標準 Enterprise Application Archive (EAR) ファイル内にコンパイルされたビジネス・オブジェクトやその他のサービス･コンポーネント・アーキテクチャー (SCA) の成果物を生成するための Enterprise Metadata Discovery コンポーネント (このコンポーネントをエンタープライズ・サービス・ディスカバリー・ウィザードで使用して EIS をイントロスペクトする)

WebSphere Adapter は、データ・オブジェクトを表すサービス・データ・オブジェクト (SDO) を使用します。

WebSphere Adapter と WebSphere Business Integration Adapter

WebSphere Business Integration Adapter は WebSphere Adapter とは異なり、 JCA に準拠していません。

WebSphere Business Integration Adapter



図に示すように、WebSphere Business Integration Adapter は分散型です。これらは、アプリケーション・サーバーの外側に存在します。サーバー、つまり統合ブローカーは、Java Messaging Service (JMS) トランスポート層を介してこのタイプのアダプターと通信します。

WebSphere Adapter と WebSphere Business Integration Adapter との違いには、このほかに以下のものがあります。

- **接続管理** WebSphere Adapter は標準 JCA 契約に準拠して停止や開始などのライフ・サイクル・タスクを管理します。また、WebSphere Business Integration Adapter は WebSphere Adapter Framework に準拠して接続を管理します。
- **イベント通知** WebSphere Adapter の Inbound イベント通知として知られています。
- **要求処理** WebSphere Adapter の Outbound サポートとして知られています。
- **オブジェクト定義** WebSphere Adapter では、Enterprise Metadata Discovery コンポーネントを使用して、EIS のプローブやビジネス・オブジェクトおよびその他の有用な成果物の作成を行います。この Enterprise Metadata Discovery コンポーネントは WebSphere Adapter の一部です。WebSphere Business Integration Adapter では、個別の Object Discovery Agent (ODA) を使用して、EIS のプローブやビジネス・オブジェクト定義スキーマの生成を行います。

対象読者

このトピック内の情報では、WebSphere Adapter 製品のユーザーを明示し、ユーザーに必要なスキルを詳しく説明します。

アダプター・ユーザーズ・ガイドの対象読者には、各アプリケーション・コンポーネントを完全なソリューションとして組み立て、このソリューションをテストおよび配置用に準備する、データおよびアプリケーションの統合担当者が含まれています。これらのユーザーには以下の一般的なスキルが必要となります。

- ビジネス・ソリューションおよびビジネス環境をよく理解していること
- アプリケーションおよびソリューションのコンポーネントの実行時の効果的なコラボレーションを可能にするための、これらのコンポーネントに関する知識
- 異種のリレーショナル・データベース、キュー、および Web サービス間のデータベース、データ・アクセスの問題、トランザクション・モデル、および接続に関する詳細な知識
- 統合ツールについての熟知

アプリケーション統合担当者は詳細なテスト・アクティビティーも担当するため、さらに以下のスキルが必要です。

- テストおよび配置に必要なスクリプト、ツール、およびテンプレートの作成
- 統合ワークスペースの作成とシステムおよびサブシステムの統合
- Enterprise Java Beans (EJB)、ワークフロー、Web ページなどのエンティティー間の相互依存性の解決
- アプリケーションまたはソリューションの検証

データ統合担当者はさらに、アプリケーション開発者が一連のデータ・ソースにアクセスできるように設定を行います。必要なスキルは以下のとおりです。

- 統合機能またはポイント・ツー・ポイント・ゲートウェイのインストールおよび構成
- データベース・アクセス・ロジックを効果的に使用するためのプロシージャーの作成
- 外部データ・アクセス・ツール用のデータ・モデルの作成
- セキュリティー対策の実施

関連概念

69 ページの『サンプル: データベース・アプリケーションの更新』

IBM(R) WebSphere(R) Adapter for JDBC にはサンプル・ファイルが提供されています。これを使用して、データベース・アプリケーション内の特定のテーブルを更新する練習ができます。アプリケーション統合担当者用とデータ統合担当者用に、2 つの段階的なシナリオが用意されています。ユーザーの役割に応じて、ビジネス・オブジェクトの生成、アダプターの配置、J2EE(TM) アプリケーションとエンタープライズ情報システムとの通信を行うアダプターの構成の練習ができます。

タスクのロードマップ: IBM WebSphere Adapter for JDBC

IBM(R) WebSphere(R) Adapter for JDBC は、J2EE(TM) アプリケーションとデータベース・プロバイダーのエンタープライズ情報システム (EIS) 間で、ビジネス・オブジェクトのデータベース・レベルでの交換を容易にします。

タスク	説明
アダプターのインストール	Adapter for JDBC 固有のインストールの注意点についての情報 (インストール手順へのリンク付き)
アダプター・プロジェクトの作成	アダプター・オブジェクトを作成し、その後ビジネス・オブジェクトおよびサービス構成の生成と構成プロパティーの設定を行い、最後にアプリケーション・サーバーにアダプター・プロジェクトを配置します。
アダプター・プロジェクトの配置	プロジェクトを Enterprise Application Archive (EAR) ファイルにエクスポートして、このファイルをアプリケーション・サーバーにインストールします。
サーバーでのアダプターの構成	アダプターが使用するプロパティーを再構成して、特定のデータベース・アプリケーションへの通信チャネルをセットアップできます。
トラブルシューティング	イベント処理の状況を追跡するログ・ファイルをセットアップし、ログ・ファイルに取り込むエラーまたは警告のレベルを決定します。 IBM ソフトウェア・サポート支援の利用方法や、技術情報へのアクセス方法を確認します。
サンプル・データベース・アプリケーションの更新	Adapter for JDBC で提供されているサンプル・ファイルを使用して、サンプル・データベース・アプリケーションの配置、構成、および実行を練習します。複雑さのレベルが異なる 2 つのシナリオが用意されています。2 番目のシナリオにはビジネス・オブジェクトの生成が含まれています。

エンタープライズ・サービス・ディスカバリー

エンタープライズ・サービス・ディスカバリー・ウィザードを使用して、エンタープライズ情報システム (EIS) エンティティーまたはデータベース・エンティティー用ビジネス・オブジェクトを生成することができます。

エンタープライズ・サービス・ディスカバリー・ウィザードには、ビジネス・オブジェクトの青写真が用意されています。これを使用して、EIS またはデータベースのメタデータ情報を参照したり、興味のある成果物を選択したり、配置可能なサービス・オブジェクトおよびサービス記述を生成したりできます。メタデータ・ツリー構造からメタ・オブジェクト・ノードを選択することによって、EIS エンティテ

ィーまたはデータベース・エンティティー用のビジネス・オブジェクトを生成することができます。メタデータは、ビジネス・グラフやビジネス・オブジェクトを含むサービス・データ・オブジェクトに変換されます。

エンタープライズ・サービス・ディスカバリー・ウィザードを使用して、以下のアクションを実行できます。

- ビジネス・オブジェクトの生成
- ビジネス・オブジェクトへのアプリケーション固有情報の設定
- プロパティーへのアプリケーション固有情報の設定
- Inbound イベントおよび Outbound イベントへのサービス記述の提供
- Inbound イベントおよび Outbound イベントへの接続記述の提供

アーキテクチャー

WebSphere(R) Adapter for JDBC は、J2EE(TM) アプリケーションとエンタープライズ情報システム (EISs) 間に双方向の接続性を提供するリソース・アダプターです。このようなアプリケーションでは、ビジネス・オブジェクトという形でのデータの交換は、データベース・レベルで行われます。

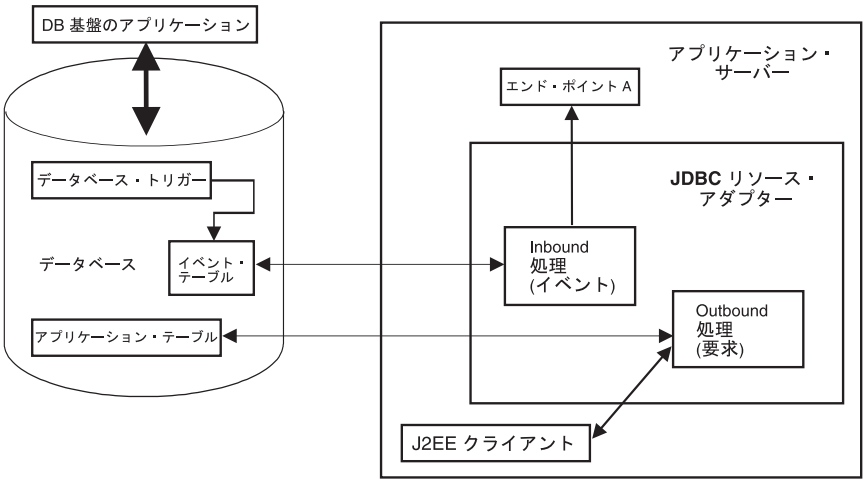
Adapter for JDBC は、データベース・プロバイダーである EIS と通信します。データベースを更新した場合、別のエンタープライズ情報システムにその更新を適用する必要が生じたり、EIS 内でデータを変更した場合、その変更をデータベースに適用する必要が生じたりする場合があります。リソース・アダプターは、データベース上で作成された、JDBC 2.0 以降の仕様をサポートする JDBC ドライバーを持つアプリケーションと統合することができます。例として、IBM(R) DB2(R)、Oracle、Microsoft(TM) SQLServer、Sybase、および Informix データベースがあります。

統合をサポートするため、リソース・アダプターは EIS から受け取った要求を処理し、データベースの更新の結果として生成されたイベントを処理します。このアダプターは、これらのイベントをアプリケーション・サーバー内のさまざまな定義済みのエンドポイントに送信します。*エンドポイント* とは、J2EE アプリケーションまたはイベントのその他のクライアント利用者を指します。データベース内の表で行われたデータ更新は、アプリケーション・サーバーに接続された、Siebel、Peoplesoft、Oracle アプリケーションなどのその他のアプリケーションに、イベント・ストアに通知されるイベント通知を通して自動的に波及させることができます。アダプターは、ビジネス・オブジェクトに指定された SQL 照会またはストアード・プロシージャーを使用してデータベース表を更新します。

Adapter for JDBC は、Java(TM) 2 Enterprise Edition (J2EE) Connector Architecture (JCA) の下で Inbound および Outbound のサポートを提供することによって、JDBC アプリケーション・プログラミング・インターフェース (API) を介してアクセス可能なデータベースの統合をサポートします。*Outbound* 操作では、ビジネス・オブジェクトは、ビジネス・オブジェクトに指定された Create、Retrieve、Update、Delete、 Retrieveall のいずれかの操作に従って処理される要求としてアダプターに渡されます。要求は、このアダプターの管理対象のデータベースに対して更新を適用させる必要のある別のEIS アプリケーションから受け取ります。これらの要求を処理すると、対応するデータベース表で行の作成、検索、更新または削除が行われます。

Inbound 操作では、データベース内のアプリケーション・テーブルでデータが変更されると、対応するイベントが、キー値のような関係のある情報と共にイベント・ストアに挿入されます。変更されたデータをキャプチャーするために、それぞれのテーブルにトリガーが設定されます。あるいは、Oracle データベースに対して提供される Oracle Change Data Capture のような他のメソッドが使用されます。Adapter for JDBC は、イベント・ストアをポーリングし、一まとまりのイベントを検索します。これらのイベントが処理されます。それぞれのイベントは JDBC ビジネス・グラフの構成に使用されます。この後、ビジネス・グラフは、特定のビジネス・オブジェクトへのサブスクリプションを持つエンドポイントにディスパッチされます。Inbound 操作では、変更後イメージのみがサポートされます。

『JDBC アダプター内での処理』の図に、Inbound 操作および Outbound 操作を示します。



JDBC アダプター内での処理

ロケールとグローバリゼーションのサポート

このアダプターは、1 バイト文字セットと 2 バイト文字セットをサポートし、メッセージ・テキストを指定された言語で配信できるようにグローバル化されています。

このアダプターは、アラビア語およびヘブライ語の双方向スクリプト・データの処理をサポートします。双方向の能力を使用するためには、双方向プロパティーを構成する必要があります。このユーザーズ・ガイドで使用される*双方向プロパティー*という用語は、双方向サポートの呼び出しを制御するプロパティーを表します。

使用しているエンタープライズ情報システム (EIS) が双方向形式を使用する場合、双方向サポート付きのすべてのプロパティーは、Windows 標準形式からターゲット EIS の双方向形式に変換されます。また、このアダプターは、そのようなデータを EIS の形式から Windows 標準形式に変換してから WebSphere Process Server に渡します。

Java 仮想マシン (JVM) 内の Java(TM) ランタイム環境では、データを Unicode 文字コード・セットで表します。 Unicode には、最もよく知られた文字コード・セット (単一バイトとマルチバイトの両方) の文字エンコードが含まれています。IBM

WebSphere Business Integration システムのほとんどのコンポーネントは Java で書かれています。そのため、WebSphere Business Integration のシステム・コンポーネント間でデータを転送するときは、ほとんどの場合文字変換の必要はありません。

エラー・メッセージや情報メッセージを適切な言語や個々の国や地域に合った形でログに記録するには、アダプターが稼働しているシステムのロケールを使用する必要があります。

WebSphere Process Server 双方向言語形式

WebSphere Process Server では、ILYNN (implicit、left-to-right、on、off、nominal) の双方向言語形式を使用します。これは Windows の双方向言語形式でもあります。その他すべての双方向言語形式は、WebSphere Process Server に導入する前に変換する必要があります。

適正な双方向言語形式にするには、5 つの属性を設定する必要があります。『双方向の属性』の表に属性および設定をリストします。

双方向の属性

文字の位置	目的	値	説明	デフォルト設定
1	スキーマの配列	I または V	暗黙 (Implicit) (論理 (Logical))、または表示 (Visual)	I
2	方向	L R C D	左から右 右から左 コンテキスト LTR コンテキスト RTL	L
3	対称スワッピング	Y または N	対称スワッピングがオンまたはオフ	Y
4	シェーピング	S N I M F B	テキストの形状を指定する (Text is shaped) テキストの形状を指定しない (Text is not shaped) 語頭形 (Initial shaping) 語中形 (Middle shaping) 語尾形 (Final shaping) 独立形 (Isolated shaping)	N
5	数字シェーピング	H、C、または N	ヒンディ語 (Hindi)、コンテキスト (Contextual)、または公称 (Nominal)	N

アダプターは、データを論理 LTR 形式に変換してから WebSphere Process Server コンポーネントに送付するのに関与します。

注: ユーザー・インターフェース (ブラウザー) のロケール設定によって、双方向言語表示および編集形式が決まります。WebSphere Process Server ユーザー・インターフェースでは、ロケール固有の形式を WebSphere Process Server のデフォルト形式に変換する必要があります。

双方向プロパティーのレベル

双方向プロパティーは、異なる複数のレベルで設定できます。これらのプロパティーの詳細およびエンタープライズ・サービス・ディスカバリー・ウィザードを使用したプロパティーの設定方法については、アダプター・プロジェクトの作成およびアダプターの構成に関するセクションを参照してください。

双方向プロパティーの編集

WebSphere Integration Developer の Business Object Editor の注釈を使用して、ビジネス・オブジェクトの双方向プロパティーおよびビジネス・オブジェクトの属性を編集することができます。注釈はビジネス・オブジェクト (*.xsd ファイル) に保管されています。詳しくは、 http://www.ibm.com/software/integration/wid の WebSphere Integration Developer Web サイトで、Business Object Editor の文書を参照してください。

また、特定の双方向プロパティーを、WebSphere Integration Developer のアセンブリー・エディターを使用して定義してから編集することもできます。実行時の双方向プロパティーの使用について詳しくは、一般技術文書、および双方向サポートについては、アダプターの技術文書を参照してください。アセンブリー・エディターについて詳しくは、http://www.ibm.com/software/integration/wid の WebSphere Integration Developer Web サイトで、アセンブリー・エディターの文書を参照してください。

ビジネス・オブジェクトの概要

この一連のトピックでは、ビジネス・オブジェクトの構造、属性プロパティー、サポートされる操作、アプリケーション固有情報、および命名規則について説明します。これには、トランザクション管理に関する情報が含まれます。

ビジネス・オブジェクトの構造

それぞれのビジネス・オブジェクトは、データベース表やビューに対応し、オブジェクト内の単純属性がそれぞれ表やビュー内の列に相当します。

単純属性 とは、String、Integer、または Date などの単一値を表す属性です。したがって、同じビジネス・オブジェクトに含まれる属性を、別々のデータベース表に格納することはできませんが、次の状況が考えられます。

- データベース表に、対応するビジネス・オブジェクトに含まれる単純属性の数よりも多くの列が含まれる場合があります。つまり、データベースの列の一部が、ビジネス・オブジェクト内に表されていません。ビジネス・オブジェクトの処理にとって必要な列のみを実際の設計に含めるようにします。
- ビジネス・オブジェクトに、対応するデータベース表に含まれる列の数よりも多くの単純属性が含まれる場合があります。つまり、ビジネス・オブジェクト内の属性の一部が、データベース表内に表されていません。データベースの列を表し

ていない属性には、アプリケーション固有情報が含まれていないでデフォルト値が設定されているか、またはストアード・プロシージャーが指定されています。

- ビジネス・オブジェクトは、複数のデータベース表にまたがるビューを表すことができます。アダプターでは、アプリケーション内でトリガーされた Create、Update、および Delete などの各イベントを処理するときに、このようなビジネス・オブジェクトを使用することができます。ただし、ビジネス・オブジェクトの要求を処理する場合には、Retrieve および RetrieveAll 要求に対してのみ、このようなビジネス・オブジェクトを使用できます。

注: ビジネス・オブジェクトが、ストアード・プロシージャー (SP) を基にしている場合は、各単純属性 (特殊な SP 属性を除く) に、アプリケーション固有情報が含まれている場合も、含まれていない場合もあります。詳細については、『ストアード・プロシージャー定義』を参照してください。

ビジネス・オブジェクトは、フラットまたは階層です。フラットのビジネス・オブジェクトの属性は、すべて単純属性であり、データベース表の 1 行を表します。*階層* ビジネス・オブジェクトという用語は、あらゆるレベルの子ビジネス・オブジェクトをすべて含む、完全なビジネス・オブジェクトを指します。*個別* ビジネス・オブジェクトという用語は、そのビジネス・オブジェクトが子オブジェクトを含んでいるか、そのビジネス・オブジェクトが子として属しているかに関係なく、1 つのビジネス・オブジェクトを指します。個別ビジネス・オブジェクトは、複数のデータベース表にまたがるビューを表すことができます。*最上位* ビジネス・オブジェクトという用語は、階層の頂点にあり、それ自体は親ビジネス・オブジェクトを持たない個別ビジネス・オブジェクトを指します。

階層ビジネス・オブジェクトは、子ビジネス・オブジェクト、子ビジネス・オブジェクトの配列、またはその組み合わせを表す属性を持ちます。そして、子ビジネス・オブジェクトも、それぞれ自身の子ビジネス・オブジェクトまたはビジネス・オブジェクトの配列を持つことができます。この関係は階層の下に向かって続きます。*単一カーディナリティー関係* は、親ビジネス・オブジェクト内の属性が 1 つの子ビジネス・オブジェクトを表すときに発生します。この場合、その属性は、その子ビジネス・オブジェクトと同じタイプです。

複数カーディナリティー関係 は、親ビジネス・オブジェクト内の属性が子ビジネス・オブジェクトの配列を表すときに発生します。この場合、その属性は、その子ビジネス・オブジェクトと同じタイプです。

アダプターでは、ビジネス・オブジェクト間での以下の関係がサポートされます。

- 単一カーディナリティー関係
- 単一カーディナリティー関係および所有権のないデータ
- 複数カーディナリティー関係

どちらのカーディナリティーのタイプでも、親ビジネス・オブジェクトと子ビジネス・オブジェクトの間の関係は、その関係を保管するビジネス・オブジェクトのキー属性に含まれるアプリケーション固有情報によって記述されます。このアプリケーション固有情報の詳細については、『ForeignKey』を参照してください。

関連資料

29 ページの『ストアード・プロシージャー定義』
ストアード・プロシージャーは動詞レベルで定義されます。ストアード・プロシージャー定義はそれぞれ、エレメント StoredProcedureType、StoredProcedureName、ResultSet、および Parameters から構成されます。

13 ページの『ビジネス・オブジェクト属性のプロパティー』
ビジネス・オブジェクトのアーキテクチャーでは、属性に適用されるさまざまなプロパティーが定義されています。このセクションでは、アダプターによるこれらのプロパティーの解釈方法を説明します。

ビジネス・オブジェクトの単一カーディナリティー関係

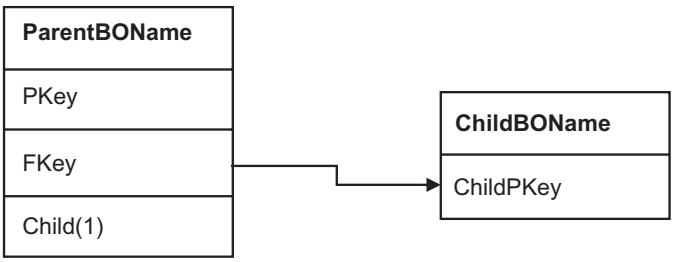
このトピックでは、このアダプターがサポートする単一カーディナリティー関係について説明します。

単一カーディナリティー関係

通常、単一カーディナリティーの子ビジネス・オブジェクトを含むビジネス・オブジェクトには、関係を表すための属性が 2 つ以上含まれます。1 つの属性のタイプは、子ビジネス・オブジェクトのタイプと同じになります。もう 1 つの属性は、子の基本キーを、外部キーとして親に格納するための単純属性です。親には、子に含まれる基本キー属性と同数の外部キー属性が含まれます。

関係を設定する外部キーが親に保管されるため、各親には特定のタイプの子ビジネス・オブジェクトを 1 つだけ格納できます。

『一般的な単一カーディナリティー関係 (Typical single-cardinality relationship)』の図に、一般的な単一カーディナリティー関係を示します。この例では、ParentBOName ボックス内の FKey は、子の基本キーを含む単純属性であり、同様に、ParentBOName ボックス内にある Child(1) は、子ビジネス・オブジェクトを表す属性です。



典型的な単一カーディナリティー関係

親ビジネス・オブジェクトは、単一カーディナリティーの OWNERSHIP の子と、単一カーディナリティーの NOOWNERSHIP の子を持つことができます。

単一カーディナリティー関係および所有権のないデータ

通常、各親ビジネス・オブジェクトは、その親ビジネス・オブジェクトに含まれる子ビジネス・オブジェクトの内部のデータを所有 しています。例えば、各 Customer ビジネス・オブジェクトが Address ビジネス・オブジェクトを 1 つ含んでいる場合に、新しいカスタマーが作成されると、Customer テーブルと Address テーブルの両方に新しい行が 1 行挿入されます。挿入された新しい住所は、その新し

いカスタマーに固有です。同様に、Customer テーブルからカスタマーを削除すると、Address テーブルからそのカスタマーの住所も削除されます。

ただし、複数の階層ビジネス・オブジェクトに同一のデータが含まれており、どのビジネス・オブジェクトもそのデータを所有していない場合があります。例えば、Address ビジネス・オブジェクトに *StateProvince[1]* 属性があり、これが単一カーディナリティーの *StateProvince* 参照テーブルを表しているとします。この参照テーブルは、ほとんど更新されることがないものであり、住所データからは独立して保守されています。このため、住所データの作成または変更により、この参照テーブル内のデータが影響を受けることはありません。アダプターは、既存の都道府県名を検出するか、検出に失敗するかのいずれかです。この参照テーブル内の値を追加または変更することはありません。

複数のビジネス・オブジェクトに同一の単一カーディナリティーの子ビジネス・オブジェクトが含まれている場合、各親ビジネス・オブジェクトの外部キー属性では、関係が `NOOWNERSHIP` に指定されていなければなりません。アプリケーション・サーバーからアダプターに、階層ビジネス・オブジェクトが Create、Delete、または Update 要求とともに送信された場合、アダプターは所有関係にない単一カーディナリティーの子を無視します。アダプターは、このようなビジネス・オブジェクトに対しては、Retrieve 操作のみを実行します。このような単一カーディナリティーのビジネス・オブジェクトの検索に失敗した場合、アダプターはエラーを戻して処理を停止します。

非正規化データおよび所有権のないデータ

所有関係を伴わない包含関係には、静的参照テーブルの使用を容易にするだけでなく、正規化データと非正規化データを同期化するという別の機能があります。

正規化データから非正規化データへの同期: 関係を `NOOWNERSHIP` に指定すると、正規化アプリケーションから非正規化アプリケーションへの同期化を行うときに、データを作成または変更することができます。例えば、正規化されたソース・アプリケーションが、A と B という 2 つのテーブルにデータを格納するものとします。また、非正規化されている宛先アプリケーションでは、1 つのテーブルにすべてのデータが格納され、エンティティー A のそれぞれにエンティティー B のデータが重複して格納されるものとします。

この例では、テーブル B のデータの変更をソース・アプリケーションから宛先アプリケーションに同期化するには、テーブル B のデータが変更されるたびにテーブル A のイベントを起動する必要があります。さらに、テーブル B のデータはテーブル A に重複して格納されているので、テーブル A の行ごとに、テーブル B で変更されたデータが含まれるビジネス・オブジェクトを送信しなければなりません。

注: 非正規化されたテーブルを更新する場合、1 行を更新した結果複数の行が変更されることがないように、レコードごとに固有キーを持たせるようにしてください。そのようなキーが存在しない場合、 Adapter for JDBC では複数レコードが更新されたことを示すエラーが発生します。

非正規化データから正規化データへの同期: 非正規化されているソース・アプリケーションから正規化されている宛先アプリケーションにデータを同期化する場合、

アダプターは、正規化されているアプリケーションに含まれる所有関係にないデータに関しては、作成、削除、または更新を行いません。

正規化されているアプリケーションにデータを同期化する場合、アダプターは、所有関係にない単一カーディナリティーの子をすべて無視します。そのような子のデータを作成、除去、または変更するには、データを手動で処理する必要があります。

ビジネス・オブジェクトの複数カーディナリティー関係

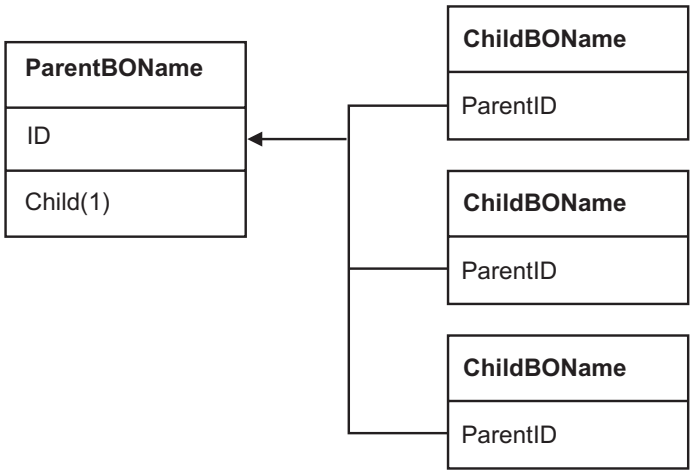
アダプターは、複数カーディナリティー関係をサポートしています。

通常、子ビジネス・オブジェクトの配列を含むビジネス・オブジェクトには、関係を表すための属性が 1 つだけ含まれています。この属性のタイプは、子ビジネス・オブジェクトと同じタイプの配列です。親が複数の子を含むようにするため、関係を設定する外部キーは子に格納されます。

したがって、どの子にも、親の基本キーを外部キーとして含む単純属性が 1 つ以上存在します。子には、親に含まれる基本キー属性と同数の外部キー属性が含まれます。

関係を設定する外部キーが子に保管されるので、それぞれの親は、1 つ以上の子を持つことができます (子を持たないことも可能です)。

『ビジネス・オブジェクトの複数カーディナリティーの関係』の図に、複数カーディナリティーの関係を示します。この例では、3 つの ChildBOName ボックス内の parentId は、親の基本キーを含む単純属性であり、ParentBOName ボックス内にある Child1 は、子ビジネス・オブジェクトの配列を表す属性です。

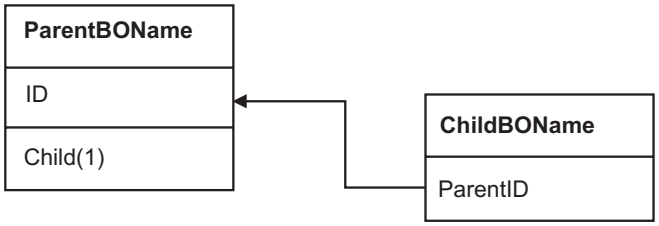


ビジネス・オブジェクトの複数カーディナリティーの関係

複数カーディナリティーの関係は、N=1 の関係である場合があります。アプリケーションによっては、親子関係を親ではなく子に格納するように、子エンティティーを 1 つ格納するものがあります。つまり、子には、親の基本キーに格納されている値と同一の値の外部キーが格納されます。

このタイプの関係がアプリケーションで使用されるのは、子のデータが親から独立して存在しておらず、親を介してのみそのデータにアクセスできる場合です。この

ような子のデータでは、子とその外部キー値を作成するために、親とその基本キー値があらかじめ存在していなければなりません。『N=1 の場合の複数カーディナリティーの関係』の図でこのタイプの関係を示します。



N=1 の場合の複数カーディナリティーの関係

ビジネス・オブジェクト属性のプロパティー

ビジネス・オブジェクトのアーキテクチャーでは、属性に適用されるさまざまなプロパティーが定義されています。このセクションでは、アダプターによるこれらのプロパティーの解釈方法を説明します。

次の「属性プロパティー」というタイトルの表に、これらのプロパティーの解釈と設定値を示します。

属性プロパティー

プロパティー	解釈と設定値
Cardinality	子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を表す各ビジネス・オブジェクト属性は、それぞれ、値 1 または 複数 (n) のカーディナリティーを持ちます。また、子ビジネス・オブジェクトを表す属性はすべて、ContainedObjectVersion プロパティー (子のバージョン番号を指定) と Relationship プロパティー (値 Containment を指定) を持ちます。
Foreign Key	カーディナリティーが *n* の子ビジネス・オブジェクトの配列が検索されると、SELECT ステートメントの WHERE 文節で外部キーが使用されます。 **注:** アダプターでは、子ビジネス・オブジェクトを表す属性を外部キーとして指定することについては、サポートしていません。 RetrieveAll 動詞は、キーおよび外部キーの使用を指定変更します。

プロパティー	解釈と設定値
Key	どのビジネス・オブジェクトでも、1 つ以上の単純属性がキーに指定されなければなりません。 **注:** アダプターでは、子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を表す属性をキー属性として指定することについては、サポートしていません。 単純属性のキー・プロパティーを `true` に設定すると、アダプターは、ビジネス・オブジェクトの処理中に生成する SELECT および、UPDATE の SQL ステートメントの WHERE 文節にその属性を追加します。RetrieveAll 動詞は、キーおよび外部キーの使用を指定変更します。
Name	このプロパティーは、属性が単純属性の場合は、属性の固有の名前。属性が子ビジネス・オブジェクトの場合は、ビジネス・オブジェクトの名前を表します。
Required	属性が値を含む必要があるかどうかを指定します。カーディナリティーが単一 (1) のコンテナーに対して、このプロパティーが `true` に設定されている場合、アダプターでは、その親ビジネス・オブジェクトが、この属性に対応する子ビジネス・オブジェクトを含んでいる必要があります。Create、Update、および Delete 操作でアダプターに渡されるビジネス・オブジェクトは、子ビジネス・オブジェクトも含んでいなければなりません。単純属性のカーディナリティーは単一 (1) で、コンテナー属性のカーディナリティーは複数 (n) です。ビジネス・オブジェクトが必須属性に対して有効な値またはデフォルト値を持っていないと、アダプターでは Create 操作が失敗します。このオブジェクトに対するデータベースからの検索時に使用可能なデータがない場合も、Create 操作は失敗します。
Type	単純属性の場合は属性の型 (Integer、String、Date、Boolean、Double、または Float など)。子ビジネス・オブジェクトの場合はビジネス・オブジェクトのタイプ。アダプターはサポートされない型の属性を検出すると、値を引用符で囲み、その値を文字データとして処理します。

関連概念

8 ページの『ビジネス・オブジェクトの構造』
それぞれのビジネス・オブジェクトは、データベース表やビューに対応し、オブジェクト内の単純属性がそれぞれ表やビュー内の列に相当します。

サポートされる操作

アダプターは、Inbound 操作と Outbound 操作を実行します。 Outbound 操作では、変更後イメージと差分がサポートされます。アダプターは、Inbound 操作に対して変更後イメージだけをサポートします。サポートされる操作を以下にリストします。

Outbound 操作

アダプターは、ビジネス・オブジェクトによって伝えられる情報の量および目的に関連する、変更後イメージと差分という 2 つのビジネス・オブジェクト・スタイルをサポートします。*変更後イメージ* とは、ビジネス・オブジェクトに対するすべての変更が行われた後の、ビジネス・オブジェクトの状態のことです。*差分* とは、Update 操作で使用される、キー値および変更対象のデータのみを含むビジネス・オブジェクトのことです。アダプターは、次の操作に対して、変更後イメージと差分の両方をサポートします。

- Create
- Update
- Delete
- ApplyChanges
- Retrieve
- RetrieveAll

動詞は、変更後イメージ・ビジネス・オブジェクトに対してのみ定義できます。*動詞* にはビジネス・オブジェクトの状態が反映され、*操作* にはアダプターによって実行される操作が反映されます。変更後イメージに対してサポートされる最上位の動詞は、次のとおりです。

- Create
- Update
- Delete
- UpdateWithDelete

Inbound 操作

アダプターは、Inbound 操作に対して変更後イメージだけをサポートします。アダプターは、更新用のビジネス・オブジェクトを受信すると、そのビジネス・オブジェクトがデータの更新後の正しい状態を表しているものと想定します。次の操作がサポートされています。

- Create
- Update
- Delete

トランザクション管理

Adapter for JDBC は、ローカル・トランザクションと XA トランザクションの両方をサポートします。

このアダプターでは、トランザクションとは、バックエンド・データベースまたはエンタープライズ情報システム (EIS) との独立した相互作用です。トランザクションは、アトミックな単位で実行する、データベースへの複数の操作から構成されます。これらの操作は、データベースの他のクライアント・アプリケーションから同時に実行される操作の影響は受けません。

Adapter for JDBC がトランザクションをサポートするのは、バックエンド・データベースがトランザクションをサポートする場合のみです。サポートされるトランザクションのタイプは、ローカル・トランザクションと XA トランザクションです。

- *ローカル・トランザクション* では、指定されたクライアント・アプリケーションが、データベースを使用したトランザクションの開始および終了を定義します。このトランザクションでは、1 フェーズ・コミット・プロトコルを使用します。
- *XA トランザクション* では、トランザクションは複数の異種データベースを使用します。このトランザクションでは、グローバル・プロトコルまたは 2 フェーズ・コミット・プロトコルを使用します。

このアダプターは、IBM(R) DB2(R) および Oracle データベース用 XA トランザクションをサポートします。

XA トランザクションでは、XADataSourceName および XADatabaseName プロパティーを使用してください。これらのプロパティーについて詳しくは、『参照』セクションの『J2C 接続ファクトリー・プロパティー』を参照してください。

注: DB2 データベースを使用している場合、XADatabaseName プロパティーを構成する必要があります。

『参照』セクションには、ローカル・トランザクションおよび XA トランザクションで使用されるプロパティー値の例を示しています。

関連資料

87 ページの『J2C 接続ファクトリー・プロパティー』
J2C 接続ファクトリー・プロパティーは、ターゲットのエンタープライズ情報システム (EIS) インスタンスを構成するのに使用します。これらのプロパティーは、Outbound 処理に 影響を与え、J2EE(TM) Connector Architecture 仕様の ManagedConnectionFactory インターフェースに対応しています。

82 ページの『参照』
このセクションでは、構成、接続、およびオブジェクト選択のプロパティーについて説明します。

Inbound サポート

Adapter for JDBC は、非同期イベント送達を行う Inbound イベント管理をサポートします。

非同期イベント送達は、イベント・ストアと呼ばれるイベント・テーブルと、ステージング・テーブル (イベント配布テーブル) を使用して実行されます。ユーザー・テーブル内で変更を行うと、アプリケーションはエンタープライズ情報システム (EIS) 内のイベント・テーブルにデータを取り込みます。ユーザー・テーブルにトリガーを配置することによって更新を行います。ユーザー・テーブルは、ユーザー・テーブルへの更新に対応してイベント・テーブルにイベントを記録します。

ステージング・テーブルは Cloudscape(TM) のような XA 準拠データベースの一部として配置されます。Cloudscape(TM) は WebSphere(R) Application Server バージョン 6.0 に含まれています。EIS からイベントが取得されると、アクティブなエンドポイントごとにイベントへの参照がステージング・テーブルに書き込まれます。ステージング・テーブル内の各イベントは、アプリケーション・サーバーによって制御される固有の XA トランザクションの一部として、対応するエンドポイントに配信されます。エンドポイントがイベントを消費し、トランザクションがコミットされると、ステージング・テーブル内のイベントへの参照は、そのエンドポイントに対応するもののみ除去されます。

イベント・テーブルは以下の表に記述されています。

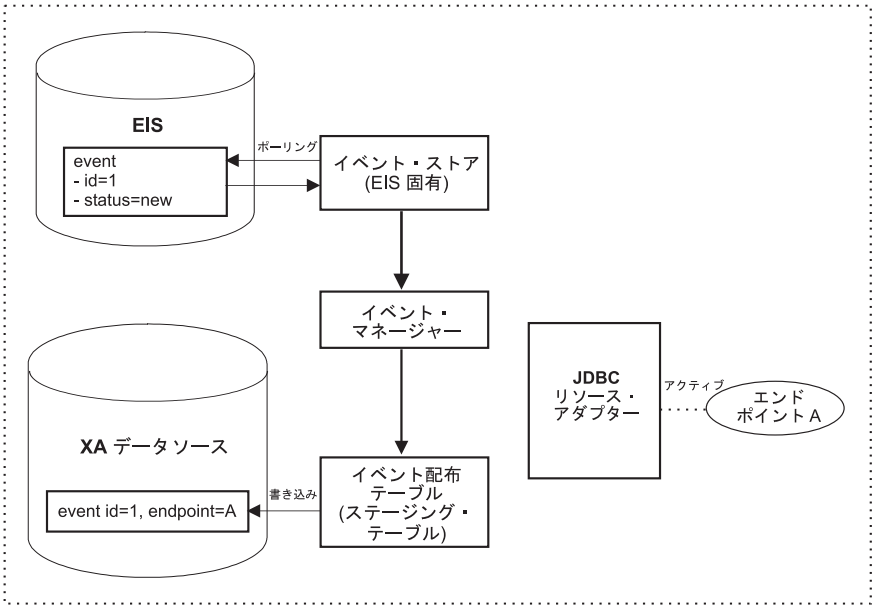
イベント・テーブル

名前	型	説明
event_id	Number	テーブル用の基本キーである固有のイベント ID
object_key	String	ビジネス・オブジェクト用のキーを含む、区切り文字で区切られたストリング (NULL 以外)
object_name	String	ビジネス・オブジェクトの名前 (NULL 以外)
object_function	String	イベントに対応した操作 (Delete、 Create、Update など。NULL 以外)
event_priority	Number	NULL 以外
event_time	Date	イベントが生成された日時
event_status	Number	進行中、成功、失敗など (NULL 以外)
event_comment	String	説明

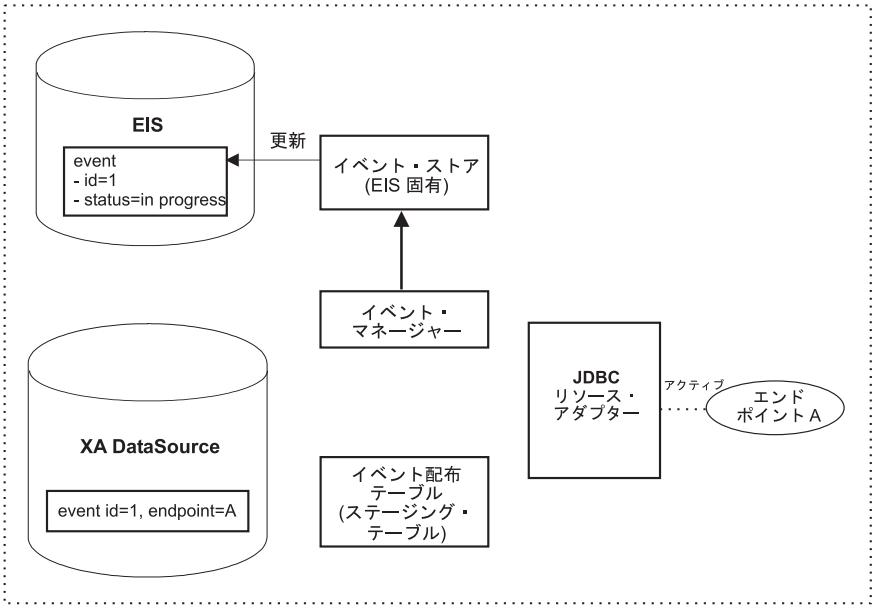
Adapter for JDBC におけるイベント処理

Adapter for JDBC によるイベント処理の全フローは、以下の 4 つの図に示す 4 つのステップに分けて表されます。

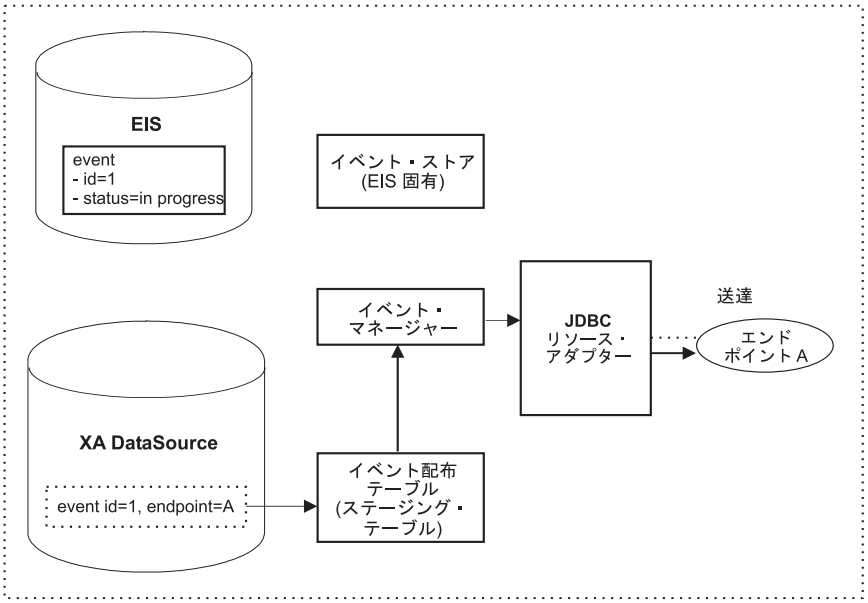
最初の図は 1 番目のステップで、ここでは EIS 内のイベントが検知され、アクティブなエンドポイントごとにステージング・テーブルにレコードが追加されています。



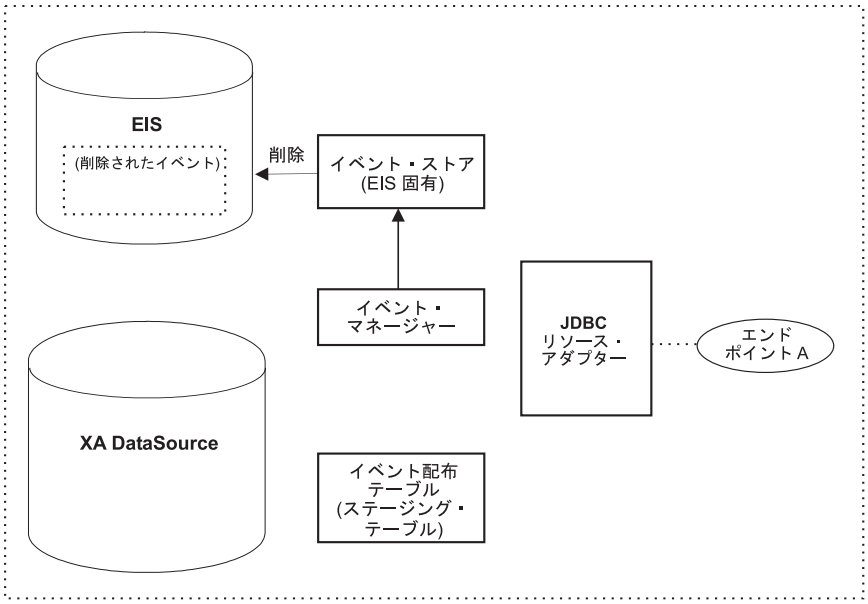
次の図は 2 番目のステップで、ここでは EIS 内のイベントが「進行中 (in-progress)」にマークされます。



次の図は 3 番目のステップで、ここではステージング・テーブルを送達が必要なエンドポイントのガイドとして使用することによって、イベントをすべてのアクティブなエンドポイントにパブリッシュします。同時に、イベントは、各エンドポイントとステージング・テーブル内のその特定のレコードを含む個々の XA トランザクションの一部としてステージング・テーブルから削除されます。実際には、各エンドポイントはイベントの受信および処理に関しては他のエンドポイントから独立した形で行います。



4 番目のステップでは、いったんすべてのエンドポイントがイベントを受信すると、ステージング・テーブル内の残った参照の不足を基にして、EIS から元のイベントが削除されます。イベントごとにステップ 1 から 4 までを繰り返します。



Outbound 操作のビジネス・オブジェクト処理

この一連のトピックでは、アダプターがビジネス・オブジェクトを作成、検索、更新、または削除する際に使用する手順について概説します。アダプターでは、階層ビジネス・オブジェクトを再帰的に処理します。つまり、個別のビジネス・オブジェクトがすべて処理されるまで、同じステップを子ビジネス・オブジェクトごとに実行します。

ビジネス・オブジェクトの比較

ビジネス・オブジェクトの動詞の操作のさまざまなポイントで、アダプターは 2 つのビジネス・オブジェクトを比較し、それらが同一であるかどうかを確認します。例えば、Update 操作時には、アダプターは、ビジネス・オブジェクトの配列内に特定のビジネス・オブジェクトが存在するかどうかを判定します。アダプターは、この検査を行うため、その特定のビジネス・オブジェクトを配列内のビジネス・オブジェクトのそれぞれと比較します。2 つのビジネス・オブジェクトが同一であるのは、次の 2 つの条件が満たされている場合です。

- 比較されている 2 つのビジネス・オブジェクトのタイプが一致していること。例えば、Customer ビジネス・オブジェクトと Contact ビジネス・オブジェクトの属性がすべて一致している場合でも、これらのビジネス・オブジェクトが同一であると見なされることはありません。
- 2 つのビジネス・オブジェクトの、対応するキー属性のすべてに、同一の値が格納されていること。キー属性が両方のビジネス・オブジェクトに設定されていない場合、アダプターはそれらを同一とみなしますが、キー属性が片方のビジネス・オブジェクトに設定されていてもう片方には設定されていない場合、それらのビジネス・オブジェクトは同一ではありません。

Create 操作

階層ビジネス・オブジェクトの場合は、Create 操作によってビジネス・オブジェクトが再帰的に全探索され、各テーブルに対応する行が作成されます。

以下で、詳しく説明します。

1. Create 操作は、所有関係を伴う単一カーディナリティーの各子ビジネス・オブジェクトを、データベース内に再帰的に挿入します。つまり、アダプターは、子ビジネス・オブジェクトおよびその子孫にあたるビジネス・オブジェクトのすべてを作成します。

   ビジネス・オブジェクト定義上、ある属性がある単一カーディナリティーの子ビジネス・オブジェクトを表すものとされている場合に、その属性が空であると、アダプターはその属性を無視します。ただし、ビジネス・オブジェクト定義により、その属性が子を表すことが必要であるにもかかわらず、子を表していない場合には、アダプターはエラーを戻して処理を停止します。

2. Create 操作は、所有関係を伴わない単一カーディナリティーの各子ビジネス・オブジェクトの有無を検索し、確認します。子がデータベース内に存在しないことを示して、Retrieve 操作が失敗した場合、アダプターはエラーを戻して処理を停止します。Retrieve 操作が成功した場合、アダプターは子ビジネス・オブジェクトを再帰的に更新します。

   注: アプリケーションのデータベースに子ビジネス・オブジェクトが存在する場合に、このアプローチが正しく機能するには、子ビジネス・オブジェクト内の基本キー属性の相互参照が、Create 操作時に正しく行われる必要があります。アプリケーション・データベースに子ビジネス・オブジェクトが存在しない場合、基本キー属性は設定してはいけません。

3. 最上位ビジネス・オブジェクトを、データベース内に次のように挿入します。
   a. 最上位ビジネス・オブジェクトの外部キー値を、対応する単一カーディナリティーの関係にある子ビジネス・オブジェクトの基本キー値に設定します。

子ビジネス・オブジェクトの値は、データベース・シーケンスまたはカウンター、あるいはデータベース自体によって、子の作成時に設定される場合があります。そのため、このステップでは、アダプターが親をデータベースに挿入する前に、親の外部キー値を正しいものにします。

b. データベースによって自動的に設定される属性のそれぞれに対して、新しい固有 ID 値を生成します。データベース・シーケンスまたはカウンターの名前は、属性のアプリケーション固有情報に格納されます。属性にデータベース・シーケンスまたはカウンターが関連付けられている場合、アダプターによって生成された値により、アプリケーション・サーバーから渡された値が上書きされます。データベース・シーケンスまたはカウンターの指定の詳細については、『単純属性のアプリケーション固有情報』の UID=AUTO を参照してください。

c. 最上位ビジネス・オブジェクトをデータベース内に挿入します。

4. 複数カーディナリティーの子ビジネス・オブジェクトのすべてを、次のように処理します。

a. それぞれの子の外部キー値を、親に含まれる対応する基本キー属性の値を参照するように設定します。親の基本キー値は、親の作成時に生成されている可能性があります。そのため、ここでは、アダプターが子をデータベースに挿入する前に、それぞれの子の外部キー値を正しいものにします。

b. 複数カーディナリティーの子ビジネス・オブジェクトのすべてを、データベースに挿入します。

Retrieve 操作

このトピックでは、階層ビジネス・オブジェクトを検索するためにアダプターが使用する手順について説明します。

アダプターは次のように検索操作を実行します。

1. 受信した最上位ビジネス・オブジェクトから、すべての子ビジネス・オブジェクトを削除します。言い換えると、子のない最上位ビジネス・オブジェクトのコピーを作成します。
2. 最上位ビジネス・オブジェクトを、データベース内で検索します。
   - 検索の結果戻された行が 1 つの場合、アダプターは処理を継続します。
   - 検索の結果戻された行がない場合 (目的の最上位ビジネス・オブジェクトがデータベース内に存在しないことを意味します)、アダプターはエラー `RecordNotFoundException` を戻します。
   - 検索の結果戻された行が複数ある場合、アダプターはエラーを戻します。
3. 複数カーディナリティーの子ビジネス・オブジェクトのすべてを、再帰的に検索します。

   注: アダプターは、ビジネス・オブジェクトの配列を取り込むときに、一意性を保証しません。一意性の保証は、データベース側で行われなければなりません。データベースから戻された子ビジネス・オブジェクトに重複があると、アダプターは、それらの重複する子を戻します。
4. 子ビジネス・オブジェクトが所有関係にあるかどうかに関係なく、各単一カーディナリティーの子を再帰的に検索します。

注: 単一カーディナリティーの子ビジネス・オブジェクトはすべて、ビジネス・オブジェクト内での出現順序に従って、親ビジネス・オブジェクトが処理される前に処理されます。子オブジェクトに対する所有関係の有無は、処理シーケンスを決定しませんが、処理のタイプは決定します。

RetrieveAll 操作

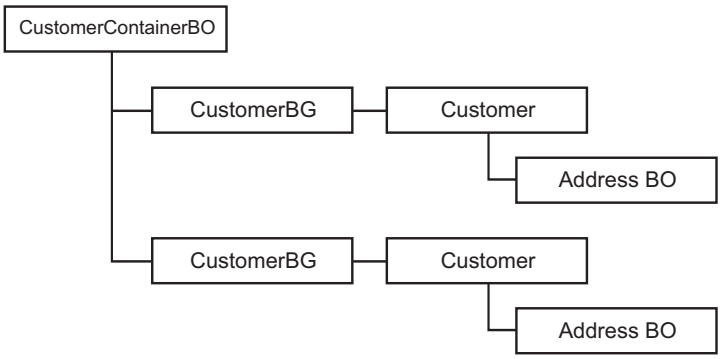
この操作によって、アダプターはデータベースからビジネス・オブジェクトの配列を検索できます。

着信ビジネス・オブジェクト内に取り込まれているすべてのキー属性および非キー属性によって、選択基準が決まります。選択した属性によっては、アダプターは、データベースから最上位ビジネス・オブジェクトの複数の行を検索する場合もあります。着信ビジネス・オブジェクト内に属性が取り込まれていない場合は、データベース内のそれぞれのテーブルからすべての行が検索されます。

アダプターは次のステップを実行してビジネス・オブジェクトの配列を検索します。

1. データベースから検索された行ごとに、アダプターは最上位ビジネス・グラフを構成し、検索された行をすべて使用してビジネス・グラフのコンテナーを作成します。コンテナー・ビジネス・グラフの名前は、*BOName + ContainerBG* です。
2. アダプターは、検索操作を使用してコンテナー内の各ビジネス・グラフを検索します。

次の図は、RetrieveAll 操作で戻されるビジネス・オブジェクトの構造を示しています。



RetrieveAll 操作によって、次のエラーが発生する可能性があります。

- 入力オブジェクト内に取り込まれたビジネス・オブジェクトが EIS に存在しない場合、アダプターはエラー `RecordNotFoundException` を戻します。
- EIS 内のヒット数が、対話仕様で定義された ResultSetLimit の値を超えると、アダプターはエラー MatchesExceededLimitException を戻します。 MatchCount プロパティーには、アダプターが EIS 内に保持するヒットの実際の数が含まれているため、制限値を高くしたり、検索を適度に詳細化することができます。

  注: ResultSetLimit を大きい数に設定すると、戻されるビジネス・オブジェクトのサイズと数によってはメモリー不足に関連する問題が発生する場合があります。

- EIS によってリカバリー不能エラーが報告された場合、アダプターはエラー EISSystemException を戻します。

Update 操作

Update 操作は、着信ビジネス・オブジェクトを、最上位の着信ビジネス・オブジェクトで指定された基本キーを使用してデータベースから検索されたビジネス・オブジェクトと比較することによって実行されます。

アダプターでは、階層ビジネス・オブジェクトの更新時に、以下のステップを実行します。

1. ソース・ビジネス・オブジェクトの基本キー値を使用して、データベース内の対応するエンティティーを検索します。検索されたビジネス・オブジェクトは、データベース内のデータの現在の状態を正確に表したものです。

   検索が失敗した場合 (最上位ビジネス・オブジェクトがデータベース内に存在しないことを意味します)、アダプターは `RecordNotFoundException` エラーを戻し、更新は失敗します。

   検索に成功した場合、アダプターは、検索されたビジネス・オブジェクトをソース・ビジネス・オブジェクトと比較して、どの子ビジネス・オブジェクトに関してデータベースに変更を加える必要があるかを判別します。ただし、アダプターはソース・ビジネス・オブジェクトの単純属性の値と検索されたビジネス・オブジェクトの単純属性の値を比較しません。アダプターは、非キーの単純属性すべての値を更新します。

   最上位ビジネス・オブジェクトのすべての単純属性がキーを表している場合、アダプターはその最上位ビジネス・オブジェクト用の更新照会を生成できません。この場合、アダプターは、警告を記録してからステップ 2 に進みます。
2. 最上位ビジネス・オブジェクトの子のうち、単一カーディナリティーのものすべてを再帰的に更新します。

   ビジネス・オブジェクト定義上、ある属性がある子ビジネス・オブジェクトを表すことが必須である場合には、その子ビジネス・オブジェクトがソース・ビジネス・オブジェクトと検索されたビジネス・オブジェクトの両方に存在している必要があります。存在しない場合、Update 操作は失敗し、アダプターはエラーを戻します。

   アダプターでは、所有関係にある単一カーディナリティーの子を、次のいずれかの方法で処理します。
   - ソース・ビジネス・オブジェクトおよび検索したビジネス・オブジェクトの両方に子が存在する場合、アダプターは、データベース内の既存の子を更新するのではなく、既存の子を削除して新規の子を作成します。
   - その子がソース・ビジネス・オブジェクトには存在するにもかかわらず、検索されたビジネス・オブジェクトには存在しない場合、アダプターはデータベース内にその子を再帰的に作成します。
   - その子が検索されたビジネス・オブジェクトには存在するにもかかわらず、ソース・ビジネス・オブジェクトには存在しない場合、アダプターはデータベース内のその子を再帰的に削除します。

所有関係にない単一カーディナリティーの子に関しては、アダプターは、ソース・ビジネス・オブジェクトに存在するそのような子のすべてを、データベースから検索しようとします。アダプターは、子の検索に成功すると、その子ビジネス・オブジェクトにデータを読み込みますが、更新は行いません。これは、所有関係にない単一カーディナリティーの子はアダプターによって変更されることがないためです。

3. 検索されたビジネス・オブジェクトのすべての単純属性を更新します。ただし、ソース・ビジネス・オブジェクト内の対応する属性が指定されていない場合を除きます。

   更新されるビジネス・オブジェクトは一意である必要があるため、アダプターは、結果として 1 行のみが処理されることを確認します。複数の行が戻されている場合、アダプターはエラーを戻します。

4. 検索されたビジネス・オブジェクトの複数カーディナリティーの子のそれぞれを、次のいずれかの方法で処理します。
   - その子がソース・ビジネス・オブジェクトの配列と検索されたビジネス・オブジェクトの配列の両方に存在する場合、アダプターはデータベース内でその子を再帰的に更新します。
   - その子がソース・ビジネス・オブジェクトの配列には存在しても、検索されたビジネス・オブジェクトの配列には存在しない場合、アダプターはデータベース内でその子を再帰的に作成します。
   - その子が検索されたビジネス・オブジェクトの配列には存在しても、ソース・ビジネス・オブジェクトの配列には存在しない場合、アダプターはデータベースからその子を再帰的に削除します。ただし、親に含まれている、その子を表す属性のアプリケーション固有情報で、KeepRelationship が `true` に設定されている場合を除きます。この場合、アダプターは、データベースからその子を削除しません。

UpdateWithDelete 操作

これは、Update 操作よりも高いパフォーマンスを実現するために使用できる Update 操作の特別な形式です。

UpdateWithDelete には ChangeSummary が必要です。*ChangeSummary* では、ビジネス・オブジェクト・レベルの作成と削除を組み込む必要があります。ChangeSummary は何を実行する必要があるかを示すので、アダプターは、EIS から既存のエンティティーを検索するオーバーヘッドもなく比較を実行する必要もなく、操作を実行できます。

ChangeSummary が空の場合、アダプターは要求に対して処理を行いません。

DeltaUpdate 操作

InteractionSpec 内の操作が ApplyChange で、ビジネス・グラフ内に動詞が存在しない場合、アダプターは DeltaUpdate 操作を実行します。アダプターは、ChangeSummary をインスペクションして入力階層内の各ビジネス・オブジェクトの操作を識別し、識別された操作を実行します。

次に示すように、DeltaUpdate 操作は Update 操作とは異なります。

- DeltaUpdate 操作では、更新の前に Retrieve 操作は実行されません。

- 着信ビジネス・オブジェクトとデータベース内のビジネス・オブジェクトの比較が行われません。
- 子はすべて、各子オブジェクトに設定されている動詞に基づいて処理されます。子に動詞が設定されていない場合、アダプターはエラーを戻します。

アダプターは、DeltaUpdate による階層ビジネス・オブジェクトの更新時に、以下のステップを実行します。このステップでは、ChangeSummary から、オブジェクトの変更内容のみが処理されます。

1. 親オブジェクトの子のうち、単一カーディナリティーのものすべてを再帰的に処理します。ビジネス・オブジェクト仕様で必須にマークされている子は、Inbound オブジェクトに存在していなければなりません。存在しない場合、DeltaUpdate 操作は失敗し、アダプターがエラーを戻します。
2. 親に含まれる外部キー値のうち、単一カーディナリティーの子の属性を参照するものすべてを、それぞれ対応する子の値に設定します。この処理が必要なのは、これ以前のステップで、単一カーディナリティーの子がデータベースに追加され、新しいシーケンス値が生成されている可能性があるためです。
3. 現在処理中のオブジェクトを、SQL Update ステートメントまたはストアード・プロシージャーを使用して更新します。個々のビジネス・オブジェクトのすべての単純属性が更新されます。アダプターは Update ステートメントにどの属性を追加しなければならないかを判断するのに、プロパティー・レベルの変更を使用しません。すべて更新されます。更新されているオブジェクトは固有であるため、アダプターは結果として 1 行のみが処理されていることをチェックします。複数の行が処理される場合、エラーが戻されます。
4. 現在のオブジェクトのカーディナリティー N のすべての子にある、親の属性を参照する外部キー値をすべて、対応する親の値に設定します。通常、これらの値はデータ・マッピング時に既に相互参照されています。ただし、これはカーディナリティーが N のコンテナーに含まれる新しい子には該当しない場合があります。このステップにより、カーディナリティーが N の子すべての外部キー値が正しい値になってから、それらの子の更新が行われることが徹底されます。
5. 現在のオブジェクトの、カーディナリティーが N のコンテナーをすべて更新します。

   子オブジェクトが処理されるときには、それぞれの子の動詞が取得され、適切な操作が実行されます。DeltaUpdate で子に対して許可される操作は、Create、Delete、および Update です。

   - Create 動詞が子で検出された場合、それが所有関係にある子であれば、検出された子がデータベース内に作成されます。所有関係にない子に関しては、検索により、データベースに存在するかどうかが確認されます。
   - Delete 動詞が子で検出された場合、子は削除されます。
   - Update 動詞が子で検出された場合、子はデータベース内で更新されます。

Delete 操作

Delete 操作は、データベースからの着信ビジネス・オブジェクトのプルーニングと、その後の完全なビジネス・オブジェクトの検索によって実行されます。Delete 操作は、その後階層内の各ビジネス・オブジェクトについて再帰的に適用されます。

Delete 操作では、オブジェクトのアプリケーション固有情報のステータス列名の値に応じて、物理削除と論理削除がサポートされます。ステータス列名の値が定義されている場合、アダプターは論理削除操作を実行します。ステータス列名の値が定義されていない場合、アダプターは物理削除操作を実行します。

物理削除の場合、アダプターは次の処理を行います。

- 複数カーディナリティーの子ビジネス・オブジェクトすべてを再帰的に削除します。
- 最上位ビジネス・オブジェクトを削除します。
- 所有関係にある単一カーディナリティーの子ビジネス・オブジェクトすべてを、再帰的に削除します。

論理削除の場合、アダプターは次の処理を行います。

- Update を発行して、ビジネス・オブジェクトの状況属性を、ビジネス・オブジェクトのアプリケーション固有情報によって指定されている値に設定します。アダプターでは、結果として 1 つのデータベース行だけが更新されることを確認します。それ以外の場合は、エラーを戻します。
- 所有関係にある単一カーディナリティーの子のすべて、および複数カーディナリティーの子のすべてに対し、論理削除を再帰的に実行します。アダプターは、所有関係にない単一カーディナリティーの子は削除しません。

ApplyChanges 操作

ApplyChanges 操作には、さまざまな処理が含まれます。この操作によって、Create、Update、または Delete 操作を要求するビジネス・オブジェクトに対して、アダプターがそれに応じた処理を実行することができます。

ビジネス・オブジェクトに最上位の動詞が存在する場合、そのビジネス・オブジェクトは変更後イメージとして処理されます。ビジネス・オブジェクトに最上位の動詞が存在しない場合は、ChangeSummary が処理されます。

アプリケーション固有情報

ビジネス・オブジェクト定義内のアプリケーション固有情報は、アダプターに対し、ビジネス・オブジェクトの処理方法に関するアプリケーション依存の指示を与えるものです。アダプターでは、ビジネス・オブジェクトの属性または動詞、あるいはビジネス・オブジェクト自体から取得したアプリケーション固有情報を解析して、Create、Update、Retrieve、および Delete 操作のための照会を生成します。

アダプターは、ビジネス・オブジェクトのアプリケーション固有情報の一部については、キャッシュに保管し、その情報をすべての動詞の照会をビルドするために使用します。

拡張または変更されたアプリケーション固有のビジネス・オブジェクトでは、ビジネス・オブジェクト定義内のアプリケーション固有情報は、アダプターの予期する構文に合致している必要があります。

関連タスク

49 ページの『データベース・オブジェクトの照会』
接続プロパティーの構成後、 データベース・オブジェクトに対する照会を実行できます。 エンタープライズ情報システム (EIS) 内のオブジェクトの構造を理

解するために メタデータ・ツリー構造をブラウズして、 サービス記述に必要なオブジェクトを選択することができます。

ビジネス・オブジェクトの命名規則

ビジネス・オブジェクト名には、ビジネス・オブジェクトが表す構造が反映されます。Customer または Address などです。ビジネス・オブジェクト名はたいてい、Enterprise Metadata Discovery のメタデータ・インポート処理の間に エンタープライズ情報システム (EIS) で指定された名前に基づいて付けられます。

ビジネス・オブジェクト名はラクダ記法に変換されます。ラクダ記法ではスペースやアンダースコアーのような分離文字は除去され、各語の最初の文字が大文字で表記されます。例えば、ORDER_LINE_ITEM は OrderLineItem に変換されます。

親ビジネス・オブジェクトのグラフは、そのオブジェクトに含まれるビジネス・オブジェクトの名前の後に BG を付けたものとなります。例えば、CustomerBG は Customer ビジネス・オブジェクトの親ビジネス・オブジェクトのグラフです。

ビジネス・オブジェクト名には、アダプターまたはデータベースを意味する値は含まれません。

ビジネス・オブジェクトはデータベース固有のメタデータを扱います。プレフィックスに JDBC または %AppName% のようなストリングを指定すると、アプリケーション固有と汎用の 2 つのタイプのビジネス・オブジェクトを識別するのに役立ちます。名前の残りの部分で、ビジネス・オブジェクトが表すテーブルまたはストアード・プロシージャーを説明することができます。例えば、Human Resources (HR) のようなデータベース・アプリケーションの Employee テーブル用のビジネス・オブジェクト定義を生成する場合、相当するビジネス・オブジェクト名は HREmployee です。

ビジネス・オブジェクト・レベルのアプリケーション固有情報

ビジネス・オブジェクト・レベルのアプリケーション固有情報は、対応するデータベース表の名前を指定するとき、および物理削除または論理削除操作の実行に必要な情報を指定するときに使用されます。

ビジネス・オブジェクト・レベルでは、アプリケーション固有情報の形式は、jdbcasi.xsd スキーマ定義で定義される xml で構成されています。

- *TableName* ビジネス・オブジェクトに関連付けられたデータベース表を示します。
- *StatusColumnName* は、論理的な削除操作を実行するために使用されるデータベース列の名前です。
- *StatusValue* は、ビジネス・オブジェクトが非アクティブか削除済みであることを意味する値です。

例えば、Customer ビジネス・オブジェクトで、そのアプリケーション固有情報に、以下の値が指定されているとします。

```
<jdbcasi:TableName>customer</jdbcasi:TableName>
<jdbcasi:StatusColumnName>status</jdbcasi:StatusColumnName>
<jdbcasi::StatusValue>deleted</jdbcasi:StatusValue>
```

アダプターが、カスタマーの削除要求を受信したとします。このような要求により、アダプターは次の SQL ステートメントを発行します。

```
UPDATE customer set status = 'deleted' where pkey = . . . .
```

アダプターは、ステータス列名が含まれていない場合や、このパラメーターに値が指定されていない場合には、データベースからビジネス・オブジェクトを物理的に削除します。つまり、ビジネス・オブジェクトでアプリケーション固有情報に StatusColumnName パラメーターが含まれる場合、アダプターは論理削除操作を実行します。ビジネス・オブジェクトでアプリケーション固有情報に StatusColumnName パラメーターが含まれない場合、アダプターは物理削除操作を実行します。

Update 操作と Delete 操作では、StatusColumnName プロパティーの値を以下のように使用できます。

- 子データを論理的に削除する場合、アダプターは StatusColumnName パラメーターの値を使用して、状況列の名前と状況値のテキストを取得します。詳細については、『Update 操作』を参照してください。
- Delete 操作を実行するとき、アダプターは、その StatusColumnName パラメーターの値を使用して、ビジネス・オブジェクト全体を物理的に削除するか、論理的に削除するかを判断します。StatusColumnName パラメーターに値が含まれている場合、アダプターは、論理削除操作を実行します。StatusColumnName パラメーターに値が含まれていない場合、アダプターは、物理削除操作を実行します。詳細については、『Delete 操作』を参照してください。

双方向言語用に使用可能に設定する ASI のパラメーターは、TableName と StatusColumnName です。これらのパラメーターの形式は、BiDi.Metadata プロパティーに設定された属性に基づいて変換されます。このプロパティーについて詳しくは、82 ページの『構成プロパティー』を参照してください。

動詞のアプリケーション固有情報

アダプターは、ビジネス・オブジェクトに指定されているとおり、SQL ステートメントのグループである SQL クエリーまたはストアード・プロシージャーを使用してデータベース表を更新します。このセクションでは、ストアード・プロシージャーとストアード・プロシージャー定義のエレメントについて説明します。ストアード・プロシージャー定義のサンプルも含まれています。

関連概念

41 ページの『オブジェクトの選択および生成』
ビジネス・オブジェクトを生成するには、データベース・オブジェクト・ノードを選択します。そうすると、エンタープライズ・サービス・ディスカバリー・ウィザードで、選択したノードのオブジェクトのビジネス・オブジェクトが生成されます。

関連資料

44 ページの『ビジネス・オブジェクトの属性およびアプリケーション固有情報』
ビジネス・オブジェクトの属性は、データベース・オブジェクトの列リストから作成されます。エンタープライズ・サービス・ディスカバリー・ウィザードで、

列名に属性名が設定されます。属性名ではグローバル化文字がサポートされます。 アダプターは、属性の名前、タイプ、およびアプリケーション固有情報を追加します。

ストアード・プロシージャーの概要:

ストアード・プロシージャーは、複数の SQL ステートメントのグループであり、1 つの論理単位を形成して特定のタスクを実行します。ストアード・プロシージャーは、アダプターがオブジェクトに対して実行する一連の操作または照会を、データベース・サーバー内にカプセル化したものです。

アダプターでは、単純な SQL ステートメントを使用して、Select、Update、Retrieve、Delete、または Retrievall 操作を実行できます。 SQL ステートメント用の列名は、属性の AppSpecificInfo プロパティーから取り出されます。WHERE 文節は、ビジネス・オブジェクトで指定されたキー値を使用して構成されます。各照会は、複数のテーブルにまたがることはできません。ただし、ビューに送ることはできます。

アダプターは、次の目的でストアード・プロシージャーを呼び出します。

- ビジネス・オブジェクトを処理する前に、操作準備処理を行う。
- ビジネス・オブジェクトを処理した後で、操作後処理を行う。
- 単純な Create、Update、Delete、Retrieve または RetrieveAll ステートメントを使用せずにビジネス・オブジェクトに対して一連の操作を実行する。

アダプターでは、階層ビジネス・オブジェクトを処理するときに、ストアード・プロシージャーを使用して、最上位ビジネス・オブジェクトまたは任意の子ビジネス・オブジェクトを処理することができます。ただし、ビジネス・オブジェクト (またはビジネス・オブジェクトの配列) には、ストアード・プロシージャーが個別に用意されていなければなりません。

ストアード・プロシージャー定義:

ストアード・プロシージャーは動詞レベルで定義されます。ストアード・プロシージャー定義はそれぞれ、エレメント StoredProcedureType、StoredProcedureName、ResultSet、および Parameters から構成されます。

StoredProcedureType は、使用するストアード・プロシージャーのタイプを定義します。これにより、ストアード・プロシージャーがいつ呼び出されるかが決まります。値は次のとおりです。

- BeforeCreateSP
- AfterCreateSP
- CreateSP
- BeforeUpdateSP
- AfterUpdateSP
- UpdateSP
- BeforeDeleteSP
- AfterDeleteSP
- DeleteSP

- BeforeRetrieveSP
- AfterRetrieveSP
- RetrieveSP
- BeforeRetrieveAllSP
- AfterRetrieveAllSP
- RetrieveAllSP

注: RetrieveAll に関連するストアード・プロシージャーのタイプは、最上位ビジネス・オブジェクトにのみ適用されます。

StoredProcedureName は、適切な StoredProcedureType と関連したストアード・プロシージャーの名前です。双方向言語で使用される場合に使用可能化されます。

ResultSet は、ストアード・プロシージャーが結果を戻すかどうかを決定します (true|false)。結果のセットが戻される場合は、結果のセットの行で戻される値を使用して、現在のビジネス・オブジェクトの N カーディナリティーの子が作成されます。

Parameters は、入力のみ (IP)、出力のみ (OP)、および入出力 (IO) の組み合わせです。 Oracle ストアード・プロシージャーの場合は、結果のセットが出力パラメーターとしてのみ戻されます。その場合、パラメーター・リスト内の値のいずれかが結果のセット (RS) になります。パラメーターは、双方向言語で使用される場合に使用可能化されます。

次に、ストアード・プロシージャー定義の例を示します。

```
<jdbcasi:JDBCBusinessObjectTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
            <jdbcasi:TableName>customer</jdbcasi:TableName><jdbcasi:Operation>
               <jdbcasi:Name>Retrieve</jdbcasi:Name>
               <jdbcasi:StoredProcedures>
<jdbcasi:StoredProcedureType>RetrieveSP</jdbcasi:StoredProcedureType>
<jdbcasi:StoredProcedureName>retrieve_cust</jdbcasi:StoredProcedureName>
<jdbcasi:ResultSet>false</jdbcasi:ResultSet>
                     <jdbcasi:Parameters>
                          <jdbcasi:Type>IP</jdbcasi:Type>
                          <jdbcasi:PropertyName>primaryKey</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                    <jdbcasi:Parameters>
                          <jdbcasi:Type>OP</jdbcasi:Type>
                           <jdbcasi:PropertyName>custCode</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                    <jdbcasi:Parameters>
                          <jdbcasi:Type>OP</jdbcasi:Type>
                           <jdbcasi:PropertyName>firstName</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                    <jdbcasi:Parameters>
                          <jdbcasi:Type>OP</jdbcasi:Type>
                              <jdbcasi:PropertyName>lastName</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                    </jdbcasi:StoredProcedures>
                    <jdbcasi:StoredProcedures>
<jdbcasi:StoredProcedureType>AfterRetrieveSP</jdbcasi:StoredProcedureType>
<jdbcasi:StoredProcedureName>retrieve_cust</jdbcasi:StoredProcedureName>
                    <jdbcasi:ResultSet>false</jdbcasi:ResultSet>
                    <jdbcasi:Parameters>
                         <jdbcasi:Type>IP</jdbcasi:Type>
                           <jdbcasi:PropertyName>primaryKey</jdbcasi:PropertyName>
```

```
                    </jdbcasi:Parameters>
                    <jdbcasi:Parameters>
                         <jdbcasi:Type>OP</jdbcasi:Type>
                          <jdbcasi:PropertyName>custCode</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                    <jdbcasi:Parameters>
                         <jdbcasi:Type>OP</jdbcasi:Type>
                          <jdbcasi:PropertyName>firstName</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                    <jdbcasi:Parameters>
                         <jdbcasi:Type>OP</jdbcasi:Type>
                         <jdbcasi:PropertyName>lastName</jdbcasi:PropertyName>
                    </jdbcasi:Parameters>
                </jdbcasi:StoredProcedures>
            </jdbcasi:Operation>
</jdbcasi:JDBCBusinessObjectTypeMetadata>
```

プロパティー ReturnDummyBOForSP は、結果のセットが true でありながら空の場合にも、出力パラメーターを戻します。 RetrieveSP の場合、結果のセットが戻されます。結果のセットが空の場合は、ビジネス・オブジェクトが生成されず、プロシージャー呼び出しの戻す出力パラメーターをリトリーブする方法も存在しません。ReturnDummyBOForSP が true の場合は、対応する属性に読み込まれた出力パラメーターと入出力パラメーターの値を持つダミーのビジネス・オブジェクトが戻されます。このプロパティーのデフォルト値は false です。

関連概念

8 ページの『ビジネス・オブジェクトの構造』
それぞれのビジネス・オブジェクトは、データベース表やビューに対応し、オブジェクト内の単純属性がそれぞれ表やビュー内の列に相当します。

属性に関するアプリケーション固有情報

このトピックでは、属性に関するアプリケーション固有情報 (ASI) について説明し、サポートされるパラメーターとその説明をリストします。

属性のアプリケーション固有情報は、属性が単純属性であるか、子ビジネス・オブジェクト (または子ビジネス・オブジェクトの配列) を表す属性であるかによって異なります。子を表す属性のアプリケーション固有情報は、親子関係が子に格納されるか、親に格納されるかによっても異なります。

単純属性のアプリケーション固有情報

単純属性では、アプリケーション固有情報の形式は、いくつかのパラメーターとその値で構成されています。属性のアプリケーション固有情報の形式は、.xsd ファイルの以下の実例セクションに示します。

```
          <jdbcasi:ColumnName>pkey</jdbcasi:ColumnName>
          <jdbcasi:PrimaryKey>true</jdbcasi:PrimaryKey>
          <jdbcasi:FixedChar>true</jdbcasi:FixedChar>
       </jdbcasi:JDBCAttributeTypeMetadata>
       </appinfo>
       </annotation>
       <simpleType>
          <restriction base="string">
          <maxLength value="10"/>
          </restriction>
       </simpleType>
       </element>
       <element name="custCode" type="string">
       <annotation>
       <appinfo source="WBI">
       <jdbcasi:JDBCAttributeTypeMetadata
```

```
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
   <jdbcasi:ColumnName>ccode</jdbcasi:ColumnName>
   <jdbcasi:ForeignKey>custinfoObj/custCode</jdbcasi:ForeignKey>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
<element name="firstName" type="string">
<annotation>
<appinfo source="WBI">
<jdbcasi:JDBCAttributeTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
   <jdbcasi:ColumnName>fname</jdbcasi:ColumnName>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
<element name="lastName" type="string">
<annotation>
<appinfo source="WBI">
<jdbcasi:JDBCAttributeTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
   <jdbcasi:ColumnName>lname</jdbcasi:ColumnName>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
```

コネクターが処理する単純属性に必須のパラメーターは、列名のみです。例えば、以下は、列名のみを指定する形式です。ここで、ccode はカスタマー・コードを表します。

```
<jdbcasi:ColumnName>ccode</jdbcasi:ColumnName>
```

双方向言語用に使用可能に設定する属性の ASI パラメーターは、ColumnName と ForeignKey です。これらのパラメーターの形式は、BiDi.Metadata プロパティーに設定されている属性に基づいて変換されます。このプロパティーについて詳しくは、『参照』のセクションの『構成プロパティー』を参照してください。双方向プロパティーについて詳しくは、双方向言語サポートの一般技術論文、および IBM developerWorks(R) Web サイトの アダプター技術論文を参照してください。

『属性のアプリケーション固有情報内のパラメーター』の表に、各パラメーターとその説明をリストします。

属性のアプリケーション固有情報内のパラメーター

パラメーター	説明
ByteArray	true の場合、アダプターはデータベースでバイナリー・データを読み取りおよび書き込みし、そのデータをストリングとしてアプリケーション・サーバーに渡します。デフォルトでは、値は false です。詳細については、『Working with binary data』を参照してください。
ColumnName	このパラメーターの値は、この属性に対するデータベース列の名前です。これは双方向言語で使用される場合に使用可能化されます。

パラメーター	説明
FixedChar	このパラメーターは、表内の列が VARCHAR 型ではなく CHAR 型である場合に、属性を固定長とするかどうかを指定します。例えば、ある特定の属性が CHAR 型の列にリンクされている場合、アダプターはデータベースを照会するときに、属性値をその属性の最大長までブランクで埋めます。デフォルトでは、値は `false` です。 ビジネス・オブジェクトの .xsd ファイルでは、このパラメーターは手動で更新する必要があります。ファイルはテキスト・モードで編集したり、WebSphere Integation Developer の Business Object Editor で編集したりできます。更新後は .xsd 内で検証エラーが発生していないことを確認してください。この表に続くこのパラメーターのコード例を参照してください。
ForeignKey	このプロパティーの値は、親子関係が親ビジネス・オブジェクトに格納されるか、子ビジネス・オブジェクトに格納されるかによって異なります。 親に保管される場合、子ビジネス・オブジェクトのタイプと、外部キー *(ChildBOname/ChildPropertyName)* として使用される子ビジネス・オブジェクト内の属性の名前の両方が含まれるように値を設定します。 子に保管される場合は、外部キーとして使用される親ビジネス・オブジェクト内の属性の名前のみを含むように値を設定します。 属性が外部キーではない場合は、アプリケーション固有情報にこのパラメーターを含めないでください。 このパラメーターは双方向言語の場合に使用可能。
KeepRelationship	`true` の場合、このパラメーターは Update 操作中に子ビジネス・オブジェクトを削除しないようにします。

パラメーター	説明
OrderBy	このパラメーターに値が指定されている場合、このパラメーターが指定されている属性が子ビジネス・オブジェクト内に存在するものであれば、アダプターでは、検索照会の ORDER BY 文節で、その属性の値を使用します。アダプターは、子ビジネス・オブジェクトを昇順 (ASC) または降順 (DESC) で検索することができます。このパラメーターがアプリケーション固有情報に含まれていない場合、アダプターは、検索順序を指定するときに、このパラメーターが指定されている属性を使用しません。
Ownership	このパラメーターは、子ビジネス・オブジェクトが親によって所有されることを指定します。`true` の場合、子ビジネス・オブジェクトに対する Create、Update、および Delete 操作が許可されます。`false` の場合、これらの更新のうちのどれも、子ビジネス・オブジェクトには適用できません。親が作成されると、データベース内で関係の整合性が保持されるように、子が存在するかどうかが検証されます。
PrimaryKey	値が `true` の場合、このプロパティーはこの属性に関連した列が、データベース内の対応するテーブルのキーであることを暗黙指定します。
UniqueIdentifier (UID)	アダプターは、ビジネス・オブジェクトの固有 ID の生成に、このパラメーターを使用します。シーケンスと ID 列 *(UID=AUTO|SequenceName)* の生成がサポートされます。シーケンスを定義できるのは、DB2 および Oracle データベースのみです。ID 列を定義できるのは、DB2 および Microsoft SQL Server です。属性で固有 ID が必要とされない場合は、このパラメーターをアプリケーション固有情報に含めないでください。

ビジネス・オブジェクトの .xsd ファイル内の FixedChar パラメーターの例

```
<element name="primaryKey">
<annotation>
<appinfo source="WBI">
        <jdbcasi:JDBCAttributeTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
                <jdbcasi:ColumnName>pkey</jdbcasi:ColumnName>
                <jdbcasi:PrimaryKey>true</jdbcasi:PrimaryKey>
                <jdbcasi:FixedChar>true</jdbcasi:FixedChar>
        </jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
<simpleType>
        <restriction base="string">
```

```
                <maxLength value="10"/>
            </restriction>
</simpleType>
</element>
```

関連資料

44 ページの『ビジネス・オブジェクトの属性およびアプリケーション固有情報』
ビジネス・オブジェクトの属性は、データベース・オブジェクトの列リストから作成されます。エンタープライズ・サービス・ディスカバリー・ウィザードで、列名に属性名が設定されます。属性名ではグローバル化文字がサポートされます。 アダプターは、属性の名前、タイプ、およびアプリケーション固有情報を追加します。

アダプターのインストール

このセクションでは、Adapter for JDBC のインストール要件と、インストール済みファイル構造について説明します。

アダプターのインストール方法については、『IBM WebSphere Adapters のインストール』を参照してください。

アダプター環境

アダプター環境のハードウェア要件およびソフトウェア要件は、オンラインで参照できます。

このアダプターの要件を検索するには、
http://www-1.ibm.com/support/docview.wss?uid=swg27006249 の「IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: software requirements」を参照してください。WebSphere アダプターのリストからご使用のアダプターを選択してください。

Adapter for JDBC に固有のインストール情報

Adapter for JDBC には、JDBC ドライバーおよびイベント・ストアのセットアップに関連するインストールの前提条件があります。

JDBC ドライバーの前提条件

JDBC ドライバーは、JDBC ドライバー JAR ファイルがクラス・パス内に含まれるように設定する必要があります。

JDBC エンタープライズ・サービス・ディスカバリーは、エンタープライズ・サービス・ディスカバリー・ウィザード内のプロジェクト内に存在します。JDBC エンタープライズ・サービス・ディスカバリー処理を実行するには、JDBC ドライバー JAR ファイルがこのツールの JDBC プロジェクトのクラス・パスに含まれている必要があります。

イベント・ストアのセットアップ

Inbound 処理を実行するには、まずデータベースでイベント・ストアをセットアップする必要があります。IBM® DB2®、Oracle、または Microsoft™ SQLServer データベース用イベント・ストアをセットアップするために、以下のようなサンプル・スクリプトが用意されています。

- `WBIA_JDBC_EventStore_DB2.sql`
- `WBIA_JDBC_EventStore_Oracle.sql`
- `WBIA_JDBC_EventStore_MSSQL.sql`

ユーザー・テーブルへの変更によって、イベント・ストアに保管されるイベントが自動的に生成されるように、必要に応じてユーザー・テーブルにトリガーをセットアップする必要があります。

インストール済みファイルの構造

アダプターをインストールしたら、インストールされたファイルおよびディレクトリーを見ることができます。それらのすべてのファイルやディレクトリーでは、インストール・ディレクトリーがルートとなります。

例えば、アダプターのインストール・ディレクトリーが c:¥WebSphereBI の場合、CWYBC_JDBC.rar ファイルの絶対パスは `c:¥WebSphereBI.¥adapter¥JDBC¥deploy¥CWYBC_JDBC.rar` になります。

リソース・アダプター・アーカイブ (RAR) ファイルには、アダプターのファイルとエンタープライズ・サービス・ディスカバリー・ツールのファイルの両方が含まれています。

UNIX(R) および Windows(TM) のプラットフォームでは、インストール済みディレクトリーおよびファイルの同じ構造が共用されます。唯一の相違はディレクトリー・パスの指定です (UNIX の場合はスラッシュ、Windows では円記号を使用)。

UNIX/Linux のディレクトリーおよびファイルの構造

以下の表に、WebSphere Adapter for JDBC の UNIX/Linux(TM) ディレクトリーおよびファイルをリストします。ディレクトリーおよびファイルはカテゴリーごとにグループ化されています。

ファイルおよびディレクトリーのカテゴリー	ディレクトリーおよびファイル
Adapter for JDBC RAR ファイル	adapter/deploy/CWYBC_JDBC.rar
IBM Support Assistant プラグイン ZIP ファイル	adapter/ISAPlugin/com.ibm.esupport.client.SS6FE6 _RAJDBC.zip
IBM Tivoli License Manager (ITLM) ファイル	adapter/5724L77E060000.sys
メッセージ・ファイル	adapter/messages/CWYBC_JDBC_messages.tar
ファウンデーション・クラスのメッセージ・ファイル	adapter/messages/CWYBS_AdapterFoundation _messages.tar
ICU4J 用通知ファイル	adapter/notices.txt

ファイルおよびディレクトリーのカテゴリー	ディレクトリーおよびファイル
サンプル・アプリケーション EAR ファイル	adapter/samples/Apps/JDBCApp.ear
IBM DB2 用サンプル・スクリプト	adapter/samples/scripts/scripts_db2.sql
Oracle 用サンプル・スクリプト	adapter/samples/scripts_oracle.sql

Windows の場合のディレクトリーおよびファイルの構造

以下の表に、WebSphere Adapter for JDBC の Windows ディレクトリーおよびファイルをリストします。ディレクトリーおよびファイルはカテゴリーごとにグループ化されています。

ファイルおよびディレクトリーのカテゴリー	ディレクトリーおよびファイル
Adapter for JDBC RAR ファイル	adapter¥deploy¥CWYBC_JDBC.rar
IBM Support Assistant プラグイン ZIP ファイル	adapter¥ISAPlugin¥com.ibm.esupport.client.SS6FE6 _RAJDBC.zip
IBM Tivoli License Manager (ITLM) ファイル	adapter¥5724L77E060000.sys
メッセージ・ファイル	adapter¥messages¥CWYBC_JDBC_messages.zip
ファウンデーション・クラスのメッセージ・ファイル	adapter¥messages¥CWYBS_AdapterFoundation _messages.zip
ICU4J 用通知ファイル	adapter¥notices.txt
サンプル・アプリケーション EAR ファイル	adapter¥samples¥Apps¥JDBCApp.ear
IBM DB2 用サンプル・スクリプト	adapter¥samples¥scripts¥scripts_db2.sql
Oracle 用サンプル・スクリプト	adapter¥samples¥scripts_oracle.sql

アダプター・プロジェクトの作成

WebSphere(R) Integration Developer でアダプター・プロジェクトを作成する必要があります。その後 WebSphere Integration Developer のエンタープライズ・サービス・ディスカバリー・ウィザードを使用して、ビジネス・オブジェクトおよびサービス構成を生成します。構成プロパティーを設定します。最後に、アダプター・プロジェクトをアプリケーション・サーバーに配置します。

アダプターを配置する前に、以下の製品をインストールする必要があります。

- WebSphere Integration Developer バージョン 6.0
- WebSphere Process Server -- WebSphere Process Server 管理コンソールを使用してアダプター・プロジェクトをアプリケーション・サーバーに配置し、プロパティー値を再構成する。

http://www.ibm.com/software/integration/wps から IBM(R) WebSphere Process Server のインストール手順を参照してください。

WebSphere Adapter は、Windows(TM) または Linux(TM) オペレーティング・システムを備えたコンピューターにのみインストールできます。そこから、UNIX(R) ベース・システムに配置できます。各アダプターは、リソース・アダプター・アーカイブ (RAR) ファイルとしてインストールされます。

基本的に、アダプターの配置は WebSphere Process Server への他のコンポーネントの配置と同じです。WebSphere Process Server へのコンポーネントの配置について詳しくは、http://www.ibm.com/software/integration/widの「WebSphere Integration Developer のユーザー・ガイド」を参照してください。

アダプター・プロジェクトを作成するには、以下のタスクを実行する必要があります。

- アダプター用のプロジェクトを作成する
- ベンダー・ライブラリーを追加する
- ビジネス・オブジェクトおよびサービス構成を生成する
- 構成プロパティー値を設定し、新規ビジネス統合モジュールにその値と成果物を保管する
- アダプター・プロジェクトをサーバーに配置して、アプリケーションを開始する

アダプター用のプロジェクトの作成

アダプター配置の最初のタスクは、アダプター用の J2EE(TM) コネクター・プロジェクトの作成です。Adapter for JDBC のリソース・アダプター・アーカイブ (RAR) ファイルは、WebSphere(R) Integration Developer にインポートする必要があります。これにより、WebSphere Integration Developer 内のワークスペースにプロジェクトがセットアップされます。

1. WebSphere Integration Developer を開始する

   詳細については、 http://www/ibm.com/software/integration.widにある「WebSphere Integration Developer のユーザー・ガイド」を参照してください。

2. RAR ファイルをインポートする

   WebSphere Integration Developer で、「ビジネス・インテグレーション」パースペクティブに移動します。「**ファイル**」>「**インポート**」をクリックします。「選択」ウィンドウで、インポート・ソースとして「**RAR ファイル**」を選択し、「**次へ**」をクリックします。

   「コネクター・インポート (Connector Import)」ウィンドウで、「**ブラウズ**」ボタンを使用して RAR ファイルのロケーションを選択します。プロジェクト名が「**コネクター・プロジェクト**」フィールドに自動的に表示されますが、これは必要に応じて変更できます。

   チェック・ボックス「**EAR プロジェクトへのモジュールの追加 (Add module to an EAR project)**」を選択解除します。

RAR ファイルをインポートするには、「**完了 (Finish)**」をクリックします。これにより、ワークスペースに J2EE コネクター・プロジェクトが作成されます。

3. JDBC ドライバーをコネクター・プロジェクトに追加する

   アプリケーション・サーバーに配置する Enterprise Application Archive (EAR) ファイルの一部とするため、JDBC ドライバーをコネクター・プロジェクトに追加する必要があります。これは、RAR ファイルのインポート後、またはアプリケーション・サーバーへの EAR のインストール後に実行できます。

   RAR ファイルのインポート後、ワークスペース内の適切なフォルダーに JAR ファイルを追加します。JAR ファイルは、例えば、次のロケーションに追加できます。`C:¥workspace¥CWYBC_JDBC¥connectorModule`

   代わりに、アプリケーション・サーバーに EAR をインストールした後コネクター・プロジェクトに JDBC ドライバーを追加する場合は、アプリケーションのインストール後、WebSphere Process Server の installedApps ディレクトリーの RAR サブディレクトリーに JAR ファイルを追加します。アプリケーション・サーバーでのアプリケーションのインストールの詳細については、『アダプター・プロジェクトの配置』を参照してください。

関連タスク

58 ページの『アダプター・プロジェクトの配置』
プロジェクト・ファイルは 、WebSphere(R) Integration Developer のワークスペース内の J2EE(TM) コネクター・プロジェクトです。これを Enterprise Application Archive (EAR) ファイルとしてローカル・ファイル・システムにエクスポートする必要があります。次に、アプリケーション・サーバーにプロジェクトの EAR ファイルをアップロードおよびインストール する前に JCA コネクター・セキュリティーの認証情報を提供する必要があります。

ベンダー・ライブラリーの追加

WebSphere(R) Integration Developer でプロジェクトを作成したら、JDBC ドライバーへの参照をプロジェクトに追加する必要があります。

Java ビルド・パスに JDBC ドライバーを追加する

WebSphere Integration Developer の「構成」ウィンドウで、「**コネクター・プロジェクト**」を右クリックします。「**プロパティー**」を選択します。

外部 JAR ファイルを追加するには、「**Java のビルド・パス (Java Build Path)**」をクリックします。「**ライブラリー**」タブを選択し、「**外部 Jar の追加 (Add External Jars)**」をクリックします。「ファイル・システム」ウィンドウで、「**JDBC ドライバー**」に移動して JAR ファイルを選択します。

ビジネス・オブジェクトの生成

ビジネス・オブジェクトを生成するには、まず接続プロパティーを設定します。その後データベース・オブジェクトを照会するクエリーを実行して、サービス記述に必要なオブジェクトを選択できます。次に、インポート・ファイルおよびエクスポ

ート・ファイル用の選択プロパティーの値を指定する必要があります。最後に、構成プロパティーを設定して、新規ビジネス統合モジュールに成果物およびプロパティー値を保管します。

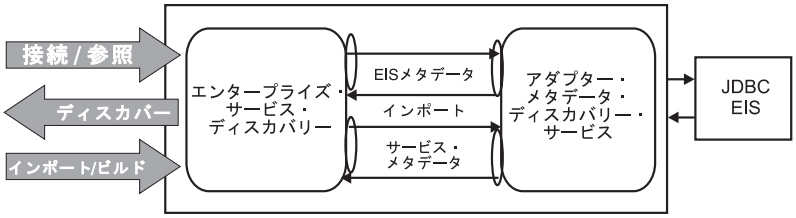
ビジネス・オブジェクト生成タスクを開始する前に、メタデータ・インポート、システム機能のディスカバリー、およびデータ記述についての各セクションで処理の詳細を参照することができます。

メタデータ・インポート

WebSphere(R) Adapter for JDBC のエンタープライズ・サービス・ディスカバリー・ウィザードを使用して、データベース内のオブジェクトを検出し、選択したオブジェクトからビジネス・オブジェクトを生成します。また、エンタープライズ・サービス・ディスカバリー・サポートは、アダプターが Service Component Architecture (SCA) コンポーネントとして稼働できるようにするサービス構成も生成します。

ビジネス・オブジェクトの作成元オブジェクトには、テーブル、ビュー、ストアード・プロシージャー、同義語/ニックネームが含まれます。アダプターを使用すると、データベース内のすべてのスキーマのリストを生成することによってオブジェクトをディスカバーできます。各スキーマ内には、テーブル、ビュー、ストアード・プロシージャー、および同義語/ニックネームのリストが含まれています。それらを選択して、エンタープライズ・サービス・ディスカバリー・ウィザードに対応するビジネス・オブジェクトを生成するように要求できます。このウィザードでは、オブジェクトのメタデータを分析し、ビジネス・オブジェクト内に属性を生成します。属性は、すべてのデータベース・オブジェクトの列名に基づいて生成されます。その機能をまとめるために、エンタープライズ・サービス・ディスカバリー・ウィザードでは以下の処理を行います。

- データベース・オブジェクトに対応するビジネス・オブジェクトを生成する。
- データベース・オブジェクト内のプロパティーに対応するビジネス・オブジェクト内のプロパティーを生成する。
- ビジネス・オブジェクトに関するアプリケーション固有情報を設定する。
- 入出力ファイルおよび Web サービス記述言語 (WSDL) ファイルを生成するのに使用するサービス記述 (Inbound および Outbound) を作成する。



ESD 上位図

システム機能のディスカバリー

アダプターは、データベースを分析し、スキーマを識別することによって、データベース内のビジネス・オブジェクトをディスカバーします。次に、データベースからすべてのオブジェクトのリストを作成し、それをツリー構造として示します。

スキーマは、ツリー内の最上位ノードとして表示されます。スキーマ・ノードは、生成用には選択できません。

各スキーマの下には、Tables、Views、Stored Procedures、および Synonyms/Nicknames とラベル付けされたノードが存在します。これらのノードは生成用に選択できません。これらのノードの下に示されるオブジェクトは、テーブル、ビュー、ストアード・プロシージャー、および同義語/ニックネームの名前です。これらのノードは生成用に選択可能とマークされます。特定のスキーマにテーブル、ビュー、ストアード・プロシージャー、または同義語がない場合は何もリストされません。

ツリー内に表示するスキーマのリストを絞り込むにはフィルター・プロパティーを指定します。指定しないとすべてのスキーマが表示されます。『参照』セクションの『フィルターおよびノードのプロパティー』で、指定が必要なプロパティーについて説明しています。

テーブル、ビュー、ストアード・プロシージャー、および同義語/ニックネームのいずれかのノードを選択して「**フィルター (Filter)**」をクリックすると、エンタープライズ・サービス・ディスカバリー・ウィザードが ObjectNameFilter を照会します。ObjectNameFilter プロパティーを使用して、表示するデータベース・オブジェクトのリストをフィルタリングできます。『参照』セクションの『フィルターおよびノードのプロパティー』で、ObjectNameFilter プロパティーについて説明しています。

関連資料

95 ページの『フィルターおよびノードのプロパティー』
ツリーに表示するスキーマのリストを絞り込む場合、ビジネス・オブジェクトのディスカバリー中にフィルター・プロパティーを指定できます。指定しないと、すべてのスキーマが表示されます。表示するデータベース・オブジェクトのリストを絞り込む場合、ノードのプロパティーを設定します。

オブジェクトの選択および生成:

ビジネス・オブジェクトを生成するには、データベース・オブジェクト・ノードを選択します。そうすると、エンタープライズ・サービス・ディスカバリー・ウィザードで、選択したノードのオブジェクトのビジネス・オブジェクトが生成されます。

複数のデータベース・オブジェクト・ノードを選択できます。フィルター・プロパティーを指定する場合、「**ビジネス・オブジェクト ASI の追加**」をチェックすると、エンタープライズ・サービス・ディスカバリー・ウィザードは選択したノードごとにステータス列名、ステータス値、およびストアード・プロシージャー関連のオブジェクト・レベル・パラメーターを要求します。

ステータス列名については、ステータス列名を設定する特定のオブジェクトの実際の列名のリストが示されます。ステータス値を入力する必要があります。これらの値は、ビジネス・オブジェクト・レベルのアプリケーション固有情報 (ASI) に設定されます。

ストアード・プロシージャーをビジネス・オブジェクトに関連付けることもできます。サポートされるすべてのストアード・プロシージャー・タイプのリストが示されます。各ストアード・プロシージャー・タイプは、データベースで使用可能なス

トアード・プロシージャーのリストを持っています。1 つのストアード・プロシージャーを特定のストアード・プロシージャー・タイプに割り当てることができます。

割り当てられた各ストアード・プロシージャーに対して、ストアード・プロシージャーの入出力パラメーターのリストが示されます。各パラメーターは、ビジネス・オブジェクトの属性のリストを持っています。パラメーターのタイプは、プロパティーの記述にリストされます。ストアード・プロシージャー・パラメーターごとに、ビジネス・オブジェクト属性を 1 つ選択できます。ビジネス・オブジェクトの動詞アプリケーション固有情報に、ストアード・プロシージャーが 1 つ割り当てられているすべてのストアード・プロシージャー・タイプが追加されます。『動詞のアプリケーション固有情報』を参照してください。

データベース・オブジェクトを選択した後、選択プロパティーの値を設定する必要があります。エンタープライズ・サービス・ディスカバリー・ウィザードが、選択プロパティーを照会します。これらのプロパティーについて詳しくは、『参照』セクションの『選択プロパティー』を参照してください。

関連タスク

96 ページの『選択プロパティー』
データベース・オブジェクトを選択した後、 選択プロパティーの値を設定する必要があります。

関連資料

28 ページの『動詞のアプリケーション固有情報』
アダプターは、ビジネス・オブジェクトに指定されているとおり、SQL ステートメントのグループである SQL クエリーまたはストアード・プロシージャーを使用してデータベース表を更新します。このセクションでは、ストアード・プロシージャーとストアード・プロシージャー定義のエレメントについて説明します。ストアード・プロシージャー定義のサンプルも含まれています。

データ記述

データ記述は、アダプターが Service Component Architecture (SCA) コンポーネントとして稼働できるようにするためにエンタープライズ・サービス・ディスカバリー処理によって生成されるサービス構成の一部です。

データ記述には、クライアント・アプリケーションとアダプターの間で実行時に引き渡しされる、アダプターのビジネス・オブジェクトの構造と内容の定義が含まれます。データ記述を使用すると、クライアント・アプリケーションが要求に対応する適切なデータ・オブジェクトを作成し、応答として戻されたデータ・オブジェクトを解釈することができます。データベース・コンポーネントから生成されたデータ記述は、XML スキーマとして表されます。

- ビジネス・オブジェクトは、複合タイプの定義にマップします。
- ビジネス・オブジェクトの属性は、エレメント・タイプの定義にマップします。
- ビジネス・オブジェクトのアプリケーション固有の情報は、複合タイプでの注釈に含まれます。
- ビジネス・オブジェクト内の各プロパティーのアプリケーション固有情報は、エレメント・タイプに関する注釈に含まれます。

注: ビジネス・オブジェクトおよび属性レベルのアプリケーション固有のプロパティーのテンプレートが、JDBC アダプターのメタデータ・スキーマに定義されています。このスキーマ・ファイルの名前は JDBCASI.xsd です。生成されたスキーマ・ファイルの注釈には、このテンプレートへの参照が含まれます。

ビジネス・オブジェクトのスキーマおよびアプリケーション固有情報:

ビジネス・オブジェクトのスキーマは、選択したデータベース・コンポーネントから作成されます。各コンポーネントは最上位ビジネス・オブジェクトになります。

エンタープライズ・サービス・ディスカバリー・ウィザードで、
PrefixSchemaNameObjectName の形のビジネス・オブジェクト名が生成されます。ここで、

- *Prefix* はプレフィックスという名前の接続プロパティーで指定された値です。Prefix は必須ではありません。指定しないと、ビジネス・オブジェクト名にプレフィックスが追加されません。
- *SchemaName* は、オブジェクトが属するスキーマの名前です。
- *ObjectName* はテーブル、ビュー、ストアード・プロシージャー、または同義語/ニックネームの名前です。

ビジネス・オブジェクト名ではグローバル化文字がサポートされます。

エンタープライズ・サービス・ディスカバリー・ウィザードで、アプリケーション固有情報属性の TableName に *schemaname.tablename* の形の値が設定されます。また、このウィザードでは、ビジネス・オブジェクト・レベルのアプリケーション固有情報が『ビジネス・オブジェクトのアプリケーション固有情報 (ASI)』の表にリストされているように設定されます。ビジネス・オブジェクトに選択した操作が設定されます。生成されたすべてのビジネス・オブジェクトは、生成元のオブジェクト・タイプ (テーブル、ビュー、ストアード・プロシージャー、または同義語/ニックネーム) に関係なく同じ構造を持ちます。すべてのビジネス・オブジェクトは列に基づいた属性を持っています。オブジェクト名には TableName ASI が設定されます。

ビジネス・オブジェクトのアプリケーション固有情報 (ASI)

ビジネス・オブジェクト ASI	エンタープライズ・サービス・ディスカバリー・ウィザードによって設定	追加情報
TableName	はい	実際の列名に設定します。
ステータス列名	はい	オブジェクトの選択時に指定します。
ステータス値	はい	オブジェクトの選択時に指定します。

作成されたすべてのビジネス・オブジェクトは最上位になります。エンタープライズ・サービス・ディスカバリー・ウィザードでは、再帰的 (子) ビジネス・オブジェクトは作成されません。また、エンタープライズ・サービス・ディスカバリー・ウィザードでは、すべてのビジネス・オブジェクトが最上位であるため、すべてのビジネス・オブジェクトのビジネス・グラフも生成されます。ビジネス・グラフの名

前は、ビジネス・オブジェクト名に「BG」が付いたものです。例えば、JDBCSchema1Customer という名前のビジネス・オブジェクトのビジネス・グラフの名前は JDBCSchema1CustomerBG となります。ビジネス・オブジェクトに設定された操作はビジネス・グラフにも設定されます。

ビジネス・オブジェクトの属性およびアプリケーション固有情報:

ビジネス・オブジェクトの属性は、データベース・オブジェクトの列リストから作成されます。エンタープライズ・サービス・ディスカバリー・ウィザードで、列名に属性名が設定されます。属性名ではグローバル化文字がサポートされます。アダプターは、属性の名前、タイプ、およびアプリケーション固有情報を追加します。

JDBC メタデータによって戻されるタイプは、『JDBC メタデータの列タイプとビジネス・オブジェクト属性タイプ』の表にリストされているとおり、ビジネス・オブジェクト属性タイプにマップされます。リストされている JDBC タイプのみがアダプターでサポートされます。リストされていないタイプの列は、ビジネス・オブジェクトに追加されません。例えば、「テーブル *yyyy* の列名 *xxxx* のタイプはサポートされていません。ビジネス・オブジェクトには追加されません」という情報メッセージが作成されます。

JDBC メタデータ列タイプとビジネス・オブジェクト属性タイプ

JDBC メタデータの列タイプ	ビジネス・オブジェクト属性のタイプ
BIT	BOOLEAN
CHAR LONGVARCHAR VARCHAR	STRING
INTEGER NUMERIC SMALLINT TINYINT BIGINT	INTEGER
TIME TIMESTAMP DATE	DATE
DECIMAL	STRING
DOUBLE FLOAT	DOUBLE
REAL	FLOAT

『属性情報』の表には、エンタープライズ・サービス・ディスカバリー・ウィザードによって設定された属性情報がリストされ、その設定方法が説明されています。

属性情報

属性情報	エンタープライズ・サービス・ディスカバリーによって設定	追加情報
Cardinality	いいえ	

属性情報	エンタープライズ・サービス・ディスカバリーによって設定	追加情報
Name	はい	属性の名前。これは双方向言語の場合に使用可能化されます。
MinOccurs/MaxOccurs	はい	列が基本キーではなくかつ NULL が不可能な場合、この属性は必須で、値は 1 以上に設定されます。
Type	はい	『JDBC メタデータ列とビジネス・オブジェクト属性タイプ』の表のとおりに設定します。

エンタープライズ・サービス・ディスカバリー・ウィザードは、ビジネス・オブジェクトの属性に関するアプリケーション固有情報 (ASI) を『属性に関するアプリケーション固有情報 (ASI)』の表で示したとおりに設定します。属性のアプリケーション固有情報について詳しくは、『単純属性のアプリケーション固有情報』を参照してください。

属性に関するアプリケーション固有情報

属性 ASI	エンタープライズ・サービス・ディスカバリーによって設定	追加情報
ColumnName	はい	実際の列名に設定します。これは双方向言語の場合に使用可能化されます。
FixedChar	いいえ	ビジネス・オブジェクト .xsd ファイルでは手動で更新する必要があります。このファイルは、テキスト・モードを使用しても、WebSphere Integration Developer の Business Object Editor を使用しても編集できます。ファイルを更新したら、検証エラーがないことを確認してください。『単純属性のアプリケーション固有情報』セクション内の .xsd ファイルの FixedChar の例を参照してください。
ForeignKey	いいえ	
OrderBy	いいえ	
PrimaryKey	はい	列が基本キーの場合、PrimaryKey は `true` に設定されます。

属性 ASI	エンタープライズ・サービス・ディスカバリーによって設定	追加情報
UID	いいえ	

ビジネス・オブジェクトにストアード・プロシージャーを追加する場合、動詞アプリケーション固有情報 (ASI) は『動詞のアプリケーション固有情報』の表に指定されているとおりに設定されます。有効なストアード・プロシージャー・タイプについては、『動詞のアプリケーション固有情報』のセクションを参照してください。

動詞のアプリケーション固有情報

動詞 ASI またはストアード・プロシージャーのパラメーター・エレメント	エンタープライズ・サービス・ディスカバリー・ウィザードによって設定	追加情報
Parameters	はい	ストアード・プロシージャーのパラメーターをリストします。これは双方向言語の場合に使用可能化されます。
PropertyName	はい	選択したビジネス・オブジェクト属性の名前に設定します。これは双方向言語の場合に使用可能化されます。
ResultSet	いいえ	ストアード・プロシージャーから ResultSet が戻される場合、ビジネス・オブジェクト定義でこのパラメーターを `true` に設定する必要があります。
StoredProcedure	はい	ストアード・プロシージャー名に設定します。これは双方向言語の場合に使用可能化されます。
StoredProcedure Type	はい	タイプのリストから選択します。
Type	はい	ストアード・プロシージャー・パラメーターのタイプ (IP/OP/IO) に設定します。

階層ビジネス・オブジェクトの作成

エンタープライズ・サービス・ディスカバリー・ウィザードでは、フラット・ビジネス・オブジェクトを生成します。このウィザードでは、データベースに定義された異なるテーブル間の外部キー制限は使用せずに、関係を自動的に作成します。手動でリンクさせるにはこれらが必要です。ビジネス・オブジェクト定義は、テキスト・モードで更新したり、Business Object Editor を使用して更新したりできます。

単一または複数カーディナリティー子ビジネス・オブジェクトの .xsd 定義ファイルの例をここに示します。エレメント custInfoObj は単一カーディナリティー子ビジネス・オブジェクトで、addressObj は複数カーディナリティー子ビジネス・オブジェクトです。

```
<element name="addressObj" minOccurs="1" type="Address:Address"
maxOccurs="unbounded">
                        <annotation>
                        <appinfo source="WBI">
                        <pasi:JDBCAttributeTypeMetadata xmlns:pasi=
"urn:app:jdbc:asi">
                                <pasi:Ownership>true</pasi:Ownership>
                        </pasi:JDBCAttributeTypeMetadata>
                        </appinfo>
                        </annotation>
                        </element>
                        <element name="custInfoObj" minOccurs="0" type=
"CustInfo:CustInfo" maxOccurs="1">
                        <annotation>
                        <appinfo source="WBI">
                        <pasi:JDBCAttributeTypeMetadata xmlns:pasi=
"urn:app:jdbc:asi">
                                <pasi:Ownership>false</pasi:Ownership>
                        </pasi:JDBCAttributeTypeMetadata>
                        </appinfo>
                        </annotation>
                        </element>
```

関連資料

31 ページの『属性に関するアプリケーション固有情報』
このトピックでは、属性に関するアプリケーション固有情報 (ASI) について説明し、サポートされるパラメーターとその説明をリストします。

28 ページの『動詞のアプリケーション固有情報』
アダプターは、ビジネス・オブジェクトに指定されているとおり、SQL ステートメントのグループである SQL クエリーまたはストアード・プロシージャーを使用してデータベース表を更新します。このセクションでは、ストアード・プロシージャーとストアード・プロシージャー定義のエレメントについて説明します。ストアード・プロシージャー定義のサンプルも含まれています。

接続プロパティーの設定

アダプター・プロジェクトを作成した後、エンタープライズ・サービス・ディスカバリー・ウィザードを Adapter for JDBC 用に初期化して、ご使用のデータベース・インスタンス用に接続プロパティーの値を設定する必要があります。

1. エンタープライズ・サービス・ディスカバリーを初期化する

   WebSphere Integration Developer で、「**ビジネス・インテグレーション**」パースペクティブに移動します。 JDBC コネクター・プロジェクトが強調表示されている「**ビジネス・インテグレーション**」タブのペインを右クリックします。ポップアップ・メニューで、「**新規**」>「**エンタープライズ・サービス・ディスカバリー**」を選択します。

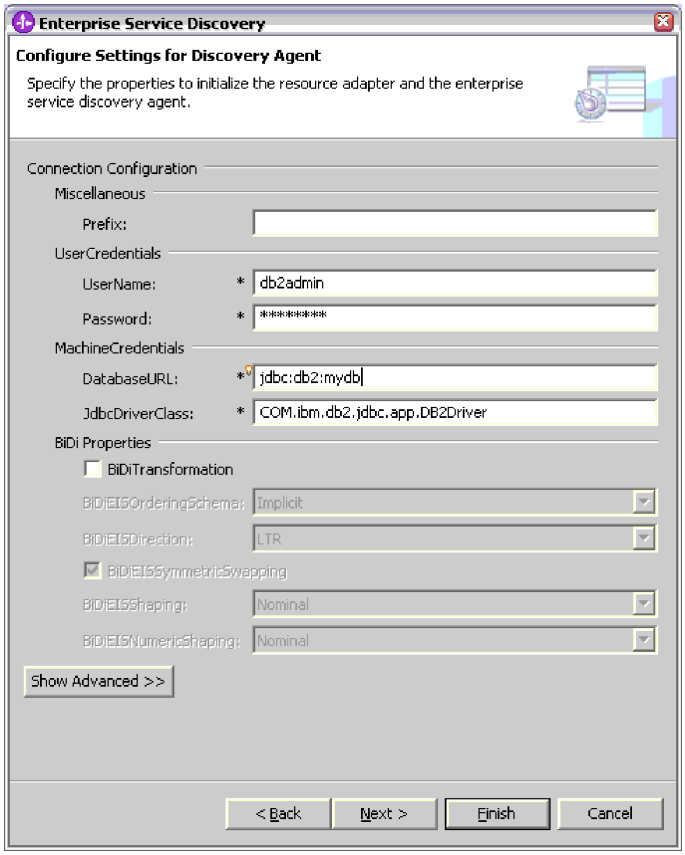
   「エンタープライズ・サービス・リソース・アダプターを選択 (Select an Enterprise Service Resource Adapter)」ウィンドウで、ご使用のアダプターのオプションを選択して「**次へ**」をクリックします。まだ RAR ファイルをインポートしていない場合は、このウィンドウの「**リソース・アダプターのインポート (Import Resource Adapter)**」をクリックしてインポートします。

2. 接続プロパティーの値を設定する

   ディスカバリーおよびサービス記述作成のため、ターゲット EIS インスタンスへの接続に使用するメタデータ・ディスカバリー接続プロパティーの値を設定する必要があります。双方向スクリプトデータ処理を使用可能にする場合は、双方向変換を活動状態にして双方向プロパティーの値を設定する必要があります。

   次の「ディスカバリー・エージェントの設定の構成 (Configure Settings for Discovery Agent)」ウィンドウで、接続構成プロパティーの値を入力します。これらのプロパティーの詳細については、『参照』セクションの『メタデータ・ディスカバリー接続プロパティー』を参照してください。

   双方向機能を活動状態にするには、「**BiDi 変換**」の横にあるチェック・ボックスを選択します。次に、双方向プロパティーの値を設定します。これらのプロパティーの詳細については、『参照』セクションの『双方向接続プロパティー』を参照してください。



ディスカバリー・エージェント・ウィンドウの設定を構成する

3. ロギング・オプションを選択する

   「**詳細表示 (Show Advanced)**」をクリックすると、「**ロギング・オプション (Logging options)**」が表示されます。「**ロギング・オプション (Logging options)**」は、エンタープライズ・サービス・ディスカバリー処理のロギングとトレースのセットアップにのみ使用されます。ただし、ロギング・レベルとトレ

ース・レベルは、アダプターのものと同じです。アダプターのロギング・レベルとトレース・レベルの詳細については、『ロギングの可能化』および『トレースの使用可能化』を参照してください。

「**ロギング・オプション (Logging options)**」 で、ログ・ファイルの出力ロケーションを入力するか、参照します。エンタープライズ・サービス・ディスカバリーのロギング・レベルとトレース・レベルを選択します。「**次へ**」をクリックします。

ディスカバリー・サービスは、接続プロパティーを使用して、オブジェクトの選択およびナビゲーション用に表示されるメタデータ・ツリーを作成します。

関連タスク

63 ページの『ロギングの可能化』
WebSphere(R) Adapter for JDBC は、イベント処理の状況を 判別するために表示できるログ・ファイルを保守します。ログ・ファイルでは、アダプターに関連するすべてのイベントおよびエラーが、 ログ項目ごとの日付、時刻、イベントとともに追跡されます。 アダプターは、エラーまたは警告条件が発生したときにエラー・メッセージをログに記録する ので、ログ・ファイルは問題のトラブルシューティングを開始するよいソースになります。

65 ページの『トレースの使用可能化』
トレースは、アダプター・ログ・ファイルで どのレベルのエラーまたは警告を取り込むかを決定します。 トレース・レベルを定義することによって、アダプター処理に関係するメッセージをトレースできます。

関連資料

93 ページの『メタデータ・ディスカバリー接続プロパティー』
エンタープライズ・サービス・ディスカバリー処理には、ディスカバリーのエンタープライズ情報システム (EIS) への接続やサービス記述の作成にこれらのプロパティーが必要となります。

94 ページの『双方向接続プロパティー』
これらのプロパティーを指定するとエンタープライズ・サービス・ディスカバリー・ウィザードでエンタープライズ情報システム (EIS) に渡すデータに適切な双方向変換を適用できます。

データベース・オブジェクトの照会

接続プロパティーの構成後、データベース・オブジェクトに対する照会を実行できます。エンタープライズ情報システム (EIS) 内のオブジェクトの構造を理解するためにメタデータ・ツリー構造をブラウズして、サービス記述に必要なオブジェクトを選択することができます。

照会を実行する前に、フィルター・プロパティーを指定して、ツリー構造内に表示されるスキーマ、ノード、またはオブジェクトのリストを絞り込むことができます。

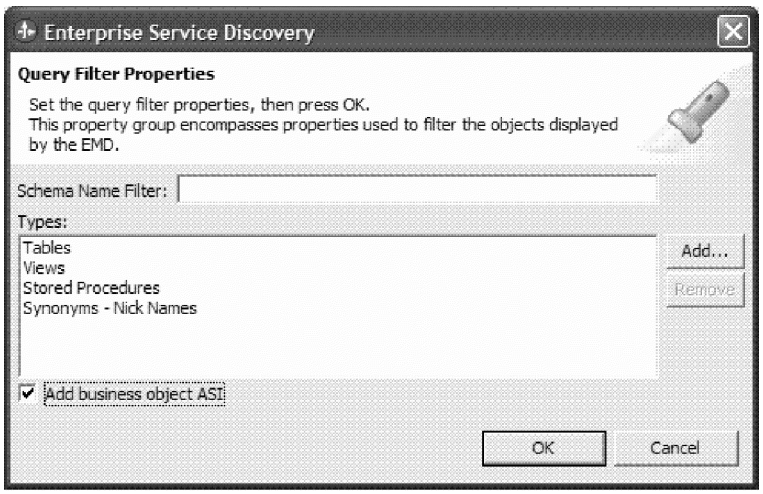
1. フィルター・プロパティーの指定

   「エンタープライズ・サービスの検索とディスカバー (Find and Discover Enterprise Services)」ウィンドウで、「**照会の編集 (Edit Query)**」をクリックします。「フィルター・プロパティーの照会 (Query Filter Properties)」ポップアッ

プ・ウィンドウで、 「**スキーマ名フィルター**」プロパティー・フィールドにテキストを入力します。指定されたストリングで始まるスキーマが表示されます。使用するスキーマを選択します。

「**タイプ**」プロパティー・フィールドに、テーブル、ビュー、ストアード・プロシージャー、同義語/ニックネームなどの項目がリストされます。そのリストからノードを追加または除去することができます。

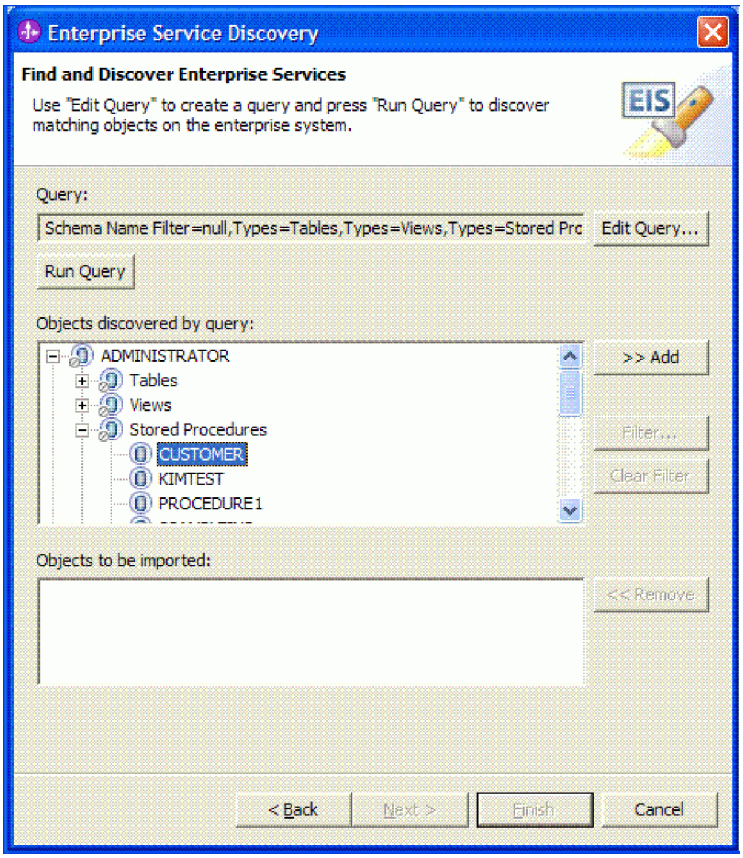
「フィルター・プロパティーの照会 (Query Filter Properties)」ウィンドウで、「**ビジネス・オブジェクト ASI の追加**」にチェック・マークを付けることができます。そうすると、ステップ 2 でメタデータ照会を実行したときにオブジェクトを追加するたびに、アプリケーション固有の情報を入力するための「*(オブジェクトの名前)* の構成パラメーター (Configuration Parameters for *(name of object)*)」というウィンドウが表示されます。



「フィルター・プロパティーの照会 (Query Filter Properties)」ウィンドウ

2. メタデータ照会を実行する
   a. 照会によって発見されたオブジェクトの表示

      次に示す「エンタープライズ・サービスの検索とディスカバー(Find and Discover Enterprise Services)」ウィンドウで、「**照会の実行 (Run Query)**」をクリックします。上部のペインにオブジェクトが表示されます。



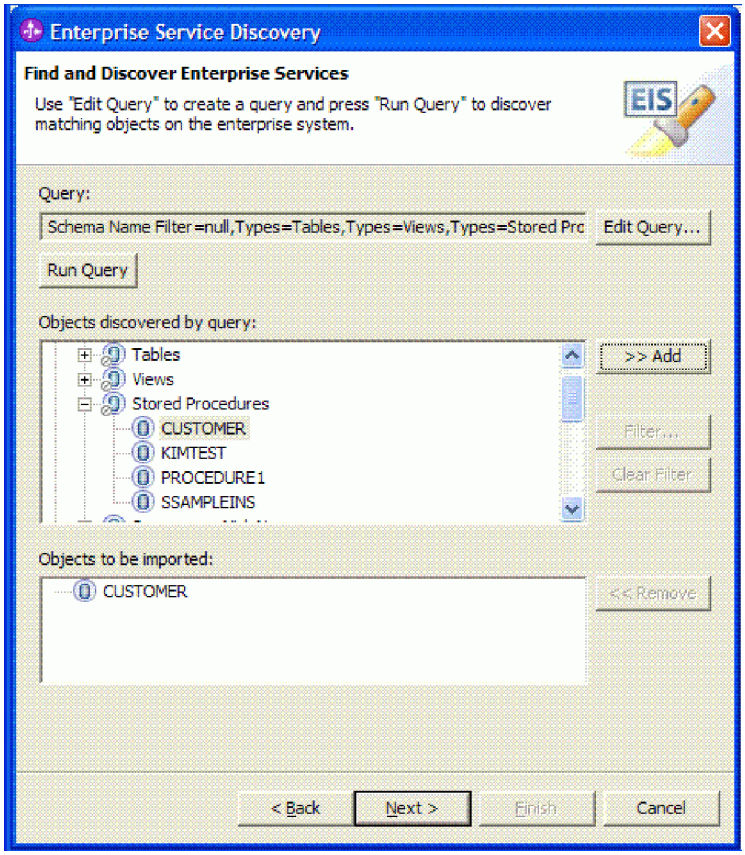
「エンタープライズ・サービスの検索とディスカバー (Find and Discover Enterprise Services)」ウィンドウ

b. オブジェクトのフィルター

スキーマのフィルタリングと同様の方法で、オブジェクトをフィルタリングすることができます。テーブル、ビュー、ストアード・プロシージャー、または同義語/ニックネーム・ノードを選択します。「**フィルター (Filter)**」をクリックします。エンタープライズ・サービス・ディスカバリー・ウィザードによって、そのノードに対して表示するデータベース・オブジェクトのリストをフィルタリングするための「**オブジェクト名フィルター**」が照会されます。テキストで入力すると、指定されたストリングで始まるデータベース・オブジェクトが表示されます。

c. インポートするオブジェクトの選択

オブジェクトを強調表示し、「**追加**」をクリックしてインポートするオブジェクトを選択します。選択したオブジェクトが下部ペインに表示されます。選択したオブジェクトを除去するには、それを強調表示して「**除去 (Remove)**」をクリックします。



インポート対象として CUSTOMER オブジェクトが選択された状態

d. ビジネス・オブジェクト ASI の追加

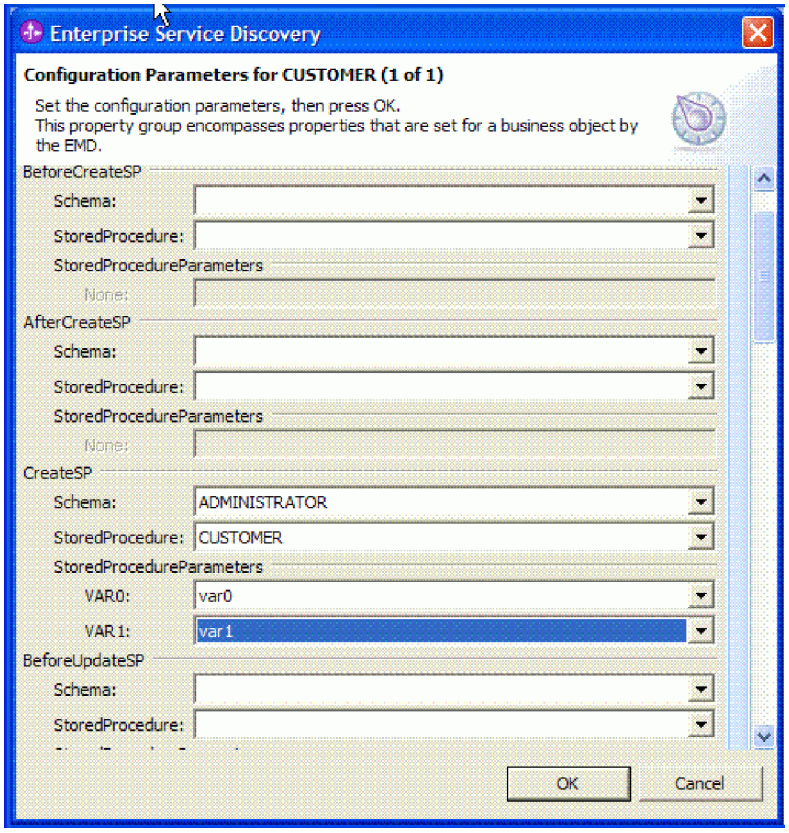
「フィルター・プロパティーの照会 (Query Filter Properties)」ウィンドウで「**ビジネス・オブジェクト ASI の追加**」を選択すると、オブジェクトを追加するたびに、アプリケーション固有の情報を入力するための「*(オブジェクトの名前)* の構成パラメーター (Configuration Parameters for *(name of object)*)」というウィンドウが表示されます。このウィンドウは、ここで、**CUSTOMER** オブジェクトの動詞に関するアプリケーション固有の情報パラメーターとともに示されています。

ストアード・プロシージャーを関連付ける動詞を選択します。例えば、ユーザーのストアード・プロシージャーによってテーブルにレコードを作成する場合は、**CreateSP** 構成パラメーターの詳細を入力します。「**スキーマ**」名、「**StoredProcedure**」名、および「**StoredProcedureParameters**」の値を入力します。

ASI の入力を完了したら、「**OK**」をクリックします。

複数の追加オブジェクトを選択した場合は、最初のオブジェクトのウィンドウが表示されます。ASI を入力して「**OK**」をクリックすると、次のオブジェクトのウィンドウが表示されます。オブジェクト・レベル、動詞、および属性に関するアプリケーション固有情報については、『アプリケーション固有の情報』を参照してください。

「**次へ**」をクリックします。



「CUSTOMER の構成パラメーター (Configuration Parameters for CUSTOMER)」ウィンドウ

関連タスク

76 ページの『シナリオ 2 の配置と構成』
シナリオ 2 では、アダプター接続プロパティーを設定し、 ビジネス・オブジェクトを生成します。プロジェクトを Enterprise Application Archive (EAR) ファイルにエクスポートし、 アプリケーション・サーバーに配置して、構成プロパティーをリセットします。

関連資料

26 ページの『アプリケーション固有情報』
ビジネス・オブジェクト定義内のアプリケーション固有情報は、アダプターに対し、ビジネス・オブジェクトの処理方法に関するアプリケーション依存の指示を与えるものです。アダプターでは、ビジネス・オブジェクトの属性または動詞、あるいはビジネス・オブジェクト自体から取得したアプリケーション固有情報を解析して、Create、Update、Retrieve、および Delete 操作のための照会を生成します。

選択プロパティーの設定

データベース・オブジェクトを選択した後、インポート・ファイルとエクスポート・ファイルの選択プロパティーの値を指定する必要があります。

選択プロパティーの詳細については、『参照』セクションを参照してください。

1. ネーム・スペースを指定する

   ネーム・スペースは、最初はすべてのビジネス・オブジェクトでデフォルトのネーム・スペースに設定されています。このデフォルト値は、「オブジェクトの構成 (Configure objects)」ウィンドウに表示されます。これは、メタデータ・スキーマ JDBCASI.xsd のネーム・スペースです。

   ネーム・スペースは、ビジネス・オブジェクト・スキーマを論理的に分離するため、ビジネス・オブジェクト名の前に付加されます。
2. サービス・タイプを選択する

   サービス・タイプとして「**Inbound**」または「**Outbound**」を選択します。
3. 操作を選択する

   「オブジェクトの構成 (Configure objects)」ウィンドウの「**操作**」フィールドに、選択したサービス・タイプに対してアダプターがサポートしている操作がリストされます。これらの操作のリストを追加する場合は、「**追加**」ボタンをクリックします。例えば、操作を削除し、後でそれを組み込む場合は次のようにします。「追加」ウィンドウで、操作のリストから選択し、「**OK**」をクリックします。完了したら、「**次へ**」をクリックします。

   指定された操作は、生成されるすべてのビジネス・オブジェクトに対して設定されます。
4. レコード最大数を設定する

   RetrieveAll 操作で検索するレコードの最大数を入力します。
5. BO ロケーションを指定する

   生成された .xsd ファイルが保管されるロケーションへのパスを入力します。

   関連タスク

   96 ページの『選択プロパティー』
   データベース・オブジェクトを選択した後、 選択プロパティーの値を設定する必要があります。

   76 ページの『シナリオ 2 の配置と構成』
   シナリオ 2 では、アダプター接続プロパティーを設定し、 ビジネス・オブジェクトを生成します。プロジェクトを Enterprise Application Archive (EAR) ファイルにエクスポートし、 アプリケーション・サーバーに配置して、構成プロパティーをリセットします。

アダプター・プロジェクトの保管

選択プロパティーを指定したら、特定のデータベースへの通信チャネルをセットアップするためにアダプターが使用するプロパティーを構成します。これらには、リソース・アダプター、J2C 接続ファクトリー、J2C アクティブ化仕様、および双方向プロパティーが含まれます。また、すべての成果物およびプロパティー値を保管できる新規ビジネス・インテグレーション・モジュールを作成する必要があります。

1. 新規モジュール名を指定する

以下に示されている「成果物の生成 (Generate artifacts)」ウィンドウで、「**モジュール (Module)**」フィールドにモジュール名が表示されていない場合は、「**新規 (New)**」をクリックする必要があります。「新規モジュール (New Module)」ポップアップ・ウィンドウで、モジュール名を入力して「**完了 (Finish)**」をクリックします。

2. サービス記述のフォルダーを指定する

   「成果物の生成 (Generate Artifacts)」ウィンドウで、新規モジュール内でサービス記述を保管するフォルダーを指定または参照します。

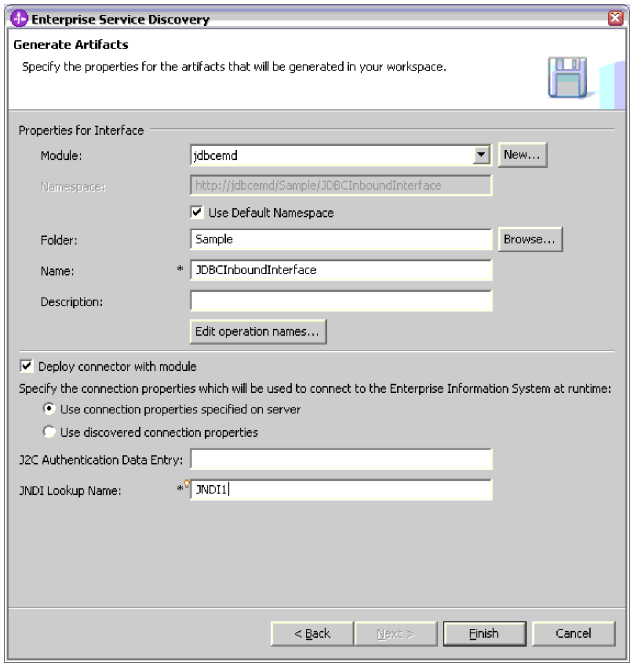
   選択したサービス・タイプに応じて、「**名前 (Name)**」フィールドにインポートまたはエクスポートのファイル名が表示されます。「**記述 (Description)**」フィールドにコメントを追加することもできます。

   「**操作の編集 (Edit Operations)**」をクリックすると、ポップアップ・ウィンドウに、作成しているビジネス・オブジェクトの全オペレーションのデフォルト名が表示されます。デフォルト名は、オペレーション名とビジネス・オブジェクト名の組み合わせからなります。

   注: 「コネクターをモジュールとともに配置 (Deploy connector with module)」 チェック・ボックスをオンにして、コネクター・プロジェクト RAR ファイルが、アプリケーション・サーバーに配置する EAR ファイルに組み込まれるようにする必要があります。

3. 構成プロパティー値を設定する

   プロジェクトの配置後、新規ビジネス・オブジェクトの作成が必要になったときに、オプション「**サーバーで指定された接続プロパティーを使用 (Use connection properties specified on server)**」を使用することもできます。これは、アプリケーション・サーバーに既にあるプロパティーを使用することを示します。「**J2C 認証データ・エントリー (J2C Authentication Data Entry)**」 フィールドは、アプリケーション・サーバーで既に設定されている認証別名を指定するために使用します。「**JNDI ルックアップ名 (JNDI Lookup Name)**」は、アプリケーション・サーバーで使用する接続ファクトリーに名前を付けるために使用します。



「成果物の生成 (Generate Artifacts)」ウィンドウ

プロパティー値を設定するには、「**ディスカバーされた接続プロパティーを使用 (Use discovered connection properties)**」を選択します。

Inbound 処理の場合は、J2C アクティブ化仕様およびリソース・アダプター・プロパティーのプロパティー・フィールドが表示されます。 Outbound 処理の場合は、J2C 接続ファクトリーおよびリソース・アダプター・プロパティーのプロパティー・フィールドが表示されます。双方向変換を活動化してある場合は、Inbound 処理または Outbound 処理の双方向プロパティー・フィールドが表示されます。必須のプロパティー・フィールドにはアスタリスクが付いています。これらの構成プロパティーの詳細については、『参照』セクションを参照してください。

重要: ここでエンタープライズ・サービス・ディスカバリー・ウィザードを使用して J2C アクティブ化仕様プロパティーを指定すると、プロパティー設定を後で変更することはできなくなります。アプリケーション・サーバーで新規アプリケーションとしてプロジェクトを配置した後に、WebSphere Process Server 管理コンソールを使用して J2C アクティブ化仕様プロパティーを更新することはできません。ただし、管理コンソールを使用してその他の構成プロパティーを変更することはできます。プロパティーの更新を試みても、アダプターは更新した値を認識しません。アプリケーション・サーバーにプロジェクトをインストールした後に J2C アクティブ化仕様プロパティーを設定したい場合は、これらのプロパティー値を設定しないでください。

関連資料

82 ページの『構成プロパティー』
このプロパティー・グループには、 特定のデータベース・アプリケーションへの通信チャネルをセットアップするために アダプターが使用する属性が含まれています。

参照バインディングの生成

参照バインディングは、その他の WebSphere(R) Business Integration Service Component Architecture (SCA) コンポーネントがアダプターにアクセスするために使用します。その他のサーバー・プロセスへアダプターをリンクしてプロジェクト・モジュールからアダプターへの参照を作成します。

参照バインディング生成は、テスト環境でのみ必須です。実稼働環境でアダプターを配置する際には必要ありません。

1. SCA コンポーネントを作成する

   WebSphere Integration Developer の「**ビジネス・インテグレーション**」パースペクティブに移動します。「**ビジネス・インテグレーション**」タブで、「**JDBCEMD**」モジュールを右クリックし、「**開く (Open With)**」>「**アセンブリー・エディター (Assembly Editor)**」を選択します。

   「アセンブリー・ダイアグラム (Assembly Diagram)」ウィンドウが表示され、モジュールのインポート・コンポーネントがビューに示されます。新規コンポーネントを作成するには、ウィンドウの左側 (縦) のフレームで 2 番目のアイコン (インポート) をクリックします。アイコンの新規メニューが表示されます。

   アイコンの新規メニューで下のアイコンをクリックします (これには、「**スタンドアロン参照 (Standalone References)**」という吹き出しヘルプがあります)。カーソルが配置アイコンに変わります。

   パレットをクリックして、新規コンポーネントを「アセンブリー・ダイアグラム (Assembly Diagram)」ウィンドウに追加します。
2. スタンドアロン参照を作成する

   モジュールのインポート・コンポーネントをクリックして新規コンポーネントにドラッグします。これにより、インポート・コンポーネントから新規コンポーネントへ線が引かれ、「線の追加 (Add Wire)」ウィンドウが表示されます。

   「線の追加 (Add Wire)」ウィンドウで、「**OK**」をクリックします。別の「線の追加 (Add Wire)」ウィンドウが表示され、WSDL インターフェースを Java に変換するかどうかが尋ねられます。「**いいえ**」をクリックします。

   「アセンブリー・ダイアグラム (Assembly Diagram)」ウィンドウに、新規「**スタンドアロン参照 (Standalone Reference)**」コンポーネントが、モジュールのインポート・コンポーネントに接続する「線」とともに表示されます。

   「**ファイル**」>「**保管**」をクリックしてアセンブリー・ダイアグラムを保管します。詳細については、http://www.ibm.com/software/integration/wid にある「WebSphere Integration Developer のユーザー・ガイド」を参照してください。

アダプター・プロジェクトの配置

プロジェクト・ファイルは、WebSphere(R) Integration Developer のワークスペース内の J2EE(TM) コネクター・プロジェクトです。これを Enterprise Application Archive (EAR) ファイルとしてローカル・ファイル・システムにエクスポートする必要があります。次に、アプリケーション・サーバーにプロジェクトの EAR ファイルをアップロードおよびインストールする前に JCA コネクター・セキュリティーの認証情報を提供する必要があります。

1. EAR ファイルにプロジェクトをエクスポートする

   WebSphere Integration Developer で、EAR ファイルにプロジェクトをエクスポートする必要があります。「選択」ウィンドウの「エクスポート」パネルで、リストから「**EAR ファイル**」を選択し、「**次へ**」をクリックします。

   「EAR エクスポート」ウィンドウの「**Ear プロジェクト**」フィールドで、ビジネス・インテグレーション・モジュールを選択します。これにより、モジュール名 **JDBCEMD** にサフィックス **App** が付けられます。 EAR ファイルを作成するロケーションを入力するか、ブラウズします。すべてのチェック・ボックスをクリックし、EAR ファイルで作成したものをすべて組み込みます。「**完了 (Finish)**」をクリックします。

2. 認証情報を提供する

   WebSphere Process Server 管理コンソールを開きます。左方のパネルで、「**セキュリティー**」>「**グローバル・セキュリティー**」をクリックします。「グローバル・セキュリティー」ウィンドウの「構成」ペインで、右方の「**認証**」見出しに移動します。「**JAAS 構成**」をクリックします。次に、「**J2C 認証データ**」リンクをクリックします。

   ウィンドウ「グローバル・セキュリティー」> 「J2EE Connector Architecture (J2C) 認証データ・エントリー」で、 「**新規**」をクリックします。「**別名**」フィールドに別名を入力します。データベースに接続できるユーザー ID とパスワードを入力します。「**OK**」をクリックします。「**保管**」をクリックして認証情報を保管します。

3. アプリケーション・サーバーに EAR ファイルをインストールする

   管理コンソールを使用して、左側のパネルで「**アプリケーション**」> 「**新規アプリケーションのインストール**」をクリックします。「**参照**」をクリックして EAR ファイルを選択し、「**次へ**」をクリックします。

   「**ステップ 7. リソース参照をリソースにマップ**」が表示されるまで、「**次へ**」をクリックし続けます。前に作成した認証別名を選択し、「**デフォルト・メソッドの使用 (Use default method)**」のチェック・ボックスをクリックして「**適用**」をクリックします。

   「インストール・オプションの要約」ペインが表示されるまで、以降のウィンドウで「**次へ**」または「**続行**」をクリックし続けます。「**完了 (Finish)**」をクリックします。メッセージ「アプリケーション *name* は正常にインストールされました (Application *name* installed successfully)」が表示されます。

「**マスター構成に保管**」をクリックして変更を保管します。次に、「エンタープライズ・アプリケーション」ペインで、「**保管**」をクリックします。

関連タスク

38 ページの『アダプター用のプロジェクトの作成』
アダプター配置の最初のタスクは、 アダプター用の J2EE(TM) コネクター・プロジェクトの作成です。Adapter for JDBC のリソース・アダプター・アーカイブ (RAR) ファイルは、WebSphere(R) Integration Developer にインポートする必要があります。 これにより、WebSphere Integration Developer 内のワークスペースにプロジェクトがセットアップされます。

サーバーでのアダプターの構成

アダプター・プロジェクトのプロジェクト Enterprise Application Archive (EAR) ファイルがアプリケーション・サーバーにインストールされたら、必要に応じて、特定のデータベース・アプリケーションへの通信チャネルをセットアップするためにアダプターが使用するプロパティーを再構成することができます。それから、構成済みのアダプター・アプリケーションを開始できます。

通常、構成プロパティーは、アダプター・プロジェクトの作成時にエンタープライズ・サービス・ディスカバリー・ウィザードを使用して設定されます。
WebSphere(R) Process Server 管理コンソールを使用してプロパティーを再設定するか、エンタープライズ・サービス・ディスカバリー・ウィザードで設定した値に従ってプロパティーが取り込まれているかどうかを確認します。

1. アダプター・プロジェクトを開く

   WebSphere Process Server 管理コンソールで、「構成」ペインにアダプター・プロジェクトの名前が表示されます。「**関連項目**」見出しの下で「**コネクター・モジュール**」を選択します。プロジェクト RAR ファイル名が表示されます。ファイル名の横にあるチェック・ボックスをクリックします。

   「**追加プロパティー**」の下で「**リソース・アダプター**」を選択します。

2. プロパティーを再構成する
   a. J2C 接続ファクトリー・プロパティー値を編集する

      管理コンソールで、アダプター・プロジェクトが「**一般プロパティー**」>「**名前**」 の下に表示されます。「**追加プロパティー**」の下のウィンドウの右方で、「**J2C 接続ファクトリー**」をクリックします。

      アダプター・プロジェクト名が、EJB プロジェクトで指定した JNDI 名とともに表示されます。アダプターのファクトリー接続の横にあるチェック・ボックスをクリックします。

      J2C 接続ファクトリーのプロパティーと値が表示されます。必要に応じてプロパティーの値を編集します。これらのプロパティーの詳細については、『J2C 接続ファクトリー・プロパティー』を参照してください。

   b. J2C アクティブ化仕様プロパティーの値を設定する

      これらの値をまだ設定していない 場合は、ステップ 2a に従い、必要に応じて J2C アクティブ化仕様プロパティーの値を設定します。

重要: エンタープライズ・サービス・ディスカバリー・ウィザードを使用して J2C アクティブ化仕様プロパティーを既に指定してある場合、プロパティー設定は変更できません。アプリケーション・サーバーで新規アプリケーションとしてプロジェクトを配置した後に、WebSphere Process Server 管理コンソールを使用して J2C アクティブ化仕様プロパティーを更新することはできません。設定の更新を試みても、アダプターは更新した値を認識しません。アプリケーション・サーバーにプロジェクトをインストールする前に J2C アクティブ化仕様プロパティーを設定しなかった場合は、これらのプロパティー値を設定できます。

これらのプロパティーを設定するには、「**追加プロパティー**」の下で「**J2C アクティブ化仕様 (J2C activation specifications)**」をクリックします。これらのプロパティーの詳細については、 『J2C アクティブ化仕様プロパティー』を参照してください。

c. リソース・アダプター・プロパティー値を編集する

「**追加プロパティー**」の下で、「**カスタム・プロパティー**」をクリックします。リソース・アダプター・プロパティーとその値のリストが表示されます。必要に応じてプロパティーを編集します。『リソース・アダプター・プロパティー』でプロパティーとその説明を参照してください。

注: 「**カスタム・プロパティー**」リストのアダプター・レベルで 「**データベース・ベンダー**」プロパティーを探します。このプロパティーは、アプリケーションを開始するために必要です。このプロパティーがブランクの場合は値を入力して保管します。

3. アダプター・アプリケーションを開始する

管理コンソールの左方のパネルで、「**アプリケーション**」>「**エンタープライズ・アプリケーション**」 をクリックします。「**エンタープライズ・アプリケーション**」ペインで、アプリケーション名の横にあるチェック・ボックスをクリックして、「**開始**」をクリックします。アプリケーションが正常に開始されたことを示すメッセージが表示されます。アプリケーションの開始時に問題が発生した場合は、アダプター・ログ・ファイルでエラーの説明を確認してください。

特定のメッセージについては、IBM WebSphere Adapters インフォメーション・センターの『Messages』を参照してください。ロギングとトレースの詳細については、『トラブルシューティング』を参照してください。

関連タスク

61 ページの『トラブルシューティング』
問題について報告する必要がある場合は、このセクションの指示に従って、IBM(R) ソフトウェア・サポートに連絡することができます。イベント処理の状況は、アダプターのロギングを使用可能にすることによって判別できます。また、トレース・レベルを設定することにより、アダプター・ログ・ファイルに取り込まれるエラーまたは警告のレベルを決定できます。このセクションの指示では、アダプターの Common Event Infrastructure を使用可能にする方法について説明しています。またこのセクションでは、メインフレーム・データ・アクセスについても説明します。

関連資料

87 ページの『J2C 接続ファクトリー・プロパティー』
J2C 接続ファクトリー・プロパティーは、ターゲットのエンタープライズ情報システム (EIS) インスタンスを構成するのに使用します。これらのプロパティーは、Outbound 処理に 影響を与え、J2EE(TM) Connector Architecture 仕様の ManagedConnectionFactory インターフェースに対応しています。

90 ページの『J2C アクティブ化仕様プロパティー』
J2C アクティブ化仕様プロパティーは、メッセージ・エンドポイントを活動化します。これらのプロパティーは、J2EE(TM) Connector Architecture 仕様の ActivationSpec インターフェースに対応しています。

83 ページの『リソース・アダプター・プロパティー』
これらの構成プロパティーは、リソース・アダプター・レベルで定義されます。

トラブルシューティング

問題について報告する必要がある場合は、このセクションの指示に従って、IBM(R) ソフトウェア・サポートに連絡することができます。イベント処理の状況は、アダプターのロギングを使用可能にすることによって判別できます。また、トレース・レベルを設定することにより、アダプター・ログ・ファイルに取り込まれるエラーまたは警告のレベルを決定できます。このセクションの指示では、アダプターの Common Event Infrastructure を使用可能にする方法について説明しています。またこのセクションでは、メインフレーム・データ・アクセスについても説明します。

関連タスク

59 ページの『サーバーでのアダプターの構成』
アダプター・プロジェクトのプロジェクト Enterprise Application Archive (EAR) ファイルが アプリケーション・サーバーにインストールされたら、必要に応じて、 特定のデータベース・アプリケーションへの通信チャネルをセットアップするために アダプターが使用するプロパティーを再構成することができます。それから、構成済みのアダプター・アプリケーションを開始できます。

IBM® ソフトウェア・サポートへの連絡

製品に問題があった場合は、IBM ソフトウェア・サポートにお問い合わせください。

IBM ソフトウェア・サポートに連絡するには、会社に現在有効な IBM ソフトウェア保守契約があり、担当者に IBM への問題の送信が許可されている必要があります。必要なソフトウェア保守契約のタイプは、所有している製品のタイプによって異なります。

- IBM 分散ソフトウェア・プロダクト (Windows® または UNIX® オペレーティング・システムで稼動する DB2® および WebSphere® 製品と同様に、Tivoli®、Lotus®、Rational® 製品を含むが、これらに限定されない) の場合は、次のいずれかの方法でパスポート・アドバンテージ®に登録します。
  - **オンライン:** 米国の パスポート・アドバンテージ Web ページに移動し、「How to Enroll」をクリックします。

– **電話:** 国別の連絡先電話番号は、Web 上にある米国の IBM ソフトウェア・サポート・ハンドブックのお問い合わせ窓口ページへ移動し、該当する地域名をクリックします。

- IBM eServer™ ソフトウェア製品 (zSeries®、pSeries®、および iSeries™ 環境で稼働する DB2 および WebSphere 製品を含むが、これらに限定されない) については、IBM 営業担当員または IBM ビジネス・パートナーを通してソフトウェア保守契約を購入できます。eServer ソフトウェア・プロダクトのサポートについての詳細は、米国の IBM テクニカル・サポート・アドバンテージ Web ページを参照してください。

必要なソフトウェア保守契約のタイプがわからない場合は、米国にお住まいの場合は 1-800-IBMSERV (1-800-426-7378) に電話するか、その他の国にお住まいの場合は Web 上にある米国の IBM ソフトウェア・サポート・ハンドブックのお問い合わせ窓口ページへ移動して、該当する地域名をクリックすると、その地域でサポートを提供している担当者の電話番号が表示されます。

IBM ソフトウェア・サポートに問い合わせるには、以下の手順に従います。

- 問題のビジネス・インパクトの判別。
- 問題の説明と背景情報の収集。
- 問題を IBM ソフトウェア・サポートに送信します。

1. 問題のビジネス・インパクトの判別。 IBM に問題を報告すると、重大度レベルを尋ねられます。そのため、報告する問題のビジネス・インパクトを理解し、評価する必要があります。次の基準を使用してください。

重大度	説明
重大度 1	重大なビジネス・インパクト。プログラムを使用できず、業務に重大な影響が及んでいる。この状態は、即時のソリューションを必要とする。
重大度 2	大きなビジネス・インパクト。プログラムは使用できるが、非常に制限されている。
重大度 3	いくらかのビジネス・インパクト。プログラムは使用でき、あまり重要でない機能 (業務にとって重要でない) が使用できない。
重大度 4	最小のビジネス・インパクト。問題が業務に与える影響はほとんどなく、問題に対する回避策がとられている。

2. 問題の説明と背景情報の収集。 IBM に問題を説明する際は、できるだけ具体的に説明してください。 IBM ソフトウェア・サポート・スペシャリストが問題の解決を効果的に支援できるよう、関連するすべての背景情報を伝えてください。時間を節約するため、以下の質問に対する回答を用意しておいてください。
   - 問題が発生したときに実行していたソフトウェアのバージョンは?
   - 問題の症状に関連するログ、トレース、メッセージはありますか? IBM ソフトウェア・サポートは以下の情報を尋ねることがあります。
   - 問題は繰り返し発生しますか? その場合、どのような手順が障害につながりますか?
   - システムに変更を加えましたか? (例えば、ハードウェア、オペレーティング・システム、ネットワーキング・ソフトウェアなど)
   - この問題に対する解決方法を現在実行していますか。その場合は、問題を報告する際にそれを説明できるようにしておいてください。

3. 問題を IBM ソフトウェア・サポートに送信します。 問題を送信する 2 つの方法を次に示します。
   - **オンライン:** 米国の IBM ソフトウェア・サポート・サイトの「Submit and track problems」ページへ移動します。適切な問題送信ツールに情報を入力します。
   - **電話:** 国別の連絡先電話番号は、Web 上にある米国の IBM ソフトウェア・サポート・ハンドブックのお問い合わせ窓口ページへ移動し、該当する地域名をクリックします。

送信した問題がソフトウェアの欠陥や資料の欠落または誤りである場合は、IBM ソフトウェア・サポートによって APAR (プログラム診断依頼書) が作成されます。APAR では、問題が詳細に記述されます。

可能であれば、IBM ソフトウェア・サポートは、APAR が解決されフィックスが送信されるまで、お客様が実行できる解決方法を提供します。 IBM は、同じ問題が発生した別のユーザーが同じ解決策を利用できるよう、IBM 製品サポート Web ページで常に、解決された APAR を公開しています。

ロギングの可能化

WebSphere(R) Adapter for JDBC は、イベント処理の状況を判別するために表示できるログ・ファイルを保守します。ログ・ファイルでは、アダプターに関連するすべてのイベントおよびエラーが、ログ項目ごとの日付、時刻、イベントとともに追跡されます。アダプターは、エラーまたは警告条件が発生したときにエラー・メッセージをログに記録するので、ログ・ファイルは問題のトラブルシューティングを開始するよいソースになります。

アダプターは JDBC ドライバーを介してデータベースと通信するので、SQL 照会またはストアード・プロシージャー呼び出しを実行すると SQLException メッセージが発生する可能性があります。これらは取り込まれてログに記録され、ResourceException メッセージが生成されます。メッセージは、要求処理中に発生したエラーまたはイベント・ストア内のエラーに基づいて戻されます。イベント処理のエラーは、イベント・マネージャーによって処理されます。

注: 同じアダプター ID プロパティーを持つ同じアダプター・インスタンスに Inbound 操作と Outbound 操作の両方を作成してバインドする場合は、同じログ名を使用する必要があります。これは、アダプターを WebSphere Process Server に配置した後には、ログ・ファイルが 1 つだけしか作成されないからです。Inbound 操作と Outbound 操作が、それぞれアダプター ID の異なる別々のアダプターに属する場合は、操作ごとに別々のログ・ファイルが作成されます。

アダプターのロギングは、WebSphere Process Server 管理コンソールから使用可能にします。ロギングを使用可能にするには、以下のステップを実行します。

1. ロギングとトレースを開始する

   WebSphere Process Server 管理コンソールを開始します。管理コンソールから、**「トラブルシューティング」 → 「ログおよびトレース」**を選択します。
2. ログ詳細レベルを指定する

「**コンポーネント**」をクリックして個々のコンポーネントのログ詳細レベルを指定するか、「**グループ**」をクリックしてコンポーネントの事前定義グループのログ詳細を指定します。

3. ロギング・レベルを選択する

「ロギング・レベル」テーブルでは、管理コンソールで設定できるさまざまなロギング・レベルが説明されています。

注: 詳細レベルの下にログ・イベントを表示するには、診断トレース・サービスを使用可能にする必要があります。詳細レベルまたはそれ以上のログ・イベントは、SystemOut ログ、IBM(R) Service ログ (使用可能な場合)、または診断トレース・サービス (使用可能な場合) で表示できます。

ロギング・レベル

レベル	標識	説明
致命的	F	タスクを継続不能。コンポーネントが動作不能。
重大	E	タスクを継続不能。コンポーネントは動作可能。これには、差し迫った致命的エラーを示す (すなわち、リソースが枯渇寸前であることを強く示す) 状況も含まれる。
警告	W	潜在的なエラー、または差し迫ったエラー。これには、例えばリソース・リークの可能性など、進行性の障害を示す状況も含まれる。
監査	A	サーバー状態またはリソースに影響を与える重大なイベント。
情報	I	タスクの全体的な進行状況の概要を示す一般情報。
構成	C	構成の変更または状況。
詳細	D	サブタスクの進行状況の詳細を示す一般情報。

4. ロギング情報を保管する

「**適用**」をクリックして変更を保管します。

関連タスク

47 ページの『接続プロパティーの設定』
アダプター・プロジェクトを作成した後、 エンタープライズ・サービス・ディスカバリー・ウィザードを Adapter for JDBC 用に初期化して、ご使用のデータベース・インスタンス用に接続プロパティーの値を設定する必要があります。

トレースの使用可能化

トレースは、アダプター・ログ・ファイルでどのレベルのエラーまたは警告を取り込むかを決定します。トレース・レベルを定義することによって、アダプター処理に関係するメッセージをトレースできます。

トレース・レベルは、WebSphere(R) Process Server 管理コンソールを使用して構成できます。トレース・レベルを設定するには、以下のステップを実行します。

1. トレースを開始する

   WebSphere Process Server 管理コンソールを開始します。「**トラブルシューティング**」 → 「**ログおよびトレース**」を選択します。

2. トレース・レベルを選択する

   「トレース・レベル」テーブルでは、管理コンソールで設定できるさまざまなトレース・レベルが説明されています。

   トレース・レベル

レベル	標識	説明
詳細	1	一般トレース。エンタープライズ情報システム (EIS) への接続の確立、EIS 内のイベントのビジネス・オブジェクトへの変換 (キー値のみ)、ビジネス・オブジェクトの処理 (キー値のみ) など、アダプターによって行われる幅広いアクションを含みます。
より詳細	2	EIS へのさまざまな API 呼び出し、パラメーター、戻り値など、アダプターによって実行されている論理についてより細かい情報を提供する詳細なトレース。
極めて詳細	3	最も詳細なレベル。メソッドの入力値/終了値/戻り値を含む。完全なビジネス・オブジェクト・ダンプも含まれる。このレベルでは、デバッグに必要な詳細な情報すべてが提供される。

3. トレース情報を保管する

   「**適用**」をクリックして変更を保管します。

関連タスク

47 ページの『接続プロパティーの設定』
アダプター・プロジェクトを作成した後、 エンタープライズ・サービス・ディ

スカバリー・ウィザードを Adapter for JDBC 用に初期化して、ご使用のデータベース・インスタンス用に接続プロパティーの値を設定する必要があります。

Common Event Infrastructure (CEI) の使用可能化

このトピックでは、アダプターの Common Event Infrastructure (CEI) を使用可能にする方法を説明します。

これらのイベント定義を設定する前に、IBM WebSphere Adapters Event Definitions ファイルを CEI カタログにパブリッシュします。これを実行する方法は、WebSphere Process Server Web サイト (http://www.ibm.com/software/integration/wps) にある CEI 資料を参照してください。

1. WebSphere 管理コンソールを開始します。
2. 「**トラブルシューティング**」 → 「**ログおよびトレース**」へ移動し、ご使用のサーバー名を選択します。
3. 一般プロパティー用の多数のオプションがあります。「**ログ詳細レベルの変更 (Change Log Detail Level)**」を選択し、JCA コンポーネント用の **com.ibm.j2ca.*** を選択します。このセクションには、アダプター・タイプごとにサブコンポーネントがあります。
   - com.ibm.j2ca.flatfile.* (WebSphere Adapter for Flat Files)
   - com.ibm.j2ca.jdbc.* (WebSphere Adapter for JDBC)
   - com.ibm.j2ca.peoplesoft.* (WebSphere Adapter for PeopleSoft)
   - com.ibm.j2ca.sap.* (WebSphere Adapter for SAP)
   - com.ibm.j2ca.siebel.* (WebSphere Adapter for Siebel)
4. ご使用のアダプターと一致するコンポーネントを選択します。各アダプター・コンポーネントには、ロギング用と CEI 用の 2 つのサブコンポーネントがあります。次のとおりです。
   - *subcomponent name*.log.*adapter id*
   - *subcomponent name*.cei.*adapter id*

   例えば、com.ibm.j2ca.siebel.cei.<AdapterID1>。配置されたアダプターのインスタンスごとに、システムは別々の ID を表示します。
5. 使用可能にする CEI アダプター ID を選択します。
6. ドロップダウン・メニューから、次の項目を選択できます。
   - オフ - CEI をオフにします
   - 詳細 - CEI をオンにして「イベント・コンテンツ (Event Content)」を「空 (Empty)」に設定します。
   - より詳細 - CEI をオンにして「イベント・コンテンツ (Event Content)」を「ダイジェスト (Digest)」に設定します。
   - 極めて詳細 - CEI をオンにして「イベント・コンテンツ (Event Content)」を「フル (Full)」に設定します。
   - 「すべて (all)」- (「極めて詳細」) と同じです。

各イベント・コンテンツ・レベルの意味 (空 (Empty)、ダイジェスト (Digest)、フル (Full)) と、Common Base Event モデルおよび Common Event Infrastructure の使用

の詳細については、http://www.ibm.com/software/integration/wps の WebSphere Process Server の Web サイトで資料を参照してください。

メインフレーム・データ・アクセス

Adapter for JDBC は、IBM(R) WebSphere(R) Information Integrator Classic Federation for z/OS(R) を使用するメインフレーム・データ・アクセスをサポートします。この製品は、メインフレーム・データベースへの読み取り/書き込み接続性を備えた Web および分散アプリケーションを提供します。

この製品は、高性能な SQL 駆動アクセスと、メインフレーム・データ・ソースの統合を実現します。好みのインターネット・ツールとアプリケーションを搭載したデスクトップを使用して、メインフレームの主幹業務情報に透過的にアクセスすることができます。

IBM z/OS プラットフォームで Adapter for JDBC with IBM WebSphere Information Integrator Classic Federation を使用する際に問題が発生した場合は、IBM 製品のサポート Web ページ http://www-306.ibm.com/software/integration/wbiadapters/support で、構成要件について説明している技術情報を参照してください。

実例ファイルおよびサンプル・アプリケーションの使用

エンタープライズ・サービス・ディスカバリー処理の成果物 (Service Component Architecture (SCA) の入出力ファイル、および対応する Web サービス記述言語 (WSDL) ファイル) の例を示します。さらに、アダプターを使用して J2EE(TM) アプリケーションとエンタープライズ情報システムとの間で通信する方法を理解するのに役立つサンプル・アプリケーションも示します。

インポート、エクスポートおよび WSDL ファイルの例

Service Component Architecture (SCA) のインポートおよびエクスポート・ファイル、および対応する Web サービス記述言語 (WSDL) の例を示します。これらの成果物はエンタープライズ・サービス・ディスカバリーの処理中に作成されます。

エクスポート (Inbound) ファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<scdl:Export xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:_="http://JDBCEMD/inbound"
xmlns:eis="http://www.ibm.com/xmlns/prod/websphere/scdl/eis/6.0.0"
xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"
xmlns:wsdl="http://www.ibm.com/xmlns/prod/websphere/scdl/wsdl/6.0.0"
name="inbound/JDBCInboundInterface">
  <interfaces>
    <interface xsi:type="wsdl:WSDLPortType" portType="_:JDBCInboundInterface"/>
  </interfaces>
  <esbBinding xsi:type="eis:EISExportBinding"
dataBindingType="com.ibm.j2ca.extension.emd.runtime.WBIDataBindingImpl">
    <resourceAdapter name="JDBCEMDApp.IBM JDBC Adapter"
type="com.ibm.j2ca.jdbc.JDBCResourceAdapter">
      <properties/>
    </resourceAdapter>
    <connection type="com.ibm.j2ca.jdbc.inbound.JDBCActivationSpec"
selectorType="com.ibm.j2ca.extension.emd.runtime.WBIFunctionSelectorImpl">
      <properties>
        <BONamespace>http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc</BONamespace>
        <jdbcDriverClass>COM.ibm.db2.jdbc.app.DB2Driver</jdbcDriverClass>
        <databaseURL>jdbc:db2:somedb<databaseURL>
        <password>abcdefg</password>
        <userName>db2admin</userName>
      </properties>
    </connection>
```

```
    <methodBinding method="createDb2adminCustomer"
nativeMethod="emitCreateAfterImageDb2adminCustomer"/>
    <methodBinding method="updateDb2adminCustomer"
nativeMethod="emitUpdateAfterImageDb2adminCustomer"/>
    <methodBinding method="deleteDb2adminCustomer"
nativeMethod="emitDeleteAfterImageDb2adminCustomer"/>
  </esbBinding>
</scdl:Export>
```

インポート (Outbound) サービス記述の例

```
 <?xml version="1.0" encoding="UTF-8"?>
<scdl:Import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:_="http://JDBCEMD/outbound"
xmlns:eis="http://www.ibm.com/xmlns/prod/websphere/scdl/eis/6.0.0"
xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"
xmlns:wsdl="http://www.ibm.com/xmlns/prod/websphere/scdl/wsdl/6.0.0"
name="outbound/JDBCOutboundInterface">
  <interfaces>
    <interface xsi:type="wsdl:WSDLPortType" portType="_:JDBCOutboundInterface"/>
  </interfaces>
  <esbBinding xsi:type="eis:EISImportBinding"
dataBindingType="com.ibm.j2ca.extension.emd.runtime.WBIDataBindingImpl">
    <resourceAdapter name="JDBCEMDApp.IBM JDBC Adapter"
type="com.ibm.j2ca.jdbc.JDBCResourceAdapter">
      <properties/>
    </resourceAdapter>
    <connection type="com.ibm.j2ca.jdbc.JDBCManagedConnectionFactory"
interactionType="com.ibm.j2ca.jdbc.JDBCInteractionSpec">
      <properties>
        <databaseURL>jdbc:db2:somedb</databaseURL>
        <jdbcDriverClass>COM.ibm.db2.jdbc.app.DB2Driver</jdbcDriverClass>
        <password>abcdefg</password>
        <userName>db2admin</userName>
      </properties>
    </connection>
    <methodBinding method="createDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Create</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="updateDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Update</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="deleteDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Delete</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="retrieveDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Retrieve</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="retrieveallDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>RetrieveAll</functionName>
        </properties>
      </interaction>
    </methodBinding>
```

```
    <methodBinding method="applychangesDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>ApplyChanges</functionName>
        </properties>
      </interaction>
    </methodBinding>
  </esbBinding>
</scdl:Import>
```

WSDL ファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="CUSTOMER"
        targetNamespace="http://test/j2c/jdbc/customer"
        xmlns:tns="http://test/j2c/jdbc/customer"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:datans="http://test/j2c/jdbc/customer">
        <wsdl:types>
                <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                        <xsd:import namespace="http://test/j2c/jdbc/customer"
                                schemaLocation="CUSTOMER.xsd">
                        </xsd:import>
                </xsd:schema>
        </wsdl:types>
        <wsdl:message name="CUSTOMERRequest">
                <wsdl:part name="request" element="datans:CUSTOMER"></wsdl:part>
        </wsdl:message>
        <wsdl:portType name="Customer">
                <wsdl:operation name="updateCustomer">
                        <wsdl:input message="tns:CustomerRequest"></wsdl:input>
                        <wsdl:output message="tns:CustomerRequest"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="createCustomer">
                        <wsdl:input message="tns:CustomerRequest"></wsdl:input>
                        <wsdl:output message="tns:CustomerRequest"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="retrieveCustomer">
                        <wsdl:input message="tns:CustomerRequest"></wsdl:input>
                        <wsdl:output message="tns:CustomerRequest"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
</wsdl:definitions>
```

サンプル: データベース・アプリケーションの更新

IBM(R) WebSphere(R) Adapter for JDBC にはサンプル・ファイルが提供されています。これを使用して、データベース・アプリケーション内の特定のテーブルを更新する練習ができます。アプリケーション統合担当者用とデータ統合担当者用に、2 つの段階的なシナリオが用意されています。ユーザーの役割に応じて、ビジネス・オブジェクトの生成、アダプターの配置、J2EE(TM) アプリケーションとエンタープライズ情報システムとの通信を行うアダプターの構成の練習ができます。

サンプル・アプリケーションのシナリオ

シナリオ	説明	対象読者
シナリオ 1	• 既に生成済みの成果物が用意されていて、アダプターによるビジネス・オブジェクトの処理方法を示します。このシナリオでは、成果物を生成するのにエンタープライズ・サービス・ディスカバリー・ウィザードを使用する必要はありません。 • アプリケーション・コンポーネントを解決策にアセンブルし、テストおよび配置用にこの解決策を作成する担当者を対象とします。	アプリケーション統合担当者
シナリオ 2	• エンタープライズ・サービス・ディスカバリー・ウィザードを使用して、データベース・コンポーネントをディスカバーする方法や、アダプターによって処理されるビジネス・オブジェクトを開発する方法を示します。 • アプリケーション統合担当者と同様の職責を持ち、アプリケーション開発者が一連のデータ・ソースにアクセスできるように使用可能化する担当者を対象とします。	データ統合担当者

アプリケーション統合担当者およびデータ統合担当者の役割について詳しくは、『製品の概要』内の『対象読者』のトピックを参照してください。

関連概念

3 ページの『対象読者』
このトピック内の情報では、WebSphere Adapter 製品のユーザーを明示し、ユーザーに必要なスキルを詳しく説明します。

シナリオ 1 のプロジェクト・ファイルの取得

シナリオ 1 では、サンプルに包括的なファイル・セットが含まれているので、それらを生成する必要はありません。実際の作業環境では、エンタープライズ・サービス・ディスカバリー・ウィザードを使用して、成果物を生成しアダプターを構成する必要があります。

シナリオを使用する前に、アダプターをインストールして 『Sample』 というフォルダーからサンプル・パッケージを抽出する必要があります。

アダプターのインストールの詳細については、『IBM WebSphere Adapters のインストール』を参照してください。

プロジェクト・ファイルを取得する

『jdbc/samples/Apps』 フォルダーに移動して JDBCApp.ear を見つけます。

このプロジェクトの Enterprise Application Archive (EAR) ファイルには次のファイルが含まれています。

- デフォルトでローカル・ホスト・アプリケーション・サーバーに配置されるアダプターの構成済みインスタンス: CWYBC_JDBC.rar

- さまざまな SCA 成果物を持つサービス・コンポーネント・アーキテクチャー (SCA) モジュール :
  - JDBCInboundInterface.export
  - JDBCInboundInterface.wsdl
  - JDBCOutboundInterface.import
  - JDBCOutboundInterface.wsdl
- WSDL ファイル: CustomerInterface.wsdl
- ビジネス・オブジェクト:
  - InboundRtasserCustomer.xsd
  - InboundRtasserAddress.xsd
  - InboundRtasserCustInfo.xsd
  - OutboundRtasserCustomer.xsd
  - OutboundRtasserAddress.xsd
  - OutboundRtasserCustInfo.xsd
  - InboundRtasserCustomerBG.xsd
  - InboundRtasserAddressBG.xsd
  - InboundRtasserCustInfoBG.xsd
  - OutboundRtasserCustomerBG.xsd
  - OutboundRtasserAddressBG.xsd
  - OutboundRtasserCustInfoBG.xsd
  - InboundRtasserCustomerContainer.xsd
  - InboundRtasserAddressContainer.xsd
  - InboundRtasserCustInfoContainer.xsd
  - OutboundRtasserCustomerContainer.xsd
  - OutboundRtasserAddressContainer.xsd
  - OutboundRtasserCustInfoContainer.xsd
- データベース表を作成するための SQL スクリプト:
  - CreateCustomerTable.sql
  - CreateAddressTable.sql
  - CreateCustInfoTable.sql

次に、配置と構成のタスクを実行します。

シナリオ 2 のプロジェクト・ファイルの取得

シナリオ 2 では、サンプルに含まれているファイル・セットは最小限のものなので、エンタープライズ・サービス・ディスカバリー・ウィザードを使用して包括的な成果物セットを生成する練習ができます。

シナリオを使用する前に、アダプターをインストールして 『Sample』 というフォルダーからサンプル・パッケージを抽出する必要があります。

アダプターのインストールの詳細については、『IBM WebSphere Adapters のインストール』を参照してください。

プロジェクト・ファイルを取得する

『jdbc/samples/Apps』 フォルダーに移動して JDBCApp.ear を見つけます。

このプロジェクトの Enterprise Application Archive (EAR) ファイルには次のファイルが含まれています。これらのファイルは、エンタープライズ・サービス・ディスカバリー・ウィザードを使用して成果物を生成し、配置と使用のためアダプターを構成する際にこのシナリオで作成する成果物の例です。シナリオを完了すると、これらのファイルを表示して出力と比較することができます。ファイルには、次のようにサンプル・インポート、エクスポート、WSDL、およびビジネス・オブジェクト・ファイルが含まれています。

- JDBCOutboundInterface.import
- JDBCInbound Interface.export
- JDBCOutboundInterface.wsdl
- JDBCInboundInterface.wsdl
- InboundRtasserCustomer.xsd
- InboundRtasserAddress.xsd
- InboundRtasserCustInfo.xsd
- OutboundRtasserCustomer.xsd
- OutboundRtasserAddress.xsd
- OutboundRtasser.CustInfo.xsd
- InboundRtasserCustomerBG.xsd
- InboundRtasserAddressBG.xsd
- InboundRtasserCustInfoBG.xsd
- OutboundRtasserCustomerBG.xsd
- OutboundRtasserAddressBG.xsd
- OutboundRtasserCustInfoBG.xsd
- InboundRtasserCustomerContainer.xsd
- InboundRtasserAddressContainer.xsd
- InboundRtasserCustInfoContainer.xsd
- OutboundRtasserCustomerContainer.xsd
- OutboundRtasserAddressContainer.xsd
- OutboundrtasserCustInfoContainer.xsd

ビジネス・オブジェクトについては、『ビジネス・オブジェクトの概要』を参照してください。

次に、配置と構成のタスクを実行します。

シナリオ 1 の配置と構成

シナリオ 1 では、サンプルがすべての成果物を提供しているので、エンタープライズ・サービス・ディスカバリー・ウィザードを使用してそれらを作成する必要はありません。

サンプルでは、既に構成されているアダプターのインスタンスが提供されます。次の手順では、ご使用の環境に合わせてビジネス・オブジェクトのアプリケーション固有の情報 (ASI) を更新する必要があります。また、エンタープライズ情報システム (EIS) 内のデータベース・インスタンスに合うように構成プロパティーの値を変更する必要があります。

1. WebSphere(R) Integration Developer を開始する

   詳細については、http://www.ibm.com/software/integration/wid にある「WebSphere Integration Developer のユーザー・ガイド」を参照してください。

2. ビジネス・インテグレーション・モジュールを作成する

   WebSphere Integration Developerで、「ビジネス・インテグレーション」パースペクティブに移動します。ビジネス・インテグレーション・ペインを右クリックします。「**新規 (New)**」>「**モジュール (Module)**」をクリックします。「新規モジュール (New Module)」ウィンドウで、「**JDBC**」 を「**モジュール名 (Module Name)**」として入力します。「**完了 (Finish)**」をクリックします。

3. Resource Adapter Archive (RAR) ファイルをインポートする

   a. 「**ファイル**」>「**インポート**」をクリックします。「選択」ウィンドウで、インポート・ソースとして「**RAR ファイル**」を選択し、「**次へ**」をクリックします。これにより、WebSphere Integration Developer に CWYBC_JDBC.rar がインポートされます。

   b. 「コネクター・インポート (Connector Import)」ウィンドウの「**コネクター・ファイル (Connector file)**」フィールドで、前のステップで作成したモジュールのパスと名前を参照します。例えば、 *install dir* ¥jdbc¥deploy¥CWYBC_JDBC.rar です。チェック・ボックス 「**EAR プロジェクトへのモジュールの追加 (Add module to an EAR project)**」が選択されていることを確認します。「**EAR プロジェクト**」フィールドで、「**JDBCApp**」を選択します。「**完了 (Finish)**」をクリックします。

   c. 「ビジネス・インテグレーション」パースペクティブの「ビジネス・インテグレーション」ペインで、「**JDBC**」プロジェクトを右クリックして「**依存関係エディターを開く (Open Dependency Editor)**」を選択します。矢印をクリックして「**J2EE**」セクションを展開します。「**追加**」をクリックします。「J2EE(TM) プロジェクト選択 (J2EE (TM) Project Selection)」ウィンドウで、「**CWBC_JDBC**」を選択して「**OK**」をクリックします。メニューから、「**ファイル**」>「**保管**」を選択します。

4. Java ビルド・パスに JDBC ドライバーを追加する

   JDBC ドライバーへの参照をプロジェクトに追加する必要があります。

   「Java」パースペクティブの WebSphere Integration Developer で、「**CWBC_JDBC Connector Project**」を右クリックします。「**プロパティー**」を選択します。

   外部 JAR ファイルを追加するには、「**Java のビルド・パス (Java Build Path)**」 をクリックします。「**ライブラリー**」タブを選択し、「**外部 Jar の追加 (Add External Jars)**」をクリックします。「ファイル・システム」ウィンドウで、「**JDBC ドライバー**」に移動して JAR ファイルを選択します。

「**OK**」をクリックして変更を保管します。

5. ビジネス・インテグレーション・モジュールにビジネス・オブジェクト成果物を追加する

   JDBCApp.ear を解凍する必要があります。

   次に、エンタープライズ・アプリケーション・アーカイブ (EAR) ファイルから JAR ファイルを抽出する必要があります。 JDBCAPP.ear から、¥workspace¥JDBC へ JDBC.jar を抽出します。

6. 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変更する

   a. 「ビジネス・インテグレーション」パースペクティブで、「**JDBC**」モジュールを右クリックして「**最新表示 (Refresh)**」を選択します。

   b. 「**データ型 (Data Types)**」の横にある **+** をクリックします。ビジネス・オブジェクト **InboundRtasserCustomer.xsd** を右クリックします。

   c. 「**開く (Open With)**」> 「**ビジネス・オブジェクト・エディター (Business Object Editor)**」を選択します。「**プロパティー**」タブを選択します。「**アプリケーション情報 (Application Info)**」タブを選択します。

   d. 「ASI プロパティー」ペインで、CUSTOMER テーブルが作成されたデータベース内のスキーマの名前を使用して、jdbcasi:TableName の値を **RTASSER.CUSTOMER** から *schema*.**CUSTOMER** へ変更します。メニューから、「**ファイル**」>「**保管**」をクリックします。

   次のビジネス・オブジェクトに対して、ステップ 6a から 6d を繰り返します。

   - InboundRtasserAddress
   - InboundRtasserCustInfo
   - OutboundRtasserCustomer
   - OutboundRtasserAddress
   - OutboundRtasserCustInfo

7. 認証情報を提供する

   a. WebSphere Process Server の管理コンソールを使用して、認証別名を作成する必要があります。管理コンソールを開きます。左方のパネルで、「**セキュリティー**」>「**グローバル・セキュリティー**」をクリックします。「グローバル・セキュリティー」ウィンドウの「構成」ペインで、右方の「**認証**」見出しに移動します。「**JAAS 構成**」をクリックします。次に、「**J2C 認証データ**」リンクをクリックします。

   b. ウィンドウ「グローバル・セキュリティー」> 「J2EE Connector Architecture (J2C) 認証データ・エントリー」で、 「**新規**」をクリックします。「**別名**」フィールドに別名を入力します。データベースに接続できるユーザー ID とパスワードを入力します。「**OK**」をクリックします。次に、「**保管**」をクリックして認証情報を保管します。

8. 構成プロパティーの値を変更する

   a. ご使用のデータベース・インスタンスに合うように構成プロパティー値を変更する必要があります。 WebSphere Integration Developer の「ビジネス・インテグレーション」パースペクティブの「ビジネス・インテグレーション」

ペインで、「**JDBC モジュール**」をクリックします。
「**Inbound/JDBCInboundInterface**」をダブルクリックします。

b. アセンブリー・エディターの「**プロパティー**」タブで、「**バインディング**」タブをクリックしてから、「**接続**」タブをクリックします。

c. **ActivationSpecProperties** を展開します。データベース構成に基づいて、プロパティーであるユーザー名、パスワード、データベース URL、および JDBC ドライバー・クラスを J2C アクティブ化仕様の適切な値に設定します。

d. 「**接続**」タブで、「**認証プロパティー**」を展開します。「**J2C 認証データ・エントリー (J2C Authentication Data Entry)**」フィールドに、前のステップで作成した認証別名の名前を入力します。メニューから、「**ファイル**」>「**保管**」をクリックします。

Outbound/JDBCOutboundInterface オブジェクトに対してこの手順を繰り返す必要があります。 J2C 接続ファクトリー (管理接続ファクトリーとも呼ばれます) プロパティーであるユーザー名、パスワード、データベース URL、および JDBC ドライバー・クラスの値を更新します。

9. アプリケーション・サーバーにプロジェクト・ファイルを配置する

プロジェクトの EAR ファイルを WebSphere Integration Developer から WebSphere Process Server へエクスポートする必要があります。

a. EAR ファイルをエクスポートする

WebSphere Integration Developer メニューから、「**ファイル**」>「**インポート**」をクリックします。EAR ファイルを選択して、「**次へ**」をクリックします。「EAR エクスポート」ウィンドウで、「**EAR プロジェクト**」フィールドへ移動して「**JDBCApp**」と入力します。宛先パスとファイル名を設定します。 3 つのチェック・ボックスをすべて選択して「**完了 (Finish)**」を選択します。

b. アプリケーション・サーバーに EAR ファイルをインストールする

管理コンソールを使用して、「**アプリケーション**」> 「**新規アプリケーションのインストール**」を選択します。「**参照**」をクリックして EAR ファイルを選択し、「**次へ**」をクリックします。

「**ステップ 7. リソース参照をリソースにマップ**」が表示されるまで、「**次へ**」をクリックし続けます。前に作成した認証別名を選択し、チェック・ボックスを選択して「**適用**」をクリックします。

「インストール・オプションの要約」ウィンドウが表示されるまで、以降の画面で、「**次へ**」または「**続行**」をクリックし続けます。次に、「**終了**」をクリックします。

メッセージ「**アプリケーション SampleApp は正常にインストールされました (Application SampleApp installed successfully)**」が表示されます。「**マスター構成に保管**」をクリックして変更を保管します。次に、「**保管**」をクリックします。

これで、サンプル・アプリケーションを実行できます。

シナリオ 2 の配置と構成

シナリオ 2 では、アダプター接続プロパティーを設定し、ビジネス・オブジェクトを生成します。プロジェクトを Enterprise Application Archive (EAR) ファイルにエクスポートし、アプリケーション・サーバーに配置して、構成プロパティーをリセットします。

1. WebSphere(R) Integration Developer を開始する

   詳細については、http://www.ibm.com/software/integration/wid にある「WebSphere Integration Developer のユーザー・ガイド」を参照してください。

2. ビジネス・インテグレーション・モジュールを作成する

   WebSphere Integration Developerで、「ビジネス・インテグレーション」パースペクティブに移動します。ビジネス・インテグレーション・ペインを右クリックします。「**新規 (New)**」>「**モジュール (Module)**」を選択します。「新規モジュール (New Module)」ウィンドウで、 「JDBC」 を「**モジュール名 (Module Name)**」として入力します。「**完了 (Finish)**」をクリックします。

3. アダプター RAR ファイルをインポートする

   WebSphere Integration Developer で、「**ファイル**」>「**インポート**」を選択して、アダプターの RAR ファイルをインポートします。RAR ファイルは jdbc/deploy/CWYBC_JDBC.rar です。

   「選択」ウィンドウで、インポート・ソースとして「**RAR ファイル**」を選択します。

   「コネクター・インポート (Connector Import)」ウィンドウの「**コネクター・ファイル (Connector file)**」フィールドで、前のステップで作成したモジュールのパスと名前を参照します。例えば、 *install dir*/jdbc/deploy/CWYBC_JDBC.rar です。

   チェック・ボックス 「**EAR プロジェクトへのモジュールの追加 (Add module to an EAR project)**」 が選択されています。「**EAR プロジェクト**」フィールドで、「**JDBCApp**」を選択します。「**完了 (Finish)**」をクリックします。

4. Java ビルド・パスに JDBC ドライバーを追加する

   JDBC ドライバーへの参照をプロジェクトに追加する必要があります。

   「**Java**」パースペクティブの WebSphere Integration Developer で、「**CWBC_JDBC Connector Project**」を右クリックします。「**プロパティー**」を選択します。

   外部 JAR ファイルを追加するには、「**Java のビルド・パス (Java Build Path)**」 をクリックします。「**ライブラリー**」タブを選択し、「**外部 Jar の追加 (Add External Jars)**」をクリックします。「ファイル・システム」ウィンドウで、「**JDBC ドライバー**」に移動して JAR ファイルを選択します。

   「**OK**」をクリックして変更を保管します。

5. 認証情報を提供する

WebSphere Process Server 管理コンソールを開きます。左方のパネルで、「**セキュリティー**」>「**グローバル・セキュリティー**」を選択します。「グローバル・セキュリティー」ウィンドウの「構成」ペインで、右方の「**認証**」見出しに移動します。「**JAAS 構成**」をクリックします。次に、「**J2C 認証データ**」リンクをクリックします。

ウィンドウ「グローバル・セキュリティー」> 「J2EE Connector Architecture (J2C) 認証データ・エントリー」で、 「**新規**」をクリックします。「**別名**」フィールドに別名を入力します。データベースに接続できるユーザー ID とパスワードを入力します。「**OK**」をクリックします。「**保管**」をクリックして認証情報を保管します。

6. アダプター接続プロパティーを設定する
   a. WebSphere Integration Developer で、JDBC コネクター・プロジェクトを含む「**ビジネス・インテグレーション**」タブのフレームを右クリックします。ポップアップ・メニューで、「**新規**」>「**エンタープライズ・サービス・ディスカバリー**」を選択します。
   b. 「エンタープライズ・サービス・リソース・アダプターを選択 (Select an Enterprise Service Resource Adapter)」ウィンドウで、「**JDBC EMD アダプター (JDBC EMD Adapter)**」を選択して「**次へ**」をクリックします。
   c. 「ディスカバリー・エージェントの設定の構成 (Configure Settings for Discovery Agent)」ウィンドウで、接続構成プロパティーの各フィールドにデータベースの情報入力します。このサンプルに必要な接続プロパティー設定は 4 つだけです。これらのプロパティーの詳細については、『参照』セクションの『接続プロパティー』を参照してください。次の表に、ディスカバリー・エージェントを初期化するための例の値を示します。

メタデータ・ディスカバリー接続プロパティー

プロパティー	例の値
ユーザー名	db2admin
パスワード	xxxxxxxxxxx
データベース URL	jdbc:db2:mytestdb
JDBC ドライバー・クラス	COM.ibm.db2.jdbc.app.DB2Driver

「**次へ**」をクリックします。

7. データベース・オブジェクトを照会する

接続プロパティーの構成後、データベース・オブジェクトに対する照会を実行できます。エンタープライズ情報システム (EIS) 内のオブジェクトの構造を理解するためにメタデータ・ツリー構造をブラウズして、サービス記述に必要なオブジェクトを選択することができます。

この時点で、フィルター・プロパティーを指定して、ツリー構造に表示されるスキーマのリストを絞り込むことができます。『データベース・オブジェクトの照会』で、フィルター・プロパティーを指定する手順を参照してください。

「エンタープライズ・サービスの検索とディスカバー (Find and Discover Enterprise Services)」ウィンドウで、「**照会の実行 (Run Query)**」をクリックします。

上部のペインにメタデータ・オブジェクトが表示されます。生成元のオブジェクトを強調表示して、「**追加**」をクリックします。選択したオブジェクトが下部のペインに表示されます。選択したオブジェクトを除去するには、「**除去 (Remove)**」をクリックします。

オブジェクトの選択を完了したら、「**次へ**」をクリックします。

8. 選択プロパティーを指定する

   データベース・オブジェクトを選択した後、インポート・ファイルとエクスポート・ファイルの選択プロパティーの値を指定する必要があります。

   ネーム・スペース・プロパティーは、最初はすべてのビジネス・オブジェクトでデフォルト値に設定されています。

   サービス・タイプ (Inbound または Outbound) と対応する操作の選択については、『選択プロパティーの設定』を参照してください。これらの選択プロパティーを指定した後、「**次へ**」をクリックします。

9. アダプター・プロジェクトを保管する

   モジュール名 **JDBC** が自動的に「成果物の生成 (Generate artifacts)」ウィンドウに表示されます。チェック・ボックス設定は変更しないでください。「**新規**」をクリックして、成果物を保管する新規ビジネス・インテグレーション・モジュールを作成します。

   「新規モジュール (New Module)」ウィンドウにモジュール名が表示されます。デフォルト・フォルダー名が選択されます。これは、サービス記述が保管される場所です。

   「**J2C 認証データ・エントリー (J2C Authentication Data Entry)**」フィールドに、前に作成した認証別名の名前を入力します。

   「**完了 (Finish)**」をクリックします。

   サンプル・ファイルへの出力を比較できます。「ビジネス・インテグレーション」パースペクティブの「ビジネス・インテグレーション」ペインで、「**JDBC**」モジュールの横にある **+** をクリックして展開します。「**データ型 (Data Types)**」フォルダーの横にある **+** をクリックして、ビジネス・オブジェクトすべてのリストを表示します。

10. アプリケーション・サーバーにプロジェクト・ファイルを配置する

    プロジェクト・ファイルは、WebSphere Integration Developer のワークスペース内の J2EE コネクター・プロジェクトです。これをローカル・ファイル・システムにエクスポートする必要があります。次に、アプリケーション・サーバーに EAR ファイルをアップロードおよびインストールする前に JCA コネクター・セキュリティーの認証情報を提供する必要があります。

    a. EAR ファイルをエクスポートする

WebSphere Integration Developer の「選択」ウィンドウの「エクスポート」画面で EAR ファイル **JDBCApp** を選択し、「**次へ**」をクリックします。

「EAR エクスポート」ウィンドウで、ビジネス・インテグレーション・モジュールと、EAR ファイルを作成するロケーションを選択する必要があります。「**EAR プロジェクト**」フィールドでモジュールを選択します。 EAR ファイルを作成するロケーションを入力するか、ブラウズします。すべてのチェック・ボックスをクリックして、「**完了 (Finish)**」をクリックします。

b. アプリケーション・サーバーに EAR ファイルをインストールする

管理コンソールを使用して、左側のペインで「**アプリケーション**」> 「**新規アプリケーションのインストール (Install New Application)**」を選択します。「**参照**」をクリックして EAR ファイルを選択し、「**次へ**」をクリックします。

「**ステップ 7. リソース参照をリソースにマップ**」が表示されるまで、「**次へ**」をクリックし続けます。前に作成した認証別名を選択し、「**デフォルト・メソッドの使用 (Use default method)**」のチェック・ボックスを選択して「**適用**」をクリックします。

「インストール・オプションの要約」ウィンドウが表示されるまで、以降の画面で、「**次へ**」または「**続行**」をクリックし続けます。次に、「**終了**」をクリックします。

メッセージ「**アプリケーション SampleApp は正常にインストールされました (Application SampleApp installed successfully)**」が表示されます。「**マスター構成に保管**」をクリックして変更を保管します。次に、「エンタープライズ・アプリケーション」ペインで、「**保管**」をクリックします。

11. 構成プロパティーの値を変更する

ユーザー名、パスワード、データベース URL および JDBC ドライバー・クラスのアダプター・プロパティーは既にデフォルト値に設定されていますが、管理コンソールを使用して、ご使用のデータベース・インスタンスに合うようにそれらを再構成する必要があります。

Inbound 処理の場合は、J2C アクティブ化仕様プロパティーを変更する必要があります。 Outbound 処理の場合は、J2C 接続ファクトリー・プロパティーを変更する必要があります。このサンプルでは、追加プロパティーの値を再構成する必要はありません。また、リソース・アダプター構成プロパティーをご希望の設定に変更することもできます。これらのプロパティーの詳細については、『参照』セクションの『構成プロパティー』を参照してください。

a. アダプター・プロジェクト・ファイルを開く

管理コンソールの「エンタープライズ・アプリケーション・ウィンドウ」ウィンドウの「**構成**」タブで、ご使用のアダプター・プロジェクトの名前をクリックします。「**関連項目**」見出しの下で「**コネクター・モジュール**」を選択します。プロジェクト RAR ファイル名が表示されます。ファイル名の横にあるチェック・ボックスをクリックします。

「**構成**」タブで、「**追加プロパティー**」見出しの下の「**リソース・アダプター**」を選択します。

b. プロパティーを再構成する

ご使用のアダプター・プロジェクトが「一般プロパティー」ペインの 「**名前**」の下に表示されます。「**追加プロパティー**」の下で、J2C アクティブ化仕様、J2C 接続ファクトリー、またはカスタム・プロパティーのいずれかのプロパティー・カテゴリーのプロパティー値を編集します。カスタム・プロパティーは、リソース・アダプター・プロパティーです。

これで、サンプル・アプリケーションを実行できます。

関連タスク

49 ページの『データベース・オブジェクトの照会』
接続プロパティーの構成後、 データベース・オブジェクトに対する照会を実行できます。 エンタープライズ情報システム (EIS) 内のオブジェクトの構造を理解するために メタデータ・ツリー構造をブラウズして、 サービス記述に必要なオブジェクトを選択することができます。

53 ページの『選択プロパティーの設定』
データベース・オブジェクトを選択した後、 インポート・ファイルとエクスポート・ファイルの選択プロパティーの値を 指定する必要があります。

関連資料

93 ページの『接続プロパティー』
接続プロパティーはエンタープライズ・サービス・ディスカバリー・ウィザードで、ターゲット・エンタープライズ情報システム (EIS) インスタンスに接続するのに使用されます。

82 ページの『構成プロパティー』
このプロパティー・グループには、 特定のデータベース・アプリケーションへの通信チャネルをセットアップするために アダプターが使用する属性が含まれています。

サンプル・アプリケーションの実行

このタスクでは、アダプター・サンプル・アプリケーションを実行して、Inbound 操作中のイベントの処理をテストします。イベントから出力値の正確性を確認できます。

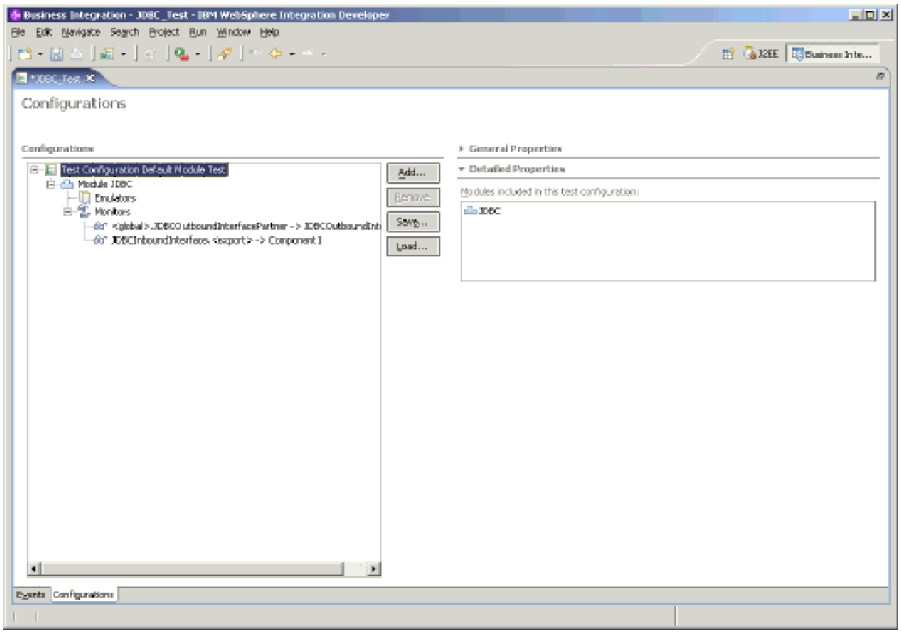
WebSphere(R) Integration Developer Test Client を使用してサンプルを実行します。

1. ビジネス・インテグレーションのモジュールおよびサーバーを選択します。

WebSphere Integration Developer の「ビジネス・インテグレーション」パースペクティブから、「ビジネス・インテグレーション」タブで「**JDBC**」モジュールを選択します。「**テスト**」>「**接続**」を右クリックします。

WebSphere Integration Developer Test Client によって、「イベント」ウィンドウが表示されます。右側で、「**デフォルト・モジュール・テスト (Default Module Test)**」が「**構成 (Configuration)**」フィールドに表示されます。「**モジュール (Module)**」フィールドに「**JDBC**」が表示されます。

左下の「**構成 (Configuration)**」タブをクリックします。「構成」ウィンドウで、「**モニター**」の横にある **+** をクリックして、モニター **JDBCInboundInterface <export> Component1** がエクスポート用に存在することを確認します。シナリオ 1 を完了した場合は、このモニター・ファイルが生成されています。シナリオ 2 を完了した場合、このファイルはステップ 9 の『アダプター・プロジェクトを保管する』で作成されています。



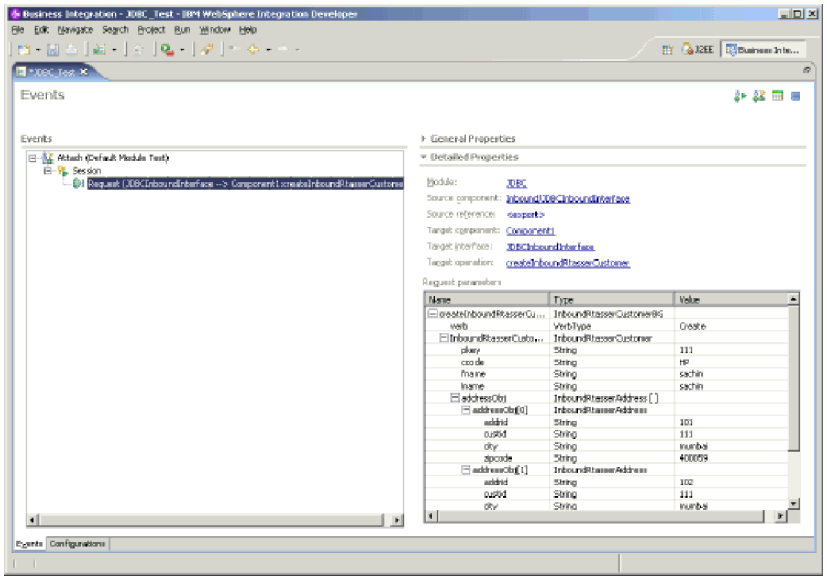
左下の「**イベント**」タブをクリックしてから、「**続行**」をクリックします。「デプロイメント・ロケーションの選択 (Select Deployment Location)」ポップアップ・ウィンドウで、アプリケーションを実行するアプリケーション・サーバーを選択します。「**WebSphere Process Server v6.0**」を選択して「**完了 (Finish)**」をクリックします。「**統合テスト・クライアントの開始 (Starting the integration test client)**」ポップアップ・ウィンドウが表示されます。

2. イベント処理をテストする

イベント処理をテストするには、サンプル・アプリケーション・イベント・テーブルにイベントを手動で挿入します。 SQL ステートメントでイベント・ストアにイベントを挿入するガイドとして、次の例を使用してください。

```
INSERT INTO wbia_jdbc_eventstore (object_key, object_name, object_function,
event_priority, event_status)
VALUES ('11234', 'InboundRtasserCustomerBG', 'Create', 1, 0)
```

モニター対象コンポーネントがイベントを受信すると、「イベント」ウィンドウに項目が表示されます。次の「イベント」ウィンドウの例を参照してください。



参照

このセクションでは、構成、接続、およびオブジェクト選択のプロパティーについて説明します。

関連概念

15 ページの『トランザクション管理』
Adapter for JDBC は、ローカル・トランザクションと XA トランザクションの両方をサポートします。

構成プロパティー

このプロパティー・グループには、特定のデータベース・アプリケーションへの通信チャネルをセットアップするためにアダプターが使用する属性が含まれています。

プロパティーのタイプは次のとおりです。

- リソース・アダプター
- J2C 接続ファクトリー
- J2C アクティブ化仕様

双方向プロパティーについては、これらのプロパティー・タイプごとに説明します。

関連タスク

54 ページの『アダプター・プロジェクトの保管』
選択プロパティーを指定したら、 特定のデータベースへの通信チャネルをセットアップするために アダプターが使用するプロパティーを構成します。 これらには、リソース・アダプター、J2C 接続ファクトリー 、J2C アクティブ化仕様、および双方向プロパティーが含まれます。また、すべての成果物およびプロパティー値を保管できる 新規ビジネス・インテグレーション・モジュールを作成する必要があります。

76 ページの『シナリオ 2 の配置と構成』
シナリオ 2 では、アダプター接続プロパティーを設定し、 ビジネス・オブジェクトを生成します。プロジェクトを Enterprise Application Archive (EAR) ファイルにエクスポートし、 アプリケーション・サーバーに配置して、構成プロパティーをリセットします。

リソース・アダプター・プロパティー

これらの構成プロパティーは、リソース・アダプター・レベルで定義されます。

『リソース・アダプター・プロパティー』の表でこれらの構成プロパティーについて説明します。双方向テキスト変換が活動化されているときに構成する必要がある双方向リソース・アダプターのプロパティーについては、このセクションの別の表で説明しています。

リソース・アダプター・プロパティー

プロパティー	型	グローバル化	説明
BO ネーム・スペース	String	いいえ	このアダプターで使用されるビジネス・オブジェクト定義のネーム・スペースです。この値は、メタデータ・ディスカバリー処理で指定された値から取得されます。このプロパティーは必須です。

プロパティー	型	グローバル化	説明
データベース・ベンダー	String	いいえ	特殊な処理の際に、アダプターが使用する RDBMS を指定します。IBM DB2、Oracle、または Microsoft SQL Server データベースを使用する場合、値は DB2、Oracle、または MSSQLServer に設定してください。異なるデータベースを使用する場合、値は該当するデータベースの名前に設定してください。デフォルト以外のデータベースを使用している場合は、適切なドライバーがロードされていることを確認してください。このプロパティーが「Others」に設定された場合、アダプターは、ドライバーを探して、使用するデータベースを決定します。アダプターで処理が正常に行われるようにするには、なんらかの値が必要です。
ログ･ファイル名	String	いいえ	ログ・ファイルの絶対パス。このプロパティーは必須です。
ログ・ファイル数	Integer	いいえ	使用するログ・ファイルの数。ログ・ファイルが最大サイズに達すると、別のログ・ファイルが開始されます。値を指定しないと、この値は 1 に設定されます。
ログ・ファイル・サイズ (LogMaxFileSize)	Integer	いいえ	ログ・ファイルのサイズ (キロバイト単位)。値を指定しないと、ファイルに最大サイズが設定されません。

プロパティー	型	グローバル化	説明
Ping クエリー	String	いいえ	データベースへの接続の有効性をテストするのに使用する SQL クエリー。
クエリー・タイムアウト	Integer	はい	これにより、すべての SQL ステートメントで照会にかかる最大時間を秒数で指定します。値を指定しない場合は、照会にタイムアウトを設定しません。照会の処理に、指定された秒数より長い時間が必要な場合は、データベースにより、キャプチャーされる SQL 例外が生成されます。関連メッセージが、ログ・ファイルに記録されます。
ReturnDummyBO ForSP	Boolean	いいえ	このプロパティーは、結果のセットが空の場合でも出力パラメーターを戻すために使用されます。RetrieveSP の場合、結果のセットが戻されます。結果のセットが空の場合は、ビジネス・オブジェクトが生成されず、プロシージャー呼び出しの戻す出力パラメーターもリトリーブできません。このプロパティー値が `true` の場合、出力の値を持つダミーのビジネス・オブジェクトと、対応する属性にデータが取り込まれた入出力パラメーターが戻されます。デフォルト値は `false` です。

プロパティー	型	グローバル化	説明
トレース・ファイル名	String	いいえ	トレース・ファイルの絶対パス。このプロパティーは必須です。
トレース・ファイル数	Integer	いいえ	使用するトレース・ファイルの数。トレース・ファイルが最大サイズに達すると、別のトレース・ファイルが開始されます。値を指定しないと、この値は 1 に設定されます。
トレース・ファイルのサイズ (TraceFileSizeMax)	Integer	いいえ	トレース・ファイルのサイズ (キロバイト単位)。値を指定しないと、ファイルに最大サイズが設定されません。

以下の表で、双方向テキスト変換が活動化されているときにのみ構成する必要があるリソース・アダプター・プロパティーについて説明します。

双方向リソース・アダプター・プロパティー

プロパティー	型	説明
BiDi コンテキスト EIS	String	ビジネス・オブジェクトのランタイム・インスタンスすべてにあるコンテンツ・データの双方向形式を定義します。
BiDiContextMetadata	String	デプロイメント記述子に保管されたメタデータまたは構成プロパティーの双方向形式を定義します。
BiDiContextSkip	Boolean	リソース・アダプター・レベルでの双方向言語サポートの呼び出しを制御するフラグ。`true` と等しければ、サポートは呼び出されません。`false` と等しければ、サポートは呼び出されます。
BiDiContextSpecialFormat	String	コネクター・プロパティーのすべてにおいて、特殊な処理を受けるプロパティーのカテゴリーを指定します。

プロパティー	型	説明
BiDiContextTurnBiDiOff	Boolean	JDBC アダプターの双方向変換の呼び出しを制御するフラグ。このパラメーターは、すべてのレベルの他の双方向パラメーターのどれよりも高い優先順位を持っています。言い換えると、この値が `true` に設定されている場合には双方向言語サポートが呼び出されず、チェックや検索も実行されません。 `false` に設定されている場合は、双方向言語サポートが呼び出されます。

関連タスク

59 ページの『サーバーでのアダプターの構成』
アダプター・プロジェクトのプロジェクト Enterprise Application Archive (EAR) ファイルが アプリケーション・サーバーにインストールされたら、必要に応じて、 特定のデータベース・アプリケーションへの通信チャネルをセットアップするために アダプターが使用するプロパティーを再構成することができます。それから、構成済みのアダプター・アプリケーションを開始できます。

J2C 接続ファクトリー・プロパティー

J2C 接続ファクトリー・プロパティーは、ターゲットのエンタープライズ情報システム (EIS) インスタンスを構成するのに使用します。これらのプロパティーは、Outbound 処理に影響を与え、J2EE(TM) Connector Architecture 仕様の ManagedConnectionFactory インターフェースに対応しています。

J2C 接続ファクトリーは、接続プールを管理します。これは、アダプターによるアプリケーションから単一の JDBC アプリケーション・インスタンスへの Outbound 接続に関する構成情報を提供します。

『J2C 接続ファクトリー・プロパティー』の表で、J2C 接続ファクトリーに関わる JDBC 構成プロパティーを定義します。このセクションにある別の 2 つの表で、双方向テキスト変換が活動化されているときにのみ構成する必要がある双方向 J2C 接続ファクトリー・プロパティーを定義します。

J2C 接続ファクトリー・プロパティー

プロパティー	型	グローバル化	説明
自動コミット	Boolean	いいえ	接続で設定される自動コミットの値。

プロパティー	型	グローバル化	説明
データベース URL	String	はい	データベースへの接続に使用されるデータベース URL。これは双方向言語で使用される場合に使用可能化されます。
JDBC ドライバー・クラス	String	いいえ	データベースへの接続に使用されるドライバーの JDBC ドライバー・クラス。
パスワード	String	はい	対応するユーザー名に対するパスワード。これは双方向言語で使用される場合に使用可能化されます。
ユーザー名	String	はい	エンタープライズ情報システムにログインするためのユーザー名。これは双方向言語で使用される場合に使用可能化されます。
XADataSourceName	String	いいえ	データベースへの XA 接続を確立するための XA データ・ソース名。
XADatabaseName	String	はい	XA 接続で使用するデータベース名。DB2 データベースを使用している場合、このプロパティーを設定する必要があります。

双方向 J2C 接続ファクトリー・プロパティー

プロパティー	型	説明
BiDi コンテキスト EIS	String	EIS への固有接続用アダプターによってサポートされるビジネス・オブジェクト・ランタイム・インスタンスすべてにあるコンテンツ・データの双方向形式を定義します。
BiDiContextMetadata	String	EIS への固有接続のメタデータまたは構成プロパティーの双方向形式を定義します。

プロパティー	型	説明
BiDiContextSkip	Boolean	リソース・アダプター・レベルでの双方向言語サポートの呼び出しを制御するフラグ。`true` と等しければ、サポートは呼び出されません。`false` と等しければ、サポートは呼び出されます。
BiDiContextSpecialFormat	String	EIS への固有接続のコネクター・プロパティーすべての場合に、特殊な処理を受けるプロパティーのカテゴリーを指定します。
データベース URL BiDi プロパティー		*DatabaseURLEIS、DatabaseURLSpecialFormat、DatabaseURLSkip
パスワード BiDi プロパティー		*PasswordEIS、PasswordSpecialFormat、PasswordSkip
ユーザー名 BiDi プロパティー		*UserNameEIS、UserNameSpecialFormat、UserNameSkip

*BiDi サポートの各プロパティーに関連付けられた 3 つの双方向 (BiDi) プロパティー。これらのプロパティーは以下の表に記述されています。

BiDi プロパティー	型	説明
BiDiContextCP<property_Name>EIS	String	EIS への固有接続の固有コネクター構成プロパティーの双方向形式を定義します。
BiDiContextCP<property_Name>SpecialFormat	String	コネクター構成プロパティー・レベルで双方向言語サポートの呼び出しを制御するフラグ。`true` と等しければ、サポートは呼び出されません。`false` と等しければ、サポートは呼び出されます。
BiDiContextCP<property_Name>Skip	Boolean	EIS への固有接続における固有コネクター・プロパティーの特殊形式を定義します。

関連概念

15 ページの『トランザクション管理』
Adapter for JDBC は、ローカル・トランザクションと XA トランザクションの両方をサポートします。

関連タスク

59 ページの『サーバーでのアダプターの構成』
アダプター・プロジェクトのプロジェクト Enterprise Application Archive (EAR)

ファイルが アプリケーション・サーバーにインストールされたら、必要に応じて、 特定のデータベース・アプリケーションへの通信チャネルをセットアップするために アダプターが使用するプロパティーを再構成することができます。それから、構成済みのアダプター・アプリケーションを開始できます。

J2C アクティブ化仕様プロパティー

J2C アクティブ化仕様プロパティーは、メッセージ・エンドポイントを活動化します。これらのプロパティーは、J2EE(TM) Connector Architecture 仕様の ActivationSpec インターフェースに対応しています。

『J2C アクティブ化仕様プロパティー』の表で、メッセージのエンドポイントの活動化に関わる JDBC アダプター固有の構成プロパティーについて説明します。このセクションの別の表で、双方向テキスト変換が活動化されているときにのみ構成する必要がある双方向 J2C アクティブ化仕様プロパティーについて説明します。

J2C アクティブ化仕様プロパティー

プロパティー	型	グローバル化	説明
自動作成 EDT	Boolean	はい	EDT テーブルが存在しない場合、アダプターが自動的に作成するかどうかを示すフラグ。デフォルト値は `true` です。
データベース URL	String	はい	エンタープライズ情報システム (EIS) への接続を作成するためのドライバー固有の URL。これは双方向言語で使用される場合に使用可能化されます。
送達タイプ	String	はい	このプロパティーにより、イベントをパブリッシュする順序が決定されます。値は、`ORDERED` または `UNORDERED` のいずれかです。*Ordered* は、イベントが一度に 1 つずつパブリッシュされることを、*Unordered* は、イベントが一度にすべてパブリッシュされることを意味します。デフォルト値は `ORDERED` です。
EDT データベース名	String	はい	イベント配布テーブルのデータベース名。
EDT ドライバー名	String	はい	Inbound イベント用イベント配布テーブルへの接続で使用する XA データベース・ドライバーの名前 (例: `com.ibm.db2j.DB2jXADataSource`)。値が存在しない場合、Event Manager はリカバリーを実行できません。
EDT スキーマ名	String	はい	イベント配布テーブルのスキーマ名。
EDT テーブル名	String	はい	イベント配布テーブルを表すテーブル名
EDT ユーザー名	String	はい	イベント配布テーブルのデータベースに接続するのに必要なユーザー名。
EDT ユーザー・パスワード	String	はい	イベント配布テーブルのデータベースに接続するのに必要なパスワード。

プロパティー	型	グローバル化	説明
イベント順序	String	いいえ	イベントが取得および処理される順序。期待される値はイベント・テーブルのコンマで区切られた列名です。このプロパティーは双方向言語の場合に使用可能。
イベント・テーブル名	String	はい	EIS が Inbound 処理のために生成するイベントを格納する EIS 内のテーブル。
JDBC ドライバー・クラス	String	いいえ	EIS へ接続するために使用される JDBC ドライバー・クラス。
パスワード	String	はい	EIS からイベントを検索する際にユーザーを認可するためのパスワード。このプロパティーは双方向言語の場合に使用可能。
ポーリング間隔	0 以上の整数	はい	新規 Inbound イベントの EIS イベント・ストアをポーリングする間隔 (ミリ秒単位)。`0` を指定すると、アダプターは周期の間待機することはありません。ポーリング周期は固定の間隔で設定されます。つまり、ポーリング周期の実行が遅延すると (例えば、前のポーリング周期の完了が予想よりも遅れた場合など)、遅れを取り戻すために次の周期がすぐに開始されます。このプロパティーは必須です。デフォルト値は `500` です。
ポーリング数量	0 より大きい整数。	はい	このプロパティーは、ポーリング周期ごとに各エンドポイントに配信するイベントの数を決定するのに使用されます。このプロパティーは必須です。
ユーザー名	String	はい	Inbound イベントのため EIS にログインする際のユーザー名。これは双方向言語で使用される場合に使用可能化されます。

双方向 J2C アクティブ化仕様プロパティー

プロパティー	型	説明
BiDi コンテキスト EIS	String	EIS への固有接続用アダプターによってサポートされるビジネス・オブジェクト・ランタイム・インスタンスすべてにあるコンテンツ・データの双方向形式を定義します。
BiDiContextMetadata	String	EIS への固有接続のメタデータまたは構成プロパティーの双方向形式を定義します。

プロパティー	型	説明
BiDiContextSkip	Boolean	リソース・アダプター・レベルでの双方向言語サポートの呼び出しを制御するフラグ。`true` と等しければ、サポートは呼び出されません。`false` と等しければ、サポートは呼び出されます。
BiDiContextSpecialFormat	String	EIS への固有接続のコネクター・プロパティーすべての場合に、特殊な処理を受けるプロパティーのカテゴリーを指定します。
データベース URL BiDi プロパティー		*DatabaseURLEIS、DatabaseURLSpecialFormat、DatabaseURLSkip
パスワード BiDi プロパティー		*PasswordEIS、PasswordSpecialFormat、PasswordSkip
ユーザー名 BiDi プロパティー		*UserNameEIS、UserNameSpecialFormat、UserNameSkip
イベント・テーブル名 BiDi プロパティー		*EventTableNameEIS、EventTableNameSpecial Format、EventTableNameSkip
イベント順序		*EventOrderByEIS、EventOrderBySpecialFormat、EventOrderBySkip
EDT テーブル名		EDT_BiDiFormat、EDT_BiDiSkip、EDTURL_BiDiSpecialFormat、EDTURL_BiDiSkip
EDT ユーザー名		EDT_BiDiFormat、EDT_BiDiSkip、EDTURL_BiDiSpecialFormat、EDTURL_BiDiSkip
EDT ユーザー・パスワード		EDT_BiDiFormat、EDT_BiDiSkip、EDTURL_BiDiSpecialFormat、EDTURL_BiDiSkip
EDT データベース名		EDT_BiDiFormat、EDT_BiDiSkip、EDTURL_BiDiSpecialFormat、EDTURL_BiDiSkip

*BiDi サポートの各プロパティーに関連付けられた 3 つの双方向 (BiDi) プロパティー。これらのプロパティーは以下の表に記述されています。

BiDi プロパティー	型	説明
BiDiContextCP<property_ Name>EIS	String	EIS への固有接続の固有コネクター構成プロパティーの双方向形式を定義します。
BiDiContectCP<property_ Name>SpecialFormat	String	コネクター構成プロパティー・レベルで双方向言語サポートの呼び出しを制御するフラグ。`true` と等しければ、サポートは呼び出されません。`false` と等しければ、サポートは呼び出されます。
BiDiContextCP<property_ Name>Skip	Boolean	EIS への固有接続の固有コネクター・プロパティーの特殊形式を定義します。

関連タスク

59 ページの『サーバーでのアダプターの構成』
アダプター・プロジェクトのプロジェクト Enterprise Application Archive (EAR) ファイルが アプリケーション・サーバーにインストールされたら、必要に応じて、 特定のデータベース・アプリケーションへの通信チャネルをセットアップするために アダプターが使用するプロパティーを再構成することができます。それから、構成済みのアダプター・アプリケーションを開始できます。

接続プロパティー

接続プロパティーはエンタープライズ・サービス・ディスカバリー・ウィザードで、ターゲット・エンタープライズ情報システム (EIS) インスタンスに接続するのに使用されます。

関連タスク

76 ページの『シナリオ 2 の配置と構成』
シナリオ 2 では、アダプター接続プロパティーを設定し、 ビジネス・オブジェクトを生成します。プロジェクトを Enterprise Application Archive (EAR) ファイルにエクスポートし、 アプリケーション・サーバーに配置して、構成プロパティーをリセットします。

メタデータ・ディスカバリー接続プロパティー

エンタープライズ・サービス・ディスカバリー処理には、ディスカバリーのエンタープライズ情報システム (EIS) への接続やサービス記述の作成にこれらのプロパティーが必要となります。

これらのプロパティーを使用して、ディスカバリー・サービスは、オブジェクトの選択およびナビゲーションで表示されるメタデータ・ツリーを作成します。

『メタデータ・ディスカバリー接続プロパティー』の表でこれらのプロパティーについて説明します。

メタデータ・ディスカバリー接続プロパティー

プロパティー	必須であるかどうか	グローバル化	説明
データベース URL	はい	はい	接続先のデータベースの URL。URL は、ドライバーによって異なる場合があります。このプロパティーはドライバーの説明に従って指定してください。このプロパティーは双方向言語の場合に使用可能。
JDBC ドライバー・クラス	はい	いいえ	データベースへの接続に使用されるデータベース・ドライバーの名前
パスワード	はい	はい	対応するユーザー名に対するパスワード。このプロパティーは双方向言語の場合に使用可能。
プレフィックス	いいえ	はい	ビジネス・オブジェクトの名前に追加されるプレフィックス。このプロパティーは双方向言語の場合に使用可能。
ユーザー名	はい	はい	データベースへのログイン用のユーザー名。このプロパティーは双方向言語の場合に使用可能。

関連タスク

47 ページの『接続プロパティーの設定』
アダプター・プロジェクトを作成した後、 エンタープライズ・サービス・ディスカバリー・ウィザードを Adapter for JDBC 用に初期化して、ご使用のデータベース・インスタンス用に接続プロパティーの値を設定する必要があります。

双方向接続プロパティー

これらのプロパティーを指定するとエンタープライズ・サービス・ディスカバリー・ウィザードでエンタープライズ情報システム (EIS) に渡すデータに適切な双方向変換を適用できます。

以下の表に、接続構成中に使用される双方向プロパティーをリストします。

双方向接続プロパティー

プロパティー	値	文字の位置	説明
BiDi 順序付けスキーマ	Implicit Visual	1	論理 (暗黙)。Implicit がデフォルト。 表示
BiDi 方向	LTR RTL contextual_LTR contextual_RTL	2	左から右。LTR (L) がデフォルト。 右から左 コンテキスト LTR コンテキスト RTL
BiDi 対称スワッピング	Yes No	3	対称スワッピングはオン。Yes (Y) がデフォルト。 対称スワッピングはオフ。
BiDi シェーピング	nominal shaped initial middle final isolated	4	公称 (N)。これはデフォルトです。 位置に従って形状を指定された文字。 文字は語頭形。 文字は語中形。 文字は語尾形。 文字は独立形。
BiDi 数字シェーピング	nominal national contextual	5	公称 (N)。これはデフォルトです。 ヒンディ語 (公用語) コンテキスト

関連タスク

47 ページの『接続プロパティーの設定』
アダプター・プロジェクトを作成した後、 エンタープライズ・サービス・ディスカバリー・ウィザードを Adapter for JDBC 用に初期化して、ご使用のデータベース・インスタンス用に接続プロパティーの値を設定する必要があります。

オブジェクト選択プロパティー

これらのプロパティーは、ビジネス・オブジェクトの選択処理で使用されます。これにはフィルター、ノード、および選択プロパティーが含まれます。

フィルターおよびノードのプロパティー

ツリーに表示するスキーマのリストを絞り込む場合、ビジネス・オブジェクトのディスカバリー中にフィルター・プロパティーを指定できます。指定しないと、すべてのスキーマが表示されます。表示するデータベース・オブジェクトのリストを絞り込む場合、ノードのプロパティーを設定します。

『フィルター・プロパティー』の表で、スキーマのフィルタリングに使用しなければならないプロパティーについて説明します。

フィルター・プロパティー

フィルター・プロパティー	型	説明
SchemaNameFilter	String	スキーマをフィルターに掛けるために使用するストリング。指定したストリングで始まるスキーマのみが表示されます。このプロパティーを設定しないと、すべてのスキーマが表示されます。
Types	複数値	データベース・オブジェクト・タイプをリストします。テーブル、ビュー、ストアード・プロシージャー、同義語/ニックネームなどです。値はエンタープライズ・サービス・ディスカバリーによって設定されます。複数選択することができます。
DefineASI	Boolean	true の場合、生成するオブジェクトを選択すると、ビジネス・オブジェクトのアプリケーション固有情報 (ASI) 値を設定するよう求めるプロンプトが出されます。

『ノード・プロパティー』の表で、データベース・オブジェクトのフィルタリングで使用しなければならないプロパティーについて説明します。

ノード・プロパティー

ノード・プロパティー	型	説明
ObjectNameFilter	String	データベース・オブジェクトをフィルターに掛けるために使用するストリング。指定したテキストで始まるデータベース・オブジェクトのみが表示されます。ObjectNameFilter を指定しないと、すべてのオブジェクトが表示されます。

関連概念

40 ページの『システム機能のディスカバリー』
アダプターは、データベースを分析し、スキーマを識別することによって、 データベース内のビジネス・オブジェクトをディスカバーします。次に、データベースからすべてのオブジェクトのリストを作成し、 それをツリー構造として示します。

選択プロパティー

データベース・オブジェクトを選択した後、選択プロパティーの値を設定する必要があります。

エンタープライズ・サービス・ディスカバリー・ウィザードが、選択プロパティーを照会します。『選択プロパティー』の表で、これらのプロパティーについて説明します。

選択プロパティー

選択プロパティー	型	説明
BO ロケーション	String	生成された .xsd ファイルが保管されるロケーションへのパス。 DataDescription で RelativePath として設定されます。
レコード最大数	Integer	RetrieveALL 操作で検索するレコードの最大数。
ネーム・スペース	String	スキーマのフィルタリングに使用するテキスト項目。
操作	複数値	デフォルトでは、選択されたサービス・タイプに対してアダプターがサポートする操作のリストです。これらの操作から選択します。指定された操作は、生成されるすべてのビジネス・オブジェクトに対して設定されます。
サービス・タイプ	String	Inbound または Outbound が選択されます。値はエンタープライズ・サービス・ディスカバリーによって設定されます。

ネーム・スペース・プロパティーは、最初はすべてのビジネス・オブジェクトでデフォルトのネーム・スペースに設定されています。デフォルトのネーム・スペースは `http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc` です。

ネーム・スペースは、ビジネス・オブジェクト・スキーマを論理的に分離するため、ビジネス・オブジェクト名の前に付加されます。例えば、`http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/Schema1Customer` のようになります。

関連概念

41 ページの『オブジェクトの選択および生成』
ビジネス・オブジェクトを生成するには、データベース・オブジェクト・ノードを選択します。そうすると、エンタープライズ・サービス・ディスカバリー・ウィザードで、選択したノードのオブジェクトのビジネス・オブジェクトが生成されます。

関連タスク

53 ページの『選択プロパティーの設定』
データベース・オブジェクトを選択した後、 インポート・ファイルとエクスポート・ファイルの選択プロパティーの値を 指定する必要があります。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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

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

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

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、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 Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010

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。

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。

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

著作権使用許諾:

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

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

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

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 LOGO
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
HelpNow
i5/OS
IMS
Informix
iSeries
Lotus
Lotus Domino
Lotus Notes
MQIntegrator
MQSeries
MVS
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
WebSphere
z/OS

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。

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。

Intel、Intel(ロゴ)、Intel Inside、Intel Inside (ロゴ)、Pentium、Intel Centrino、Intel Centrino (ロゴ)、Celeron、Intel Xeon、Intel SpeedStep、および Itanium は、Intel Corporation の米国およびその他の国における商標です。

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

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

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

WebSphere Adapters バージョン 6.0

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